The Hidden Knowledge of Untrained Neural Language Models

by

Benjamin J. Ware

(Under the Direction of Frederick Maier)

Abstract

Recent research by Schrimpf *et al.*, Pasquiou *et al.*, and Hosseini *et al.* has indicated that, surprisingly, language-elicited fMRI responses could be predicted from activations of untrained language models by simple linear regression. This present study aims to explore the untrained models predictiveness of word features to better understand what allows them to predict fMRI data. Using the code of Schrimpf *et al.* the results of were reproduced for the GPT-2 and GPT-2-XL LMs on the Blank2014 and Pereira2018 datasets. In the context of the fMRI prediction methodology, untrained LM activations were found to be predictive of many word features, especially POS and Ngram frequencies, although less so than their pretrained counterparts. Untrained LMs were also found to be more predictive of word features for the next word than previous word targets. When looking at the performance of individual layers, whereas untrained LM predictions of fMRI data appear to increase asymptotically with increasing depth, their predictions of word features appear to decrease asymptotically with increasing depth. Ultimately, although the untrained LM prediction of fMRI data might be partially explained by the LM's prediction of word features, further explanation is required to explain the degree of performance in fMRI prediction.

INDEX WORDS: Deep Neural Networks, Language Models, fMRI, Linguistics, NLP, Machine Learning

THE HIDDEN KNOWLEDGE OF UNTRAINED NEURAL LANGUAGE MODELS

by

BENJAMIN J. WARE

B.A., Bemidji State University, 2011
B.S., University of North Dakota, 2014

A Thesis Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

MASTER OF SCIENCE IN ARTIFICIAL INTELLIGENCE

ATHENS, GEORGIA

2024

The Hidden Knowledge of Untrained Neural Language Models

by

Benjamin J. Ware

Major Professor: Frederick Maier

Committee: Frederick Maier
Khaled Rasheed
Yuri V Balashov

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
May 2024

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This research aims to better understand how untrained LM activations can predict language-elicited fMRI signals.

Recently, Schrimpf *et al.* reported that language-elicited fMRI responses could be predicted from activations of untrained language models by simple linear regression [1]. This has been found in subsequent research for language models [2], as well as previous research in computer vision models [2]. The setup used by Schrimpf *et al.* involved predicting heldout fMRI data points for each word in a stimulus from the activations extracted from feeding the words into a language model (LM) by training a simple linear regression model on the training data points. Where trained models were able to perform close to the maximal level given the noise of the data, surprisingly, untrained models were able to reach about half the maximal level.[1]

Schrimpf *et al.* suggested some possible properties that may cause untrained networks to be predictive of fMRI data. The fact that the performance of the untrained model was correlated with the performance of the trained model suggests that the untrained performance is sensitive to the model architecture itself. Schrimpf *et al.* further showed that the untrained model scores were correlated with their scores on task of next-word prediction (with a similar linear prediction

[1] Performance is based on the Pearson correlation coefficient between the held-out target values and the predicted values of the untrained models.

mechanism), suggesting that the model's performance on fMRI prediction may be related to models inadvertently capturing linguistic properties. They further ruled out basic alternative explanations, such as overfitting common words and word sequences.

The main objective of my research investigation was to develop mechanical explanations for these findings by examining what word features the untrained models are able to predict. My first experiment mirrored the fMRI prediction mechanism used by Schrimpf *et al.* and examines the prediction of word features from LM activations. My second experiment is based on research by Baek *et al.* on untrained computer vision models [3]. In a study published in Nature, they found that untrained neural vision models could robustly develop activations that effectively model an unexpectedly high-level feature: the distinction between face vs non-face. Trained LMs have also been shown to have interpretable neuron activations [4], so for my second experiment, I adapt some of the Baek *et al.* methodology for use on untrained LMs to see whether any untrained LM activations can be interpreted likewise as POS classes.

1.2 High-level Core Prediction Mechanism

In this line of research, language is modeled as a sequence of words. Using human subjects and methods from neurophysiology, each word can be associated with multiple fMRI response datasets. Using language models (LMs) and methods from machine learning, each word can be associated with multiple LM response datasets (composed of model activations). Finally, a simple linear regression model with cross-validation is trained to predict the sets of fMRI responses from each of the LMs.

Both categories of responses are essentially projections of each word into a feature space. The LM projection presumably contains features that, through training, are useful for the computation of the next-word prediction. Similarly, the fMRI projection presumably contains information relevant to whatever language-specific tasks the brain performs. Thus, an LM's prediction performance depends on how similar its projections are to the brain's projections, or more precisely, on how similar a linear combination its projections can be to the brain's projections.

Under this theory, an untrained model is predictive of fMRI data because its projections are similar to brain projections. To add interpretability, my research strengthens this theory to the following:

> *A linear combination of untrained LM projections is able to model the brain's projections because a linear combination of untrained projections can model simple interpretable linguistic features.*

1.3 Background

1.3.1 fMRI prediction

My research centers around my two experiments. The methodology of my first experiment is based mainly on the previously mentioned research of Schrimpf *et al.* [1]. To my knowledge, there are two other studies that examined untrained neural language models in the context of fMRI prediction, Hosseini *et al.* [5] and Pasquiou *et al.* [2], which further focused my research. The methodology of my second experiment is based on Baek *et al.*'s research (which did not involve fMRI or language models).

Schrimpf *et al.* was primarily concerned with the effects of model architecture. Their analysis attributed the difference between the two models' fMRI prediction performance to the interpretable differences in their architectures. In some cases, this made for very convincing arguments. For example, they found that context-independent models, such as GloVe or random word embeddings of the same size as the GPT-2-XL embedding, could not predict the fMRI data very well. However, when it came to comparing between two well-performing models, they could only suggest that the GPT-2 architecture performed better than the BERT architecture because the BERT model was bidirectional and the GPT-2 model was unidirectional, so like the brain, it could not perceive words ahead of the current word.

Hosseini *et al.*, of whom Schrimpf was a coauthor, mainly examined the effect of the quantity of training data [5]. They found that a GPT-2-XL model could reach near-maximum fMRI prediction on only 100 million words of training data, which is approximately equal to the number of words children are exposed to during their first 10 years of life. Furthermore, it included a sub-experiment that found that when the first layer of an untrained GPT-2 model was reinitialized with a simple Gaussian distribution $\mathcal{N}(0, 0.02)$, the untrained model could no longer predict the fMRI data above chance. Thus, model performance is critically dependent on the initialization choices choices of untrained networks; the network architecture itself does not explain the fMRI prediction.

