Analyzing Next-Word Prediction Using Regression Models and MEG Data

Solmih Kim
DATASCI 125
June 5, 2024

Acknowledgement: Laura Gwilliams, Stanford Data Science
This short-term research project is a continuation of the research conducted by Schonmann et al., 2022

Wu Tsai
Neurosciences Institute
Stanford University

Infosys

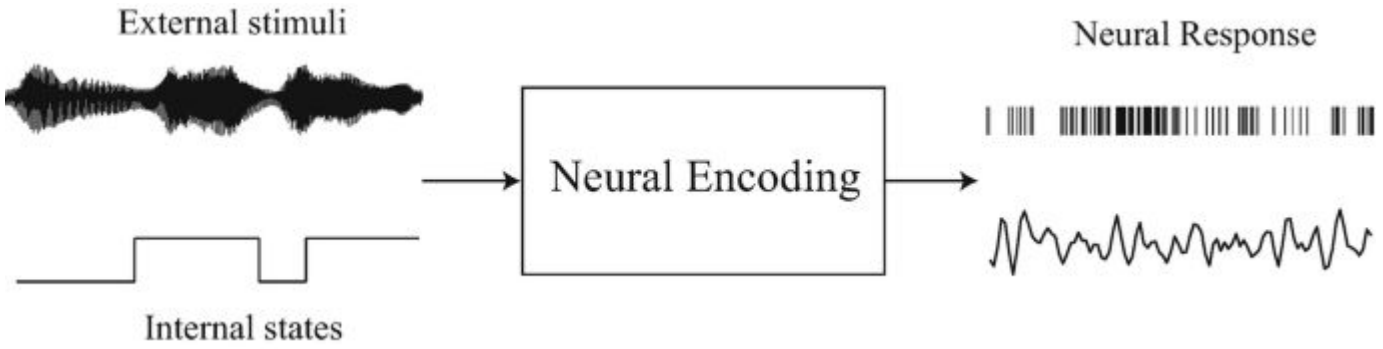
Stanford MEDICINE | Snyder Lab
Department of Genetics

peachy day

Background

- ***Language processing*** involves transforming stimulus streams into a hierarchy of representations
- Intersection of computer science and neuroscience
  - Natural Language Processing (NLP)
  - Deep Learning
- Word prediction based on a neural data gathered from text stimulus

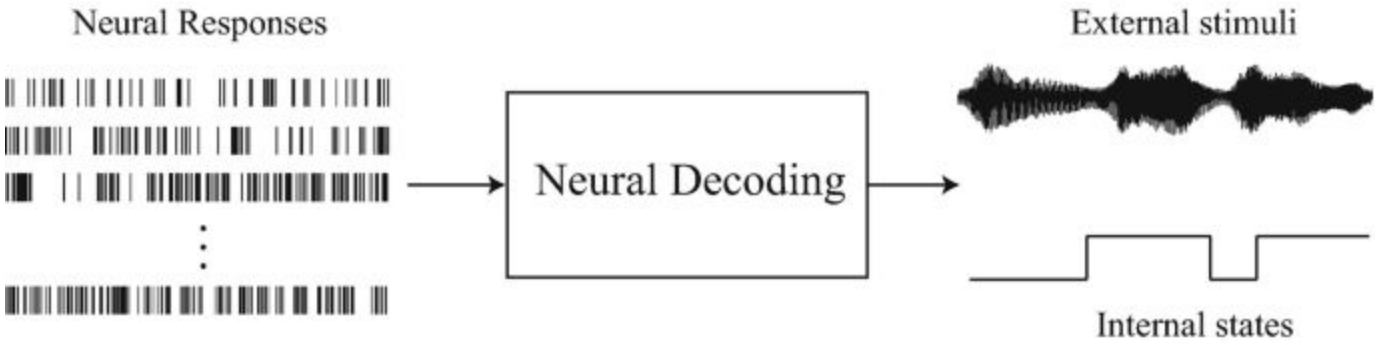
Encoding Model



GPT-2 features

Word Prediction

Decoding Model



Word Text

GPT-2 features

Methods

Neural data type: Magnetoencephalography records (MEG)

