Computational Psycholinguistics — Assignment 2

Daan Brugmans (S1080742)

I. Introduction

This report is the realization of the Assignment 2 project for the Radboud University course Computational Psycholinguistics. For this assignment, students must investigate whether the gradients computed from a recurrent neural network correlate with measured P600 component activity from a controlled experiment. The reasoning behind this assignment is that recent research ([1], [2]) has shown that the P600 component may be the backpropagation of prediction errors in the human language system. Since neural language models also backpropagate their prediction errors using gradients, there may exist similarities between the language error backpropagation of human and artificial neural language systems. This report contains the findings found by me for the Assignment 2 project.

The relevant code for this assignment can be found at the following URL: https://github.com/daanbrugmans/ru-computational-psycholinguistics-23-24/tree/main/assignment-2/code.

II. Related Work

For the Assignment 2 project, students must choose a controlled experiment where participants read English or Dutch sentences while their P600 component is measured. I have chosen to use the data from the controlled experiment performed in [3]. This data is also used by the authors of [2], and is briefly introduced in the description of the Assignment 2 project. This means that my report is (an attempt at) an extension of the work in [2].

The data from the controlled experiment in [3] consists of EEG data of 24 native British English speakers, who all read a set of 205 sentences taken from English-language novels. The authors recorded EEG data of six different ERP components, including the P600 component, and calculated the average EEG values for every ERP component, for every word of every sentence for every participant of the experiment. This dataset should fulfill the requirements set by the assignment: the experiment language is English, the participants' stimuli are independent sentences, the size of the P600 component is one dependent variable and showed an effect, and the independent variables are manipulated by varying the content of the sentence stimuli.

III. Methodology

All the code I have written can be found in the Jupyter Notebook called `main.ipynb`, which should thusly contain all results also shown in this report. It can be found at the following URL: https://github.com/daanbrugmans/ru-computational-psycholinguistics-23-24/blob/main/assignment-2/code/main.ipynb I have placed the code in the `get_predictions.py` file in a function called `get_predictions`, so that I can easily call this code from the notebook.

The dataset of the study in [3] is openly accessible through a collection of `.mat` files, which are typically used in a MatLab setting, but I use Scipy to open this data. Within the collection of code I hand in alongside this report, the raw data file can be found in the `data` folder as `code/data/stimuli_erp.mat`. The original README provided by the authors is also included. From this dataset, I collect the list of sentences read by the participants of the study, and the relevant ERP data. Although this data does not need any preprocessing, I do wrangle the ERP into the format used by the code in `get_predictions.py`, so that I can easily merge the data from the study with the data generated by the models. I also write the sentence data from the study to a plain text file in the `items` folder called `stimuli.txt`, as is required for generating model results.

The data from [3] includes the P600 component for all 24 participants per word. I have chosen to take the mean of all 24 P600s and only analyse these means. I made this decision because I wanted to have a single P600 for every word to represent the entire participant population. I have chosen the mean as opposed to the median because of its sensitivity of outliers: I wanted to include information about participants with outlying P600 values for specific words into the aggregated P600 value, and I think that the mean does a better job at this than the median due to its sensitivity to outliers.

My results are a collection of scatterplots that visualize the relationship between 3 variables on the sentence data: the Mean P600 Component, the Surprisal of a Model, and the Gradient Values of a Model. Every plot also contains a quadratic function fitted to the data to show the trend of the relationship between the variables. I have chosen for a quadratic function as opposed to a linear function, as I found that this fits the data better. Included in every plot is also the Pearson Correlation Coefficient r between variables. Since surprisal values and gradients are calculated for multiple models, I provide a plot for every model. For every pair of variables, I also provide a plot of how the Pearson Correlation Coefficient changes as the model is trained on more data, which will be the main focus of my results.

Before producing results, I set a universal seed for Python itself, NumPy, and PyTorch of `3131`. I do this in order to improve the reproducibility of my results. This is why I also set some settings for CUDA regarding PyTorch's randomness when computing on GPU.

IV. Results

V. Conclusions

References

[1] Hartmut Fitz and Franklin Chang. Language erps reflect learning through prediction error propagation. *Cognitive Psychology*, 111:15–52, 2019.

[2] Stefan L Frank. Neural language model gradients predict event-related brain potentials, Jan 2024.

[3] Stefan L. Frank, Leun J. Otten, Giulia Galli, and Gabriella Vigliocco. The erp response to the amount of information conveyed by words in sentences. *Brain and Language*, 140:1–11, 2015.

APPENDIX