The research of Pasquiou *et al.* focused on examining which regions of the brain were predicted by trained vs untrained LMs [2]. Most interestingly, they essentially found that the model's best fMRI prediction occurred in two distinct types of region. One type of region is what one would expect: The untrained models were not very predictive of the region, but the trained models were. But in the second type of region, both trained and untrained models were predictive of the fMRI data, but trained models were not significantly more predictive. This could suggest that this second is sensitive to fairly simple features, already captured by the untrained models at initialization.

Regarding the model architecture, Pasquiou *et al.*'s results directly contradicted Schrimpf *et al.*: BERT was reported to outperform GPT-2, and the less complex untrained models (GloVe and LSTM) outperformed the more complex untrained models(GPT-2 and BERT). There are a number of differences in methodologies (for example, Pasquiou *et al.* trained their models in house while Schrimpf *et al.* used pretrained models), and it is not clear which of the differences cause the discrepancies.

This seems to be a fundamental problem with the methodology of predicting the black box of the brain with the black box of a language model with no interpretable intermediate. Therefore, my research focuses on the use of interpretable linguistic features.

Fully trained neural models are known to contain some individual activations that can be directly interpretable as high-level features [4] but Baek *et al.* found that even untrained *vision* models contain neurons that can be interpreted as face selective, an unexpectedly high-level feature [3]. These face-selective neurons were defined by having significantly higher mean activations on face images than on five other non-face classes of images, Baek demonstrated with a number of tests that these activations not only correlate with face vs. non-face, but act face-selective in other ways as well. For example, they retained their selectivity when evaluated on images from other datasets and images generated with generative neural networks. Face-selective activations showed a higher response to images where local features were disrupted compared to images where global features were disrupted, showing that they were not just extrapolating from simpler local features, such as nose-shape. Furthermore, they trained a face vs. non-face image SVM classifier on the 1.1% of the face-selective neurons in the final layer, and it performed similarly well to an SVM that was trained on the entire final layer of activations. My second experiment replicates attempts to apply this methodology to language models.

1.4 Objectives

The primary objective of this research is to better understand how the activations of untrained GPT-2 models can predict language-elicited fMRI responses by examining how well they model word features.

1. **What word features can be predicted by a linear combination of LM activations?**

The theory is that if a linear combination of LM activations can model simple word features and a linear combination of simple word features can model the fMRI data, then it logically follows that a linear combination of LM activations can also model the fMRI data. To test this, I examined how well a linear combination of LM can predict the current word's part of speech, word frequency, bigram and trigram frequencies, word order, word depth in the tree, function vs. content distinction, and the next word's part of speech. Each of these is predicted for the previous, current, and next-word features.

2. **Do untrained LMs contain activations that are directly interpretable as high-level features?**

The theory here can be thought of as a stronger version of the previous theory. By disallowing linear combinations for representing word features with activations, this experiment essentially tests how explicitly word features are represented. The experiment adopted Baek *et al.*'s methodology that used neural computer vision models for use with the untrained GPT-2 models: 6 classes, 200 training images per class, with neurons evaluated using a two-sided rank sum test ($P < 0.001$) for the class with the highest mean score. Even negative results for this experiment would be interesting; I would not have expected positive results if not for Baek *et al.*.

1.5 Experiments

1.5.1 Models and Datasets

I chose to use the GPT-2 architecture because Schrimpf reported that GPT-2 outperforms all others and because it is more cognitively plausible than the other transformer models because it is unidirectional [1]. Furthermore, I wanted to examine the results of the Gaussian start which used the smallest 13 layer GPT-2 variant [5]. All experiments were carried out with a total of 21 model instances, all utilizing the GPT-2 architecture:

- XL-Trained (GPT-2-XL, 49 layers, 1.5B parameters)
- XL-Untrained (GPT-2-XL, 49 layers, 1.5B parameters)
- GPT-Trained (GPT-2, 13 layers, 117M parameters)
- $9\times$GPT-Untrained (GPT-2, 13 layers, 117M parameters)
- $9\times$GPT-Gaussian (GPT-2, 13 layers, 117M parameters)

As in Schrimpf *et al.*, all trained models were obtained from the Hugging Face transformer library [6], using the default pretrained model weights.

Schrimpf's published code was used to calculate each model's ability to predict fMRI data, reported at the layer-by-layer level, against the three fMRI datasets used in their original paper, Blank2014 [7], Fedorenko2016 [8], and Pereira2018 [9]. Of these datasets, Schrimpf *et al.* primarily focused on the results of the Pereira2018 dataset, which was approximately 3 orders of magnitude larger than the other two. For the most part, I was able to reproduce the published scores for Blank2014 and Pereira2018, but was unable to get the code to run on the Fedorenko2016 dataset. For the Pereira2018 dataset, I was able to reproduce Schrimpf *et al.* published results for the pretrained models GPT-2 and GPT-2-XL with variation in the second decimal place. For the Blank2014 dataset, I was able to reproduce the exact published results for the pretrained GPT-2 and GPT-2-XL except for the model's 0th layers (though the 0th layers scored). However, the Blank2014 scores were ultimately excluded from further analysis, primarily because of some unexpected behavior on control models. Blank2014 scores, which matched the published Schrimpf *et al.* scores, were discarded primarily due to unexpected behavior on control models. (Please see section 3.3 for a more complete discussion) Thus, in the end, only Pereira2018 was used for my experiments.

The base corpus for word features is the Hugging Face implementation of the OntoNotes corpus (formally known as the"CoNLL2012 shared task data based on OntoNotes 5.0"). [10] The following nine categories of word features were selected for experiment.

- unigram frequency (i.e., word frequency)
- bigram frequency
- trigram frequency
- word order
- tree depth
- POS-51
- POS-12
- POS-7
- Function vs content

Ngram frequencies were calculated over the entire OntoNotes corpus. Word order is the word's position in a sentence, clipped at 34. Tree depth is the word's syntactic depth in the sentence, calculated from the corpus's provided trees, clipped at 17. POS-51 is the manually annotated POS tags. POS-12 is a simplified version of the POS-51 tags, using NLTK's map_tag function. POS-7 is a manually further simplified tagset primarily for use in my second experiment[2]. Finally, the function versus content distinction can be thought of as a further simplified binary POS distinction between function words (with primarily syntactic value) and content words. To keep the number of samples consistent, I included a third category of punctuation, resulting in function versus content being a trinary class.

Activations were retrieved from the hidden states of the model on a per-layer basis. The first 500 sentences of the OntoNotes corpus "training split" were used to generate activations for each word per model per layer. Values for the first 386 words (22 sentences) were discarded due to insufficient context length, resulting in 7958 samples. This yielded 768 activations per layer/word from the base GPT-2 models and 1600 for the GPT-2-XL models.