Number of participants: 3 participants total

- MEG experiment

Stimulus:

- 10 hours of speech from 10 stories from Doyle's The Adventures of Sherlock Holmes
- For the purposes of the study, the MEG data was collected in ten 1-hour sessions

Methods Cont. (Preprocessing)

- Load MEG data for participant during session
- Define and resample frequency after downsampling
- Decimate the data by a factor of 12
    - Epochs data now has 1258 words, 269 channels, and 151 time points
- Packages used: numpy, pandas, sklearn, spaCy, torch, transformers (GPT2Tokenizer, GPT2Model), pickle, scipy.stats

Research Question

How do different encoding mechanisms or models influence the ability of word embeddings to predict brain activity?

1. What layer of GPT-2 is most optimal for extracting word embeddings?
2. What do the different regression models' spearman correlation say about their encoding performance?

Analysis 1

- Feed GPT-2 simple sentences to see which layer would express the best cosine similarity
  - “I went to the park and saw a dog.”
  - “I went to the park and saw a cat.”
  - “I went to the park and saw a truck.”

What layer of GPT-2 is most optimal for extracting word embeddings?

Results 1

- By looping through each layer and computing the cosine similarity of the last word in each sample sentence, we saw that GPT-2's layer 12 had the highest cosine similarity score (0.9954)
- Deeper layers of transformer architecture will learn abstract and high-level representations better

Fig. 1

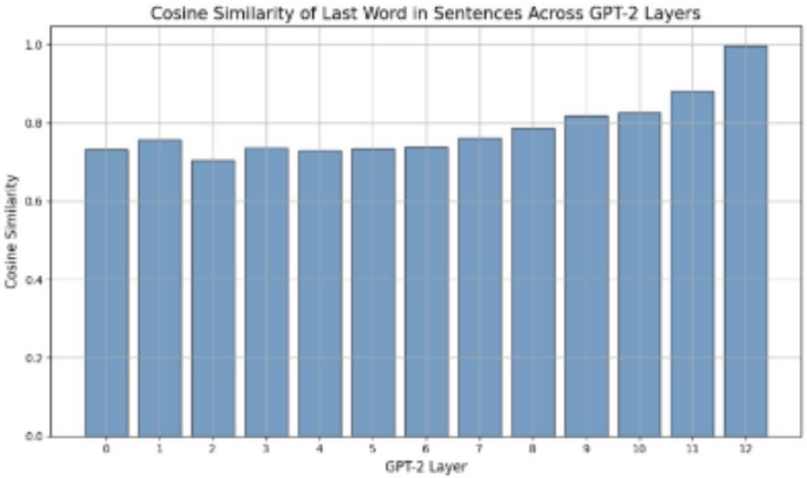


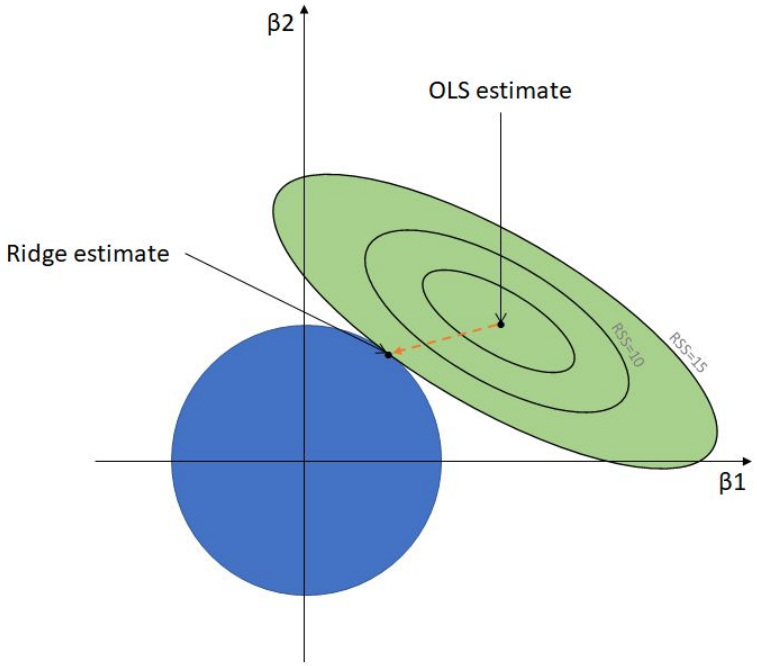
Figure 1: Cosine Similarity of the Last Word in Sentences Across GPT-2 Layers. The bar chart shows the cosine similarity values for the last word in sentences across 13 layers of GPT-2. Layer 12 exhibits the highest cosine similarity, indicating a stronger alignment in the vector space for this layer compared to others.