1.5.2 Experiment 1: *What do untrained models know?*

In this experiment, I explored how well a linear linear combination of each layer's activations of each model could predict each of the word features. For continuous word features (the ngram frequencies), ordinary linear regression was used to generate predictions for each instance using 5-fold cross-validation. For each fold, the Pearson correlation coefficient between the held-out and predicted values was calculated. Finally, the mean of these coefficients is calculated and taken to

[2] In my second experiment, the seventh "misc" category is discarded.

be the performance metric of the predictive model and reported as the mean Pearson's R for the combination of layer / target. The other word features were categorical, and so multi-target logistic regression was used to generate predictions for each instance using *stratified* 5-fold cross-validation. The performance metric for categorical targets is categorical accuracy.

1.5.3 Experiment 2: ***Do some activations directly code for features?***

This experiment adapts Baek's methodology of neuron-by-neuron study, originally used on computer vision models, for use on language models [3]. First, the data instances were partitioned by their POS-7 class, and all but the six classes with the largest populations in the entire OntoNotes corpus were discarded, resulting in the following classes.

1. Noun (includes Pronouns)
2. Verb
3. Adposition
4. Determiner
5. Adjective
6. Adverb

Since the smallest partition had 387 instances, each partition was randomly sampled for 200 training instances, 100 validation instances, and 75 training instances for a total of 2250 words of data. The activations of each unit were normalized by z-scoring over all 2250 words. For each neuron, the class with the highest mean activations was compared with the others using a two-sided rank sum test ($P < 0.001$) to determine whether the difference in the mean of two distributions is statistically significant. If the difference was significant against all other categories, then that neuron was categorized as selective for that class. The resulting counts and their distribution over the layers were then analyzed over the network.

1.6 Conclusion

My research explored the language predictiveness of untrained neural language models to better understand how their activations were predictive of neural data. As expected, pretrianed model activations were able to predict word features above chance, especially POS and ngram features. In the context of the fMRI prediction methodology, untrained LM activations were found to be predictive of many word features, especially POS and Ngram frequencies, although less so than their pretrained counterparts. Untrained LMs were also found to be more predictive of word features for the next word than previous word targets. When looking at the performance of individual layers, whereas untrained LM predictions of fMRI data appear to increase asymptotically with increasing depth, their predictions of word features appear to decrease asymptotically with increasing depth, which appears contradictory.

The larger untrained model was significantly more predictive of fMRI data than the smaller untrained model, but significantly less predictive of all linguistic targets. For trained models, the results were similar but more nuanced. Relative to the smaller model, the larger trained model again outperformed on fMRI prediction, again unperformed on POS, word order, and syntactic tree depth predictions, but had very similar performance on ngram probability prediction. These results held over these features for the current and previous words as well.

Ultimately, although the untrained LM prediction of fMRI data might be partially explained by the LM's prediction of word features, further explanation is required to explain the degree of performance in fMRI prediction. Finally, possible reasons are explored to reconcile some of the more contradictory results.

CHAPTER 2

LITERATURE REVIEW

2.1 High Level Methodological Overview

Language is composed of a sequence of words. Using methods from linguistics, we can assign word features to each word, such as part-of-speech. Using human subjects and methods from neurophysiology, we can assign an fMRI response to each word. Using methods from machine learning, we can assign a language model (LM) response to each word. My research and the research that I refer to here revolves around using simple prediction models, such as linear regression, logistic regression, and support vector machines, to predict some of these values from the others.

To be clear, the untrained/trained LM are not themselves predicting anything in the context of this research. The crucial key to this research is to contrast the simplicity of the predictive model with the complexity of the predictors and the target. This allows us to argue that the prediction outcome is the result of a direct connection between the predictor data and the target data, since the prediction model itself does a minimal amount of the heavy lifting, so to speak.[1].

[1] For example, if a simple linear regression model was able to predict some particular brain activity from the sole metric of estimated word frequency, we could conclude that that particular brain activity may involve the computation of word frequency or something very close to word frequency. However, if we did the same experiment but used a deep learning model to predict brain activity directly, we would not be able to reach this conclusion. Instead, it could be that the deep learning model internally produces an estimate of POS and that part of the brain is processing a particular part of speech that is otherwise independent of word frequency. Thus, a complex prediction model may be more predictive, but it comes at a high cost to interpretability.

It is important to note that the prediction of fMRI data is not an ultimate goal in itself, as there are no apparent practical applications for a reliable predictor or simulator of fMRI data.

This method of using task-optimized neural model activations to predict fMRI data was initially applied in visual processing (2014) [11], and later in other perceptual tasks such as rat whisker navigation (2017) [12], auditory processing (2018) [13], and language processing (2021) [1]. The results have been interpreted to mean that the internal representations that the brain uses in task cognition are closely related to the internal representations of the neural models. If that interpretation holds here, then perhaps the fact that untrained models predict fMRI data indicates that the random weights in the models happen to produce representation related to internal representations of the stimulus itself (e.g., a word's POS) or to computational processes involved in the cognition of stimulus (e.g., the time it takes to compute word's POS).

2.2 Schrimpf's research

The Schrimpf study, published in the Proceedings of the National Academy of Sciences, compared a wide variety of neural language models in their prediction of brain responses [1]. In summary, the prediction of the fMRI data involves computing the cross-validated Pearson correlation coefficients between the predicted and actual fMRI values for each voxel of each participant, aggregating the results, and normalizing the result by dividing by the prediction noise ceiling of the dataset. The trained models were out-of-the-box pretrained models. Untrained networks were initialized with the default initialization scheme of the models, and the weights were never adjusted after initialization.

Schrimpf's experiments give good initial insight into the untrained networks. Many of the untrained models could predict brain responses far above chance.

Whatever representation these untrained models are developing that allows them to predict fMRI seems to have something to do with language because their scores are predictive of other metrics. For example,

- The untrained model's fMRI predictiveness correlated with their trained counterpart's fMRI predictiveness[2].
- the untrained model's fMRI predictiveness correlated with their scores for next word predictiveness[3].

They also found that the large size of the feature space alone could not explain the results; a model that randomly projected words (independent of context) into vector spaces with the size of the GPT-2-XL activations scored only 15% of the untrained GPT-2-XL model. Model performance was not due to overfitting on common words or word sequences, because the statistical overlap between ngrams in train/test sets was low. Finally, to test whether the linear fitting methodology was inflating the scores, the model predictability with regression was, in turn, predictive of an alternative metric, Regression Depth Median (which seems to be conceptually similar to the k-nearest neighbor), which does not rely on an assumption of linear separability.

The crucial difference between my research and Schrimpf's is the use of linguistic features for interpretability. His research is centered on predicting one black box (the brain) from another black box (a neural network) and attributing the differences in score to the differences in the model architecture. [14] For example, they found that models which did not take the word's context into account (such as word2vec) failed to significantly predict fMRI response, which suggests that the integration of words with their context is necessary for predicting the brain response. However,

[2] They found that training resulted in an average of 53% increase in score.

[3] Next word prediction was not performed using the untrained models raw predictions (which, of course, are random), but by training a linear readout from the network's outputs (a similar prediction procedure as for fMRI predictions.)

the conclusions that can be drawn from this methodology are limited [14]. For example, it does not seem capable of generating explanations for performance differences between models of high prediction, since it gives limited insight into how the LMs are modeling the brain data.

2.2.1 Hosseini

Hosseini *et al.* published a preprint (coauthored by Schrimpf), which included a subexperiment that explored whether the performance of the untrained model was sensitive to initialization choices [5].

They manually reinitialized the 12 sets of weights in the first transformer block of an untrained GPT-2 model with a simple Gaussian distribution $\mathcal{N}(0, 0.02)$, and found that the model could no longer significantly predict the fMRI data. They theorized that similar to how the brain's architecture has been selected by evolutionary processes over time, the model's architecture, hyperparameters, and initialization strategies have been progressively selected by people creating language models based on their performance on language modeling. In other words, even though the particular values of random weights aren't conditioned to linguistic data, the choices in the model architecture are. For example, they choose which distributions to initialize from, how to scale initial weights, how many layers the model will use, how many units per layer, etc.

This result is not too unexpected; it has been reported in applications of neural networks with random weights that networks are sensitive to initialization choices [15, 16].

2.2.2 Pasquiou

The research of Pasquiou *et al.* focused on examining which regions of the brain were predicted by trained vs untrained LMs [2]. Like Schrimpf *et al.*, they also reported that untrained models could predict fMRI data better than chance.

The most interesting finding was that the primary regions of the brain where the untrained models are most predictive are not the primary areas where training improves the prediction of fMRI. The methodology involved the creation of a *training gain* model by subtracting the predictivity of each model trained from the untrained predictivity, over each voxel. They examined the top 10% of the voxels for trained and untrained models of the LSTM, GPT-2, and BERT architectures. They reported that there was a 79% overlap across the three untrained models, a 75% overlap across the *training gain* models, but these two overlap regions only overlapped each other by 18%. In other words, the primary regions of the brain where untrained models are most predictive are not the primary areas where training is improving the prediction of fMRI.

Contrary to Schrimpf, which reported that neither the trained nor the untrained GloVe model performed well, they found that the less complex untrained models (LSTM and GloVe) outperformed the more complex untrained transformer models (GPT-2 and BERT), but for trained models, the opposite is true.

Examining the models, this study used 768 units for GloVe for comparison with GPT-2, while the GloVe model in Schrimpf contained 300 units.

2.3 Baek's research: single-neuron representations

Baek *et al.* published a study in Nature by Baek on untrained neural *vision* models that I have replicated for untrained neural *language* models [3]. It is known that fully trained neural models (in general) contain some individual activations that are directly interpretable as representing high-level features [4], but Baek found that even untrained *vision* models contain neurons directly interpretable as representing face vs non-face.

Baek *et al.* retrieved 400 images for each of six image classes. These images fed through the model and the activations of each neuron were z-scored on all images. Using 200 training images of each category, each neuron was classified as face-selective if its mean activations on face images was significantly higher than the mean of the other five classes, using a two-sided rank sum test $(P < 0.001)$

Baek then characterized the degree of the neuron's face-selectivity index (FSI), defined by the difference of mean activations in face and non-face classes normalized by the standard deviation of the activations [17][4].:

$$\text{FSI} = \frac{\mu_+ - \mu_-}{\sqrt{(\sigma_+^2 + \sigma_-^2)/2}}$$

Baek demonstrated with a number of tests that these activations not only correlate with face vs. non-face on holdout data but act face-selective in other ways as well. For example, they retained their selectivity when evaluated on images from other datasets and images generated with generative neural networks. Face-selective activations showed a higher response to images where local features

[4] Just to sate curiosity (though its not particularly significant): this research had a few odd mathematical decisions. For example, the random division by two in the FSI definition and using a two-sided test for a clearly one-sided comparison. These appear to be historical artifacts because they continue to occur (without comment) as deep in the chain of references as I investigated.

were disrupted compared to images where global features were disrupted, showing that they were not just extrapolating from simpler local features, such as nose-shape. Furthermore, they trained a face vs. non-face image SVM classifier on the 1.1% of the face-selective neurons in the final layer, and it performed similarly well to an SVM that was trained on the entire final layer of activations. With some training, they demonstrated that the FSI of face-selective activations decreased when the network was trained on a dataset without faces and increased when the training data was appended with more face images. (One experiment was particularly convincing, but it does not appear to have an obvious linguistic analog. Baek trained used generative adversarial neural networks to generate images that maximized face-selective weights, resulting in images that visually appear to be face-like images.)

CHAPTER 3

GENERAL METHODOLOGY

3.1 Linguistic Features

The base corpus for word features is “CoNLL2012 shared task data based on OntoNotes 5.0”, and is hereafter referred to as the OntoNotes corpus, and specifically the Hugging Face implementation of the OntoNotes corpus[1] which used the data processed with the official scripts from a Mendeley repository [18].. This is a manually-annotated pretokenized corpus that includes 2.6M English words with Penn-Treebank-style POS tags for words and parse trees for sentences [10][2].

Additionally, the features for the function versus content distinction, word order, tree depth, and unigram/bigram/trigram frequencies were computed. The ngram frequencies are calculated across the entire OntoNotes corpus without smoothing. Unigram frequency should be independent of context, Bigram frequencies depend on the previous word only for context, and trigram frequencies depend on the previous two words only for context.

The original POS feature, which I label POS-51, contains 51 “Penn Treebank style” POS tags in this version of the corpus [10]. The corpus documentation was outdated, detailing only 36 standard tags of the Penn Treebank, which was used for an earlier version of the data set. The POS-51 tags were simplified to produce POS-12 with the 12 class ‘Universal’ tagset [19][3]using NLTK’s

[1] OntoNotes Corpus: huggingface.co/datasets/conll2012_ontonotesv5

[2] Though not utilized in this research, this corpus also contains datasets for Arabic and Chinese as well as semantic features for named entities, semantic role labels, and co-reference.