Analysis 2

- Based on the Schonmann paper, we replicated the encoding model by using ridge regression, looping through 269 channels and 151 time points
- Run random forest regression and PCR on best channels:
  - 4, 5, 27, 20, 10, 9, 2, 22
- Compute Spearman correlation coefficient (SCC) for each of the models and visualize the results

What does the regression models spearman correlation say about its performance?

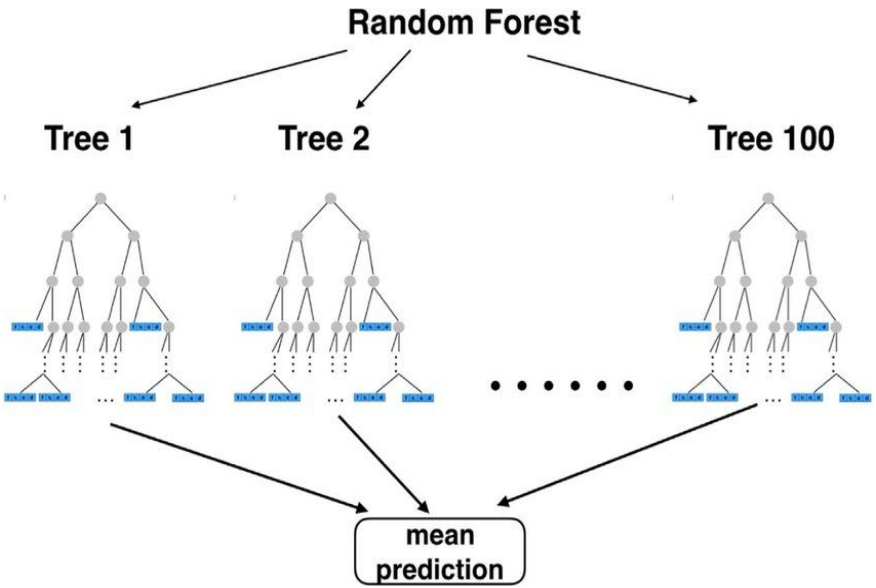
Ridge Regression Model



Multicollinearity denotes when two or more predictors have a near-linear relationship

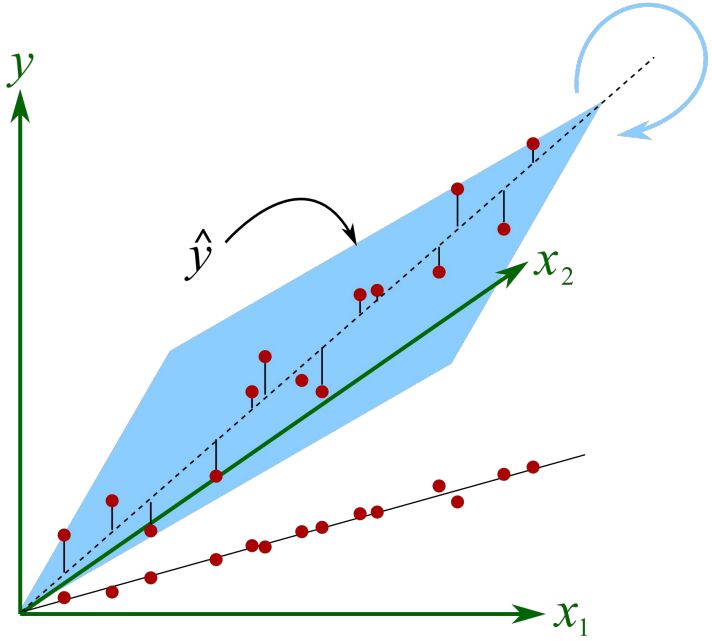
- Standard Ordinary Least Squares (OLS) vs. Ridge Regression