[3] POS-12 tag descriptions: github.com/slavpetrov/universal-pos-tags

map_tag function. Finally, since Baek's research defined feature selectivity using a preference to one of six classes, I manually simplified POS-12. For comparison to experiment 2, POS was further simplified into seven class POS-7 with the classes of noun, verb, adjective, determiner (E.g., the, a, this), adposition (prepositions and postpositions. For example, in, of, for, with), and an 'other' class which is discarded for that experiment. See section 4.2 for more details.

3.1.1 Why these features

Cognitive behavior in general is sensitive to statistical aspects of language[4]. These features were chosen because they are simple, easily accessible, and well studied in language cognition. Simplicity is desirable because one would not expect untrained networks to model more complex features if they cannot model the simpler features on which they are based. For example, I would not expect a model to distinguish between the semantic roles that relate noun to verb if they cannot distinguish between nouns and verbs. Ngram frequencies were chosen because ngrams are one of the most standard control models of next-word prediction, and see use as a control model in next-word prediction tasks. Finally, part of speech was chosen because it is one of the simplest ways to model syntactic relationships. (LMs tend to be better (more natural?) models of syntax than of semantics. For example, during training, LMs learn syntactic relationships before semantic relationships, and simple LMs tend to reliably capture many basic syntactic relationships but tend to struggle with even basic semantic relationships.)

[4] This is a fascinating area. In the simplest cases, observable cognitive behavior is sensitive to (context-independent) word frequency. For written language, the more common a word is, the faster it is recognized, the faster it can be read aloud, and the faster lexical decisions can be made [20, 21, 22, 23, 24]. Similarly, in the audio realm, subjects are more likely to recognize more common words immersed in noise [25, 26]. These results are generalized when the context is incorporated (such as the frequency of the word being used as that part of speech). The results further generalize to the predicted word probability that is output from language models. For an interesting review, see Jurafsky [27].

3.2 Neural Network Response (Activation Retrieval)

All models are of the GPT-2 architecture. The GPT-2 architecture appeared to be the best candidate for my experiments because Schrimpf reported that GPT-2 outperforms all others and because it is more cognitively plausible than the other transformer models because it is unidirectional [1]. Furthermore, I wanted to examine the results of Hosseini [5], who also used a GPT-2 model.

All experiments were run with a total of 21 model instances, all utilizing the GPT-2 architecture:

- XL-Trained (GPT-2-XL, 49 layers, 1.5B parameters)
- XL-Untrained (GPT-2-XL, 49 layers, 1.5B parameters)
- GPT-Trained (GPT-2, 13 layers, 117M parameters)
- $9\times$GPT-Untrained (GPT-2, 13 layers, 117M parameters)
- $9\times$GPT-Gaussian (GPT-2, 13 layers, 117M parameters)

This is a total of 21 models and 345 layers. All models were implemented using the Hugging Face transformer library [6]. The trained models use the pre-trained weights from this library, and the untrained models were created using the default initialization from this library. Nine models were used for GPT-Untrained to check if the results with trained models were relatively consistent. For each of the GPT-Untrained models created, a GPT-Gaussian model was created by reinitializing 12 weights in the first layer with values pulled from a Gaussian $\mathcal{N}(0, 0.02)$. [5]

Extracting activations from the model yields 768 activations per layer/word from the base GPT-2 models and 1600 for the GPT-2-XL models. The input tokens are truncated to 512, and thus the extracted activations are outputs of each block that correspond to the activations of the 512th token.

The final layer also has the softmax function applied. Finally, the 0th layer consists of the values that are fed into the first block. Crucially, this means that the 0th layer is context-independent, it is dependent only on the random input embedding for the current word.

Schrimpf truncated the input length of the tokens to a maximum of 512 tokens, likely to control for the input length between the various sizes of language models that he used in his study. There was one adjustment that I made for my experiment; I discarded the activations that were calculated with fewer than 512 total input tokens. (i.e., the first 386 words.) There does not appear to be any reason why the nth activation of one positional input would have a direct relation to the nth activation of another. Additionally, discarding the short inputs eliminates unnecessary complexities, such as checking if the distribution of the categorical data is different in shorter inputs.

3.2.1 Activation Retrieval

To keep my experiments as comparable as possible to Schrimpf's set-up, the process for my activation retrieval mostly reused the retrieval from Schrimpf's code. The first 500 sentences of the OntoNotes corpus "training split" were used to generate activations for each word per model per layer. The values for the first 386 words (22 sentences) were discarded due to insufficient context length, resulting in 7958 samples.

One challenge I had to overcome was that the corpus was tokenized using a different standard from the GPT-2 tokenizer. To overcome this, I needed to manually adjust the tokenization. (see Appendix section A.3 for details)

3.3 Prediction of fMRI

Schrimpf's published code was used to calculate each model's ability to predict fMRI data, reported at the layer-by-layer level, against the three fMRI datasets used in their original paper, Blank2014 [7], Fedorenko2016 [8], and Pereira2018 [9]. Of these datasets, Schrimpf *et al.* primarily focused on the results of the Pereira2018 dataset, which was approximately 3 orders of magnitude larger than the other two. I was unable to get the code to run without errors on the Fedorenko2016 dataset. In the end, my code returned the exact scores reported for all layers in the pre-trained GPT-2 and GPT-2-xl model on the Blank2014 dataset except for the 0th layer. It returned almost the exact Pereira2018 results, with discrepancies in the second decimal place.

However, though my results on the Blank2014 dataset seem to replicate the published results of Schrimpf *et al.*, there was a concerning inconstancy that lead me to set these results aside. The activations of the Gaussian models are approximately uniform from the second layer onward to the second to last layer on the Pereira2018 dataset. (As will be shown in later chapters, this pattern is consistent across all predictions of linguistic features as well.) This contrasts heavily with their prediction of Blank2014 where these Gaussian's Brain Scores vary significantly layer by layer. Additionally, more recent github versions of their code yield different scores on Blank2014. Finally, it does not seem necessary to focus on Blank2014 results, as the datasets were smaller than Pereira2018 by a factor of 1000, and Schrimpf *et al.* mainly focused on Pereira2018 in their publication. This result is available in Appendix B, Figure B.1.

Therefore, I only used the Pereira2018 dataset for my experiments. Pereira2018 was also the primary dataset mentioned in Schrimpf as it was approximately three orders of magnitude larger than Blank2014 and Fedorenko2016. The stimulus included a selection of 168 passages that span a diverse range of topics with 3 or 4 sentences per passage for a total of 627 sentences. [9] The fMRI data come from two experiments with 9 and 6 subjects (10 unique), and responses were averaged over 3 repetitions of each stimulus. The pre-processing and the exclusion of voxels from outside language-specific brain regions resulted in responses for total of 13553 voxels (average of 1355 voxels per participant)[5].

3.3.1 Interpretation: fMRI measures task complexity

fMRI uses an MRI machine to measure the blood oxygen level-dependent signal (BOLD). This signal is elicited from changes in oxygenated blood flow to areas of the brain in response to the neural activity that occurs in those areas. Crucially, this signal scales with the complexity of the task, rather than with the quantity of muscular energy required for the task, for example. For example, consider a finger tapping task. If you tap the fingers of your right hand on a flat surface sequentially 2-3-4-5, this will increase blood flow to the left motor cortex of your brain. An MRI machine would record a spike in the BOLD signal in the left motor cortex, with a time delay of approximately 4-6 seconds. Instead, if you tap in a more complex order, such as 3–5–4–2, you will generate a greater increase in blood flow to this cortex, resulting in a significantly higher spike in the BOLD signal, despite the fact that the muscular energy requirements of the two tasks are approximately equal. [28]

[5] I initially intended to use the results of all three datasets, so I wrote in a way that was general to the 3 datasets. However, the brain datasets are very diverse. For example, Blank2014 used an audio stimulus with 1 brain response per word, while Pereira2018 is a visual stimulus presented in visual format with 1 response per sentence. But even with that caveat, I am glossing over some major technicalities with respect to the brain datasets and their prediction. The single most relevant technicality is that the brain response is not actually recorded per word but per word grouping, where the grouping was a sentence in Pereira2018 but a 2 second interval in Blank2014. This means that the fMRI data is not predicted from the LM activations per word, but instead the averages of the fMRI responses are predicted from the averages of the LM activations. Ultimately, my research is focused on the prediction of word features, so it is easiest to understand my research by thinking of the fMRI prediction as a per-word calculation, and leave these technicalities to research more oriented on the neurophysiological aspect. See Schrimpf's Supplementary Information Appendix for more details on the preprocessing and prediction of fMRI datasets [1].

CHAPTER 4

EXPERIMENTS

fMRI Prediction and Reproduction of Schrimpf's Results

The aggregated correlation coefficients of all models on the Pereira2018 fMRI dataset are shown in item 4.1. For the trained models, the values reproduced Schrimpf's results to 5 significant figures. The only exception was for the 0th layer, though all his models and all my models scored near zero for that layer, and I couldn't determine why they differ. My untrained models differed because they are different random initializations but scored similarly to the reported untrained model. Throughout all my results, the untrained initializations behaved very similarly to each other, suggesting that the results are robost (insensitive to the randomization seed).

My Gaussian random initialization did not behave like Schrimpf's. Whereas his resulted in what appeared to just be a significantly less predictive model, my Gaussian initialization model produced activations that were relatively uniform after the 0th layer. The values in the model essentially got 'stuck' with little significant change from layer to layer. [1] In the present investigation, the Gaussian models here serve as a control[2].

All of my experiments are centered on explaining the data in this result. For example, it would be interesting to find features that the model cannot predict, because that would indicate that they

[1] This seems to be because the skip connections in the GPT-2 architecture skip over a normalization layer, which results in the non-normalized input being summed with a normalized output.

[2] The Gaussian models were originally intended for an experiment that was cut due to time constraints.

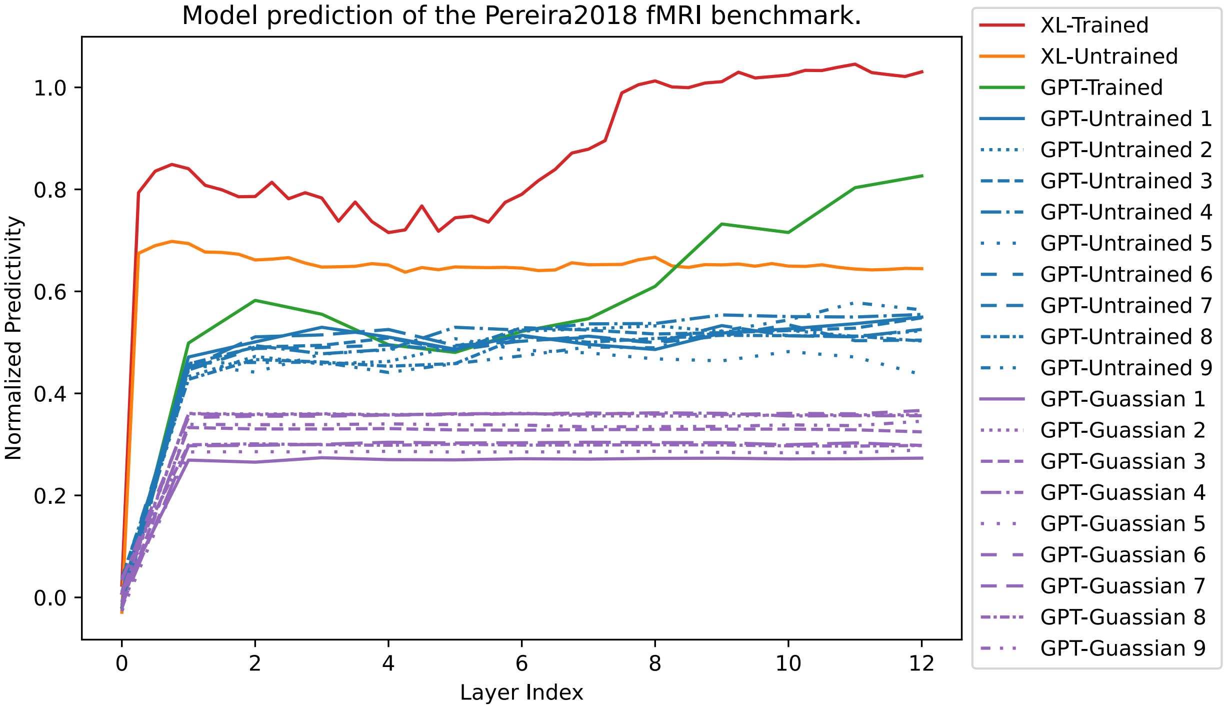

Figure 4.1: Model prediction of the Pereira2018 fMRI benchmark. The Key Takeaways: (1) The 0th layer of all models did performed near chance. (2) All models become significantly predictive of the Pereira2018 data on the first layer. (3) For the untrained models, this predictivity remains fairly constant over the rest of the layers with small fluctuations. (4) For the Gaussian models, this predictivity remains uniform over the rest of the layers. (5) For the trained models, after the first 2-3 layers, there is a slight downward trend for the first half of the values, followed by a large improvement in the second half of the layers. The layer index of the 49-layer XL models are scaled down to match the 13 layers of the non-XL models.

are not making a significant contribution towards predicting the fMRI data. On the opposite end, it would also be interesting to find features such that the models variance in prediction of those features mirrors the variance of prediction of the fMRI data. Experiment 1 does this by directly mirroring the methodology for fMRI prediction, using a linear combination of model activations on a per-layer basis to predict linguistic features. In contrast to the linear combination methodology, experiment 2 uses Becks methodology to try to characterize the activations of individual neurons as being selective of linguistic features.