Random Forest Regressor 🌲



- Ensemble learning method that utilizes multiple decision trees to make predictions
- Nodes representing features and branches representing rules
- The method averages the prediction from all the trees to make a more robust prediction

Principal Component Regression (PCR)



- Step 1: perform principal component analysis (PCA)
- Step 2: linear regression between samples scores on factors most correlated with Y and Y
- Set a linear constraint on the regression coefficients to address multicollinearity

Results 2

- Ridge Regression – As time increases, the spearman correlation increases, moving further from 0
    - Best channel: 5, Best time point: 135, SCC: 0.1082
- Random Forest Regression – As time points increase, the spearman correlation increases for *some* channels but not all
    - Best channel: 5, Best time point: 136, SCC: 0.0748
- PCR – The spearman correlation are arbitrarily above or below chance
    - Best channel: 2, Best time point: 12, SCC: 0.3308

Fig. 2

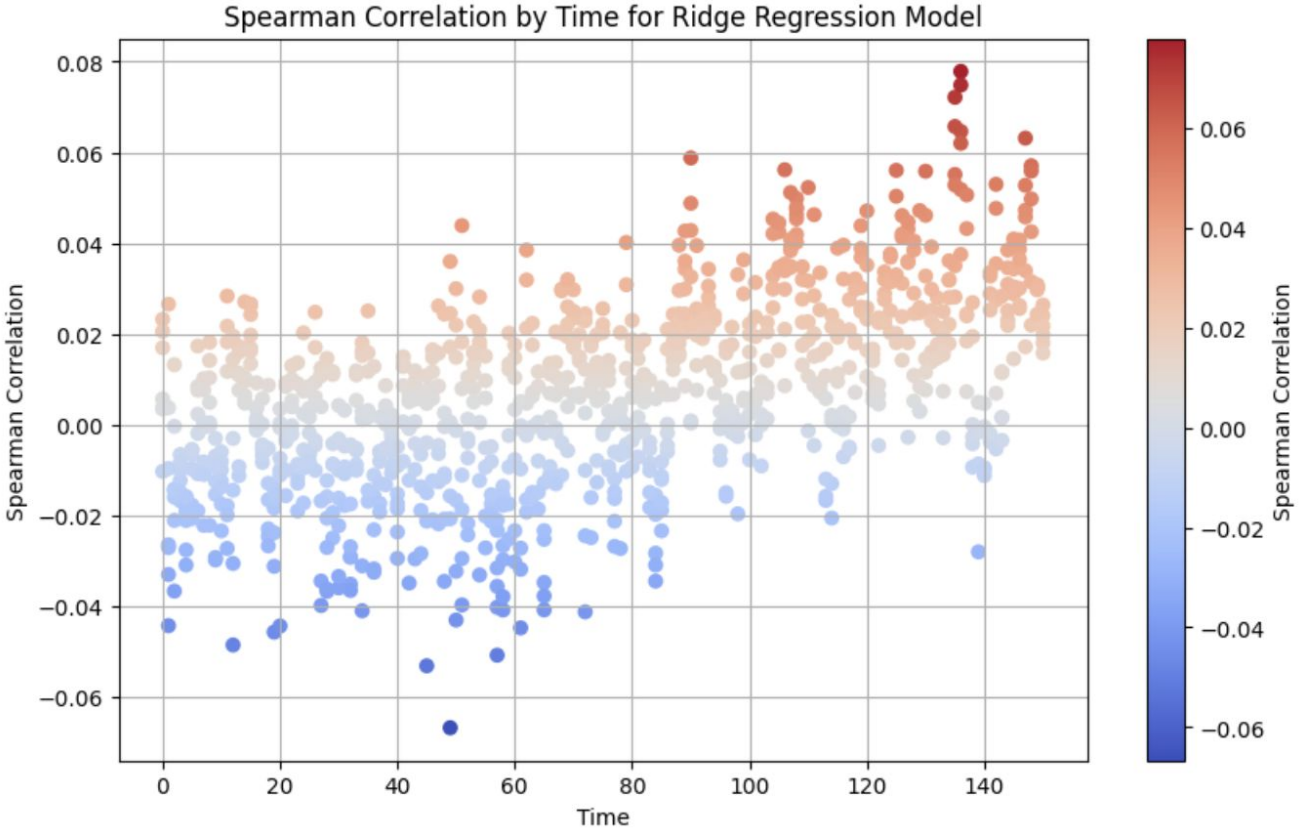

Spearman Correlation by Time for Ridge Regression Model
Spearman Correlation
Time
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
20
40
60
80
100
120
140
Spearman Correlation