4.1 Experiment 1 *Predicting word features from each layer of activations.*

This first experiment directly mirrors Schrimpf's methodology for fMRI prediction and uses a linear combination of a layer of model activations to predict linguistic features.

4.1.1 Method: Predicting word features from layer activations

At its core, each predictive model consists of predicting a single word feature from a single layer of activations over the set of all stimulus words, where each layer consists of 768 values for the 'GPT' models and 1600 predictors for the 'XL' models. The OntoNotes stimulus data consists of 7,958 words, so each predictive model consists of 7,958 data instances, one instance for each word. Each stimulus word corresponds to a total of 27 unique targets, nine word features for each for the current, previous, and next words. Since there are 345 total layers across all neural models, each stimulus word corresponds to 345 sets of predictors. Therefore, this experiment consists of a total

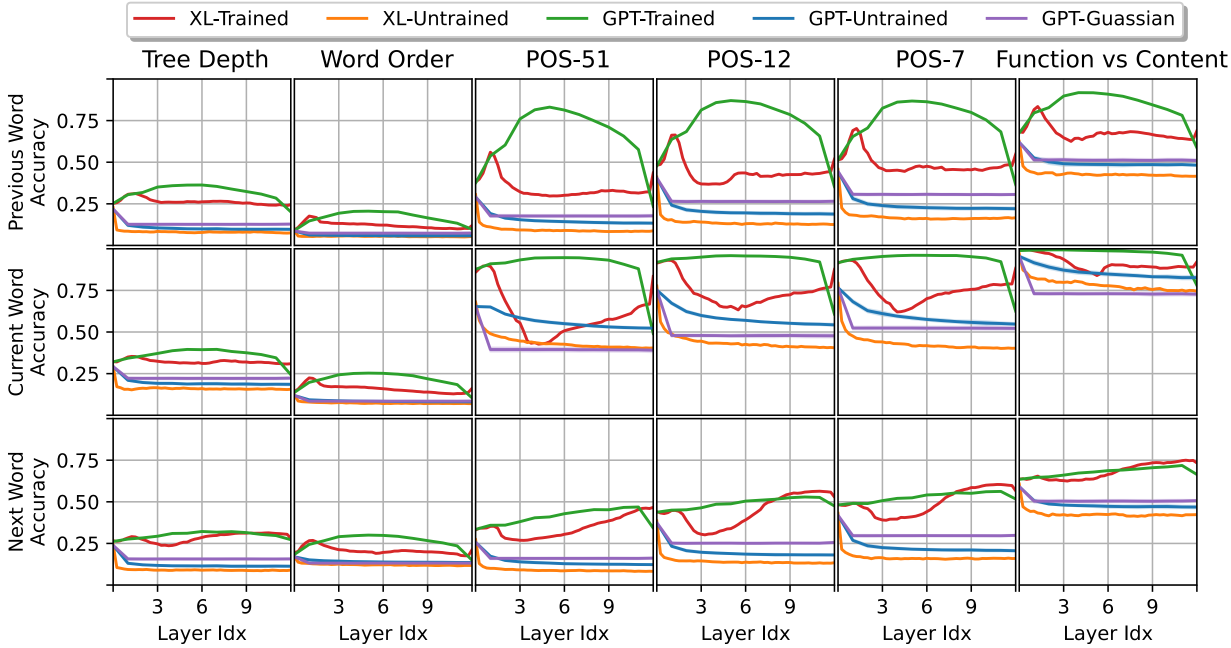

Figure 4.2: Results for POS prediction with logistic regression. Reported is the classification accuracy between the actual values and the values predicted from multiclass logistic regression with stratified 5-fold cross validation. Targets are the product of the 3 different partitions for POS classes over the previous, current, and next words values. POS-51 has 51 classes, POS-12 has 12 classes, POS-7 has 7 classes. For the 9 GPT-2-Untrained and 9 GPT-2-Gaussian models, the solid line represents their mean and the shading represents their standard deviation over the 9 models. The 49 layers of the XL models are scaled down to the 13 layers of the base models. In contrast to the ngram predictions, XL-Trained under-performs its base variant in the middle layers of the model, but, like with the ngram predictions, they are similar at the start and end layers. Like the ngram predictions, the *untrained* model underperforms its base variant GPT-Untrained on all targets, which is the opposite of the brain-score results.