Fig. 3.1

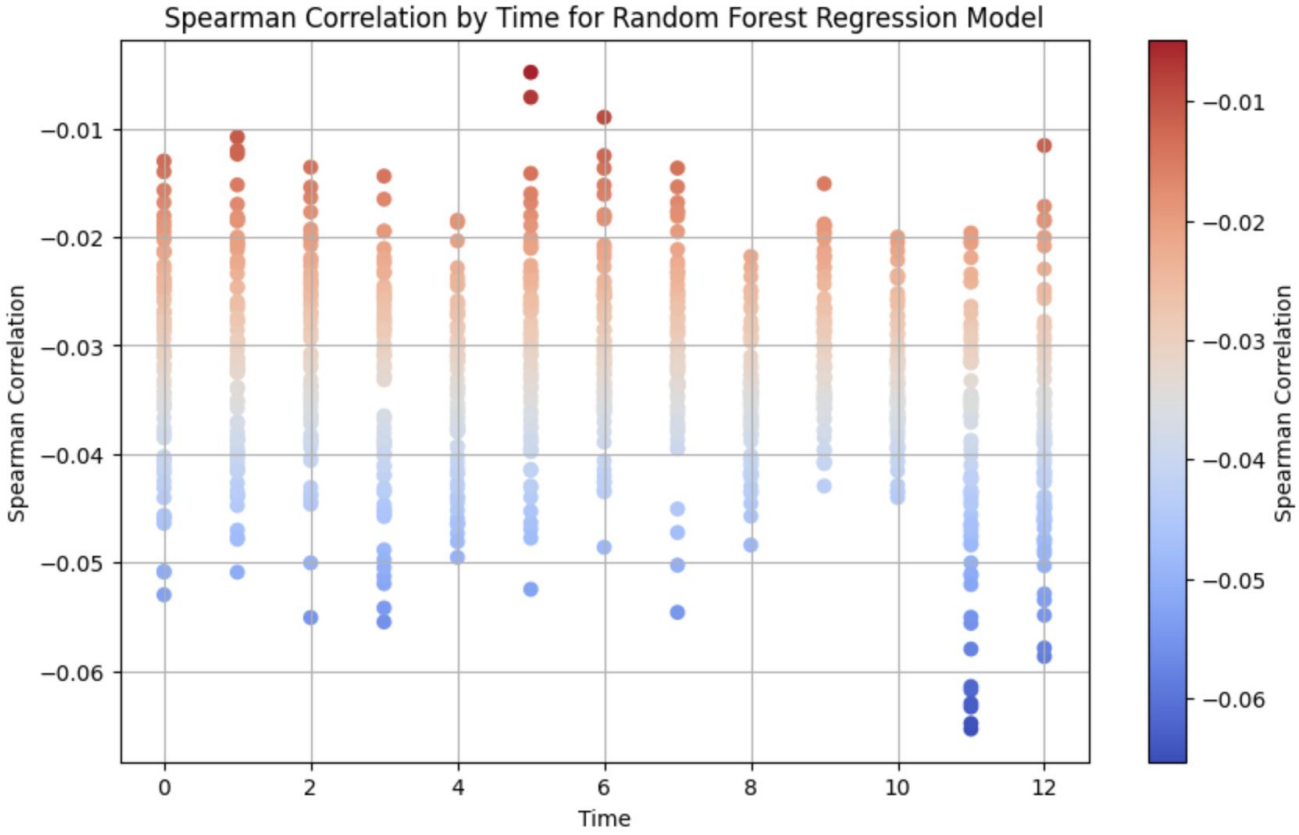

Spearman Correlation by Time for Random Forest Regression Model
Spearman Correlation
−0.01
−0.02
−0.03
−0.04
−0.05
−0.06
0
2
4
6
8
10
12
Time
Spearman Correlation
−0.01
−0.02
−0.03
−0.04
−0.05
−0.06

Fig. 3.2

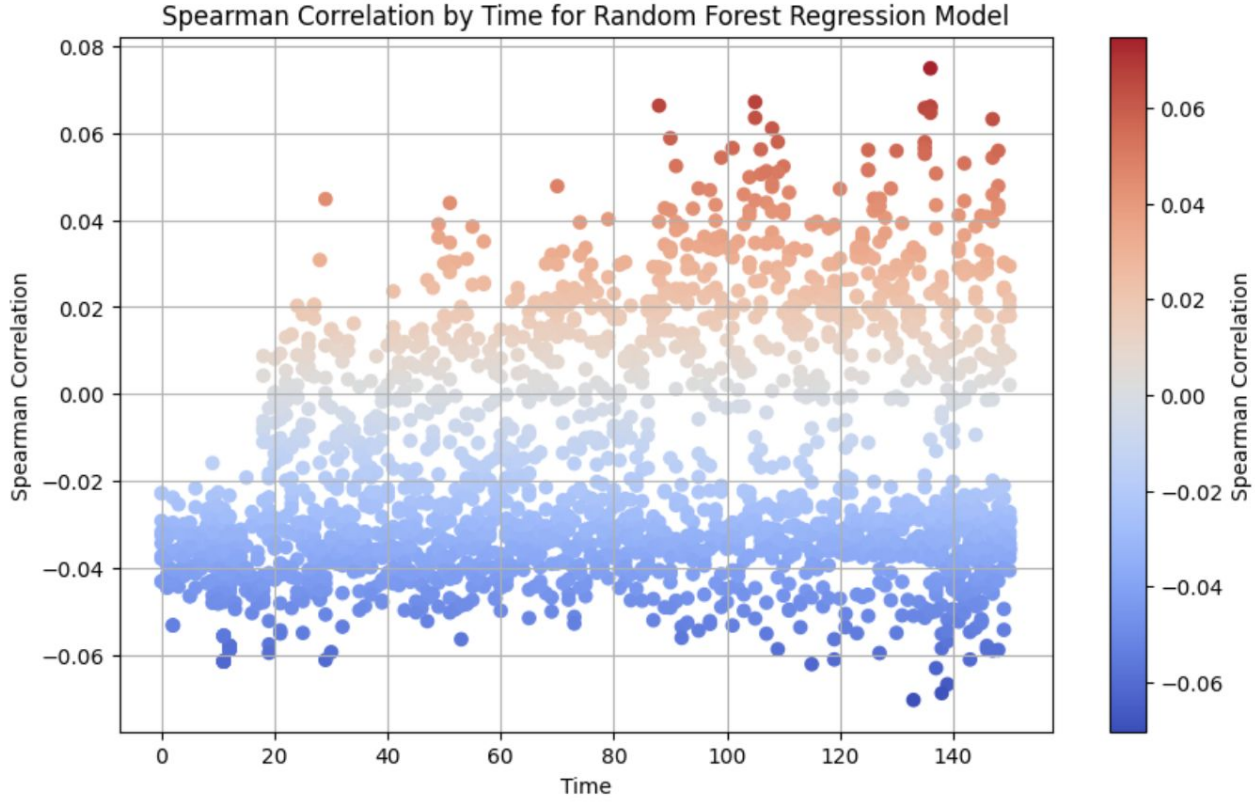
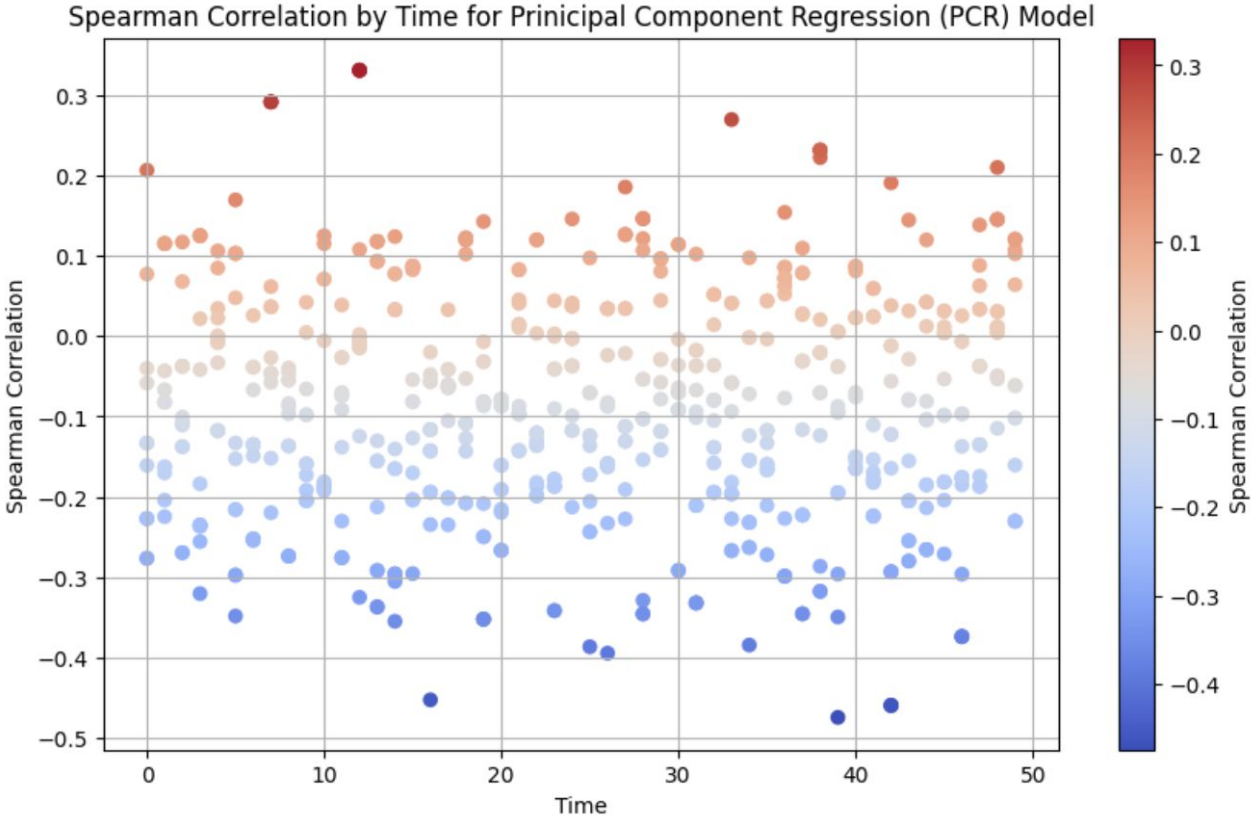

Spearman Correlation by Time for Random Forest Regression Model
Spearman Correlation
Time
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
20
40
60
80
100
120
140
Spearman Correlation

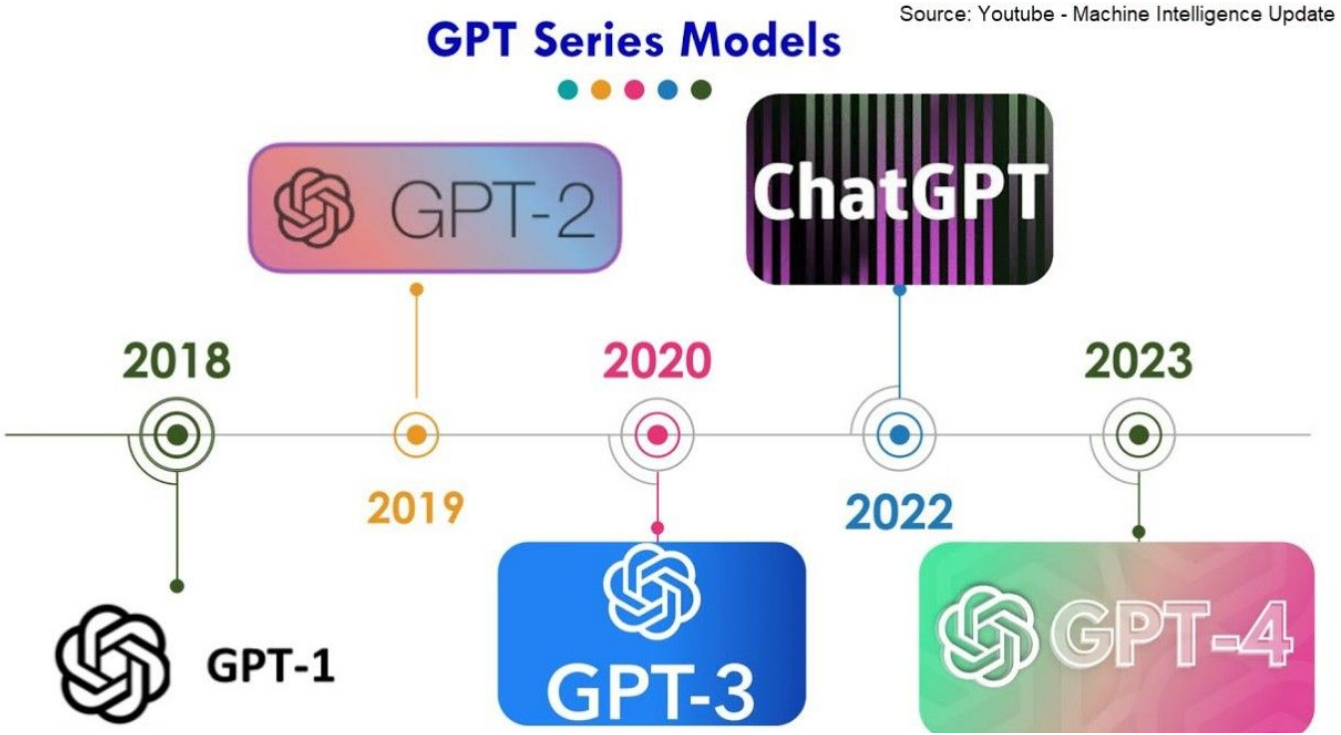
Fig. 4


Spearman Correlation by Time for Prinicipal Component Regression (PCR) Model
Spearman Correlation
Time
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
0
10
20
30
40
50
Spearman Correlation

Discussion

- The ridge regression model performed the best due to its increasing Spearman correlation coefficient with time
- A more positive coefficient implies a strong relationship between word embeddings and neural response
- Ridge regression performs better than PCR despite similar approaches
    - PCR requires PCA ➜ computationally expensive and intensive
    - When PCR uses all predictors, the principal components will explain variance in predictors. If components capture insufficient variance, important information lost ➜ underfitting

Large Language Models



Questions?

solmihk@stanford.edu