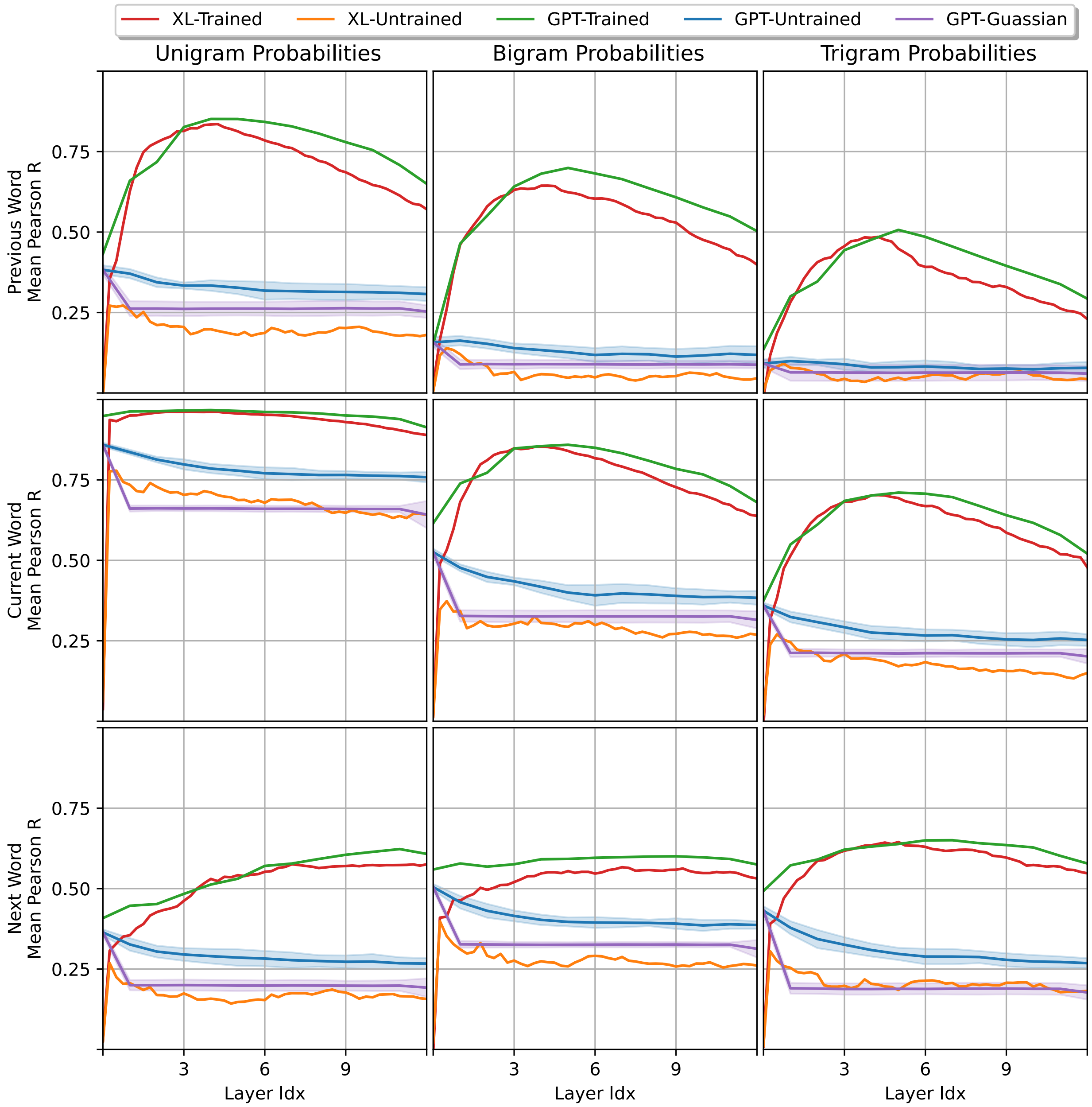

Figure 4.3: Results for ngram prediction with linear regression. Pearson correlation coefficient between the actual values and the values predicted from ordinary linear regression with 5-fold cross-validation. The reported coefficient is the mean of each fold. Targets are the product of the unigram, bigram, and trigram frequencies (calculated over the full OntoNotes corpus) over the previous, current, and next words values. For the 9 GPT-2-Untrained and 9 GPT-2-Gaussian models, the solid line represents their mean and the shading represents their standard deviation over the 9 models. Except for the 0th layer, *trained* XL model and its base perform similarly across most targets. under-performs its base variant in the middle layers of the model, but like with the ngram predictions, they are similar at the start and end layers. Like the POS predictions, the *untrained* model under-performs its base variant GPT-Untrained on all targets, which is the opposite of the brain-score results.

of

$$345 \times 6 = 2070$$

predictive model, one for each layer/target.

Three of the targets were continuous:

1. unigram frequency (i.e., word frequency)
2. bigram frequency
3. trigram frequency

For continuous targets, ordinary linear regression was used to generate predictions for each instance using 5-fold cross-validation. For each fold, the Pearson correlation coefficient between the held-out and predicted values was calculated. Finally, the mean of these coefficients is calculated and taken to be the performance metric of the predictive model and reported as the mean Pearson's R for the combination of layer / target.

Six of the targets were categorical:

1. POS-51 (51 classes)[3]
2. POS-12 (12 classes)
3. POS-7 (7 classes)

For categorical targets, multitarget logistic regression was used to generate predictions for each instance using *stratified* 5-fold cross-validation. The performance metric for categorical targets is categorical accuracy.

[3] Note that the POS categories are hierarchical. (E.g., each category in POS-51 maps to exactly one category in POS-7.)

4.1.2 Results: Predicting word features from layer activations

The results of POS prediction are shown in Figure 4.2 and the results of ngram prediction are shown in Figure 4.3. For each of the GPT2-Untrained and Gaussian models, the mean of the 9 models is plotted, and their shaded area represents the standard deviation.

The results show that all POS targets could be classified above chance in the 0th layer. This accuracy should not be too unexpected; the 0th layer is dependent only on the current word, and part of speech should be inferable for many words (E.g., ‘the’ usually a determiner). Perhaps, it is not surprising that the POS of the previous and next words can partially be guessed from the current word (E.g., ‘the’ often follows punctuation, verb, or noun, but not often an adjective. and is often followed by determiner or adjective). However, it is surprising that, for example, the previous word’s part of speech isn’t better predicted in the first layer given that it does have dependence on the previous word (as well as all other word that fit in the context length.

The results of the 0th layer on the POS targets contrast with the results of the fMRI prediction, where the 0th layer could not predict the fMRI data above chance for any model. This implies that knowledge of these POS targets is not, by itself, predictive of fMRI data, which is consistent with the results of Schrimpf, which found that context-independent LMs could not predict fMRI above chance.

Interestingly, except for the GPT-Trained, the 0th layer could not predict the ngram targets, above chance. This could suggest that the fMRI prediction could be somewhat dependent on the task of next-word prediction. However, since the 0th layer of GPT-Trained can predict the ngram targets above chance, but cannot predict fMRI, ngram prediction must also not be sufficient to predict the brain data.

For all untrained models, on all targets, the performance on the layers after the 0th layer decayed asymptotically.

The 0th layer, which is independent of the surrounding words, behaved differently. For the untrained GPT2 models, the performance of the 0th layer was higher than in the performance of the first layer, on all targets with the unique exception of the current word's POS-51. This also held for the 0th layer of the untrained XL models for the POS prediction, but was not true for n-gram protection, where the 0th layer of the XL models scored near chance.

The XL-Trained model's 0th layer exhibited the same oddity as the XL-Untrained models, where the 0th layer scored significantly on POS prediction but near chance for ngram prediction.

In ngram prediction, the trained models generally performed similar to each other and far outperformed the untrained models.

The trained model generally performed above the other models for all layers on all targets. For most targets, the scores tended to increase over the layers until a peak and then decreased for the rest of the layers. This suggests that the trained model tends to develop information about these word features through the first layers, and then discards that information in the later layers. Most interestingly, relative to the current word prediction, the peak tends to occur earlier in the previous word feature targets, and later on the next word feature targets. This suggests that previous-word information is processed (or most relevant) earlier in the network, and next-word information is produced or processed later in the network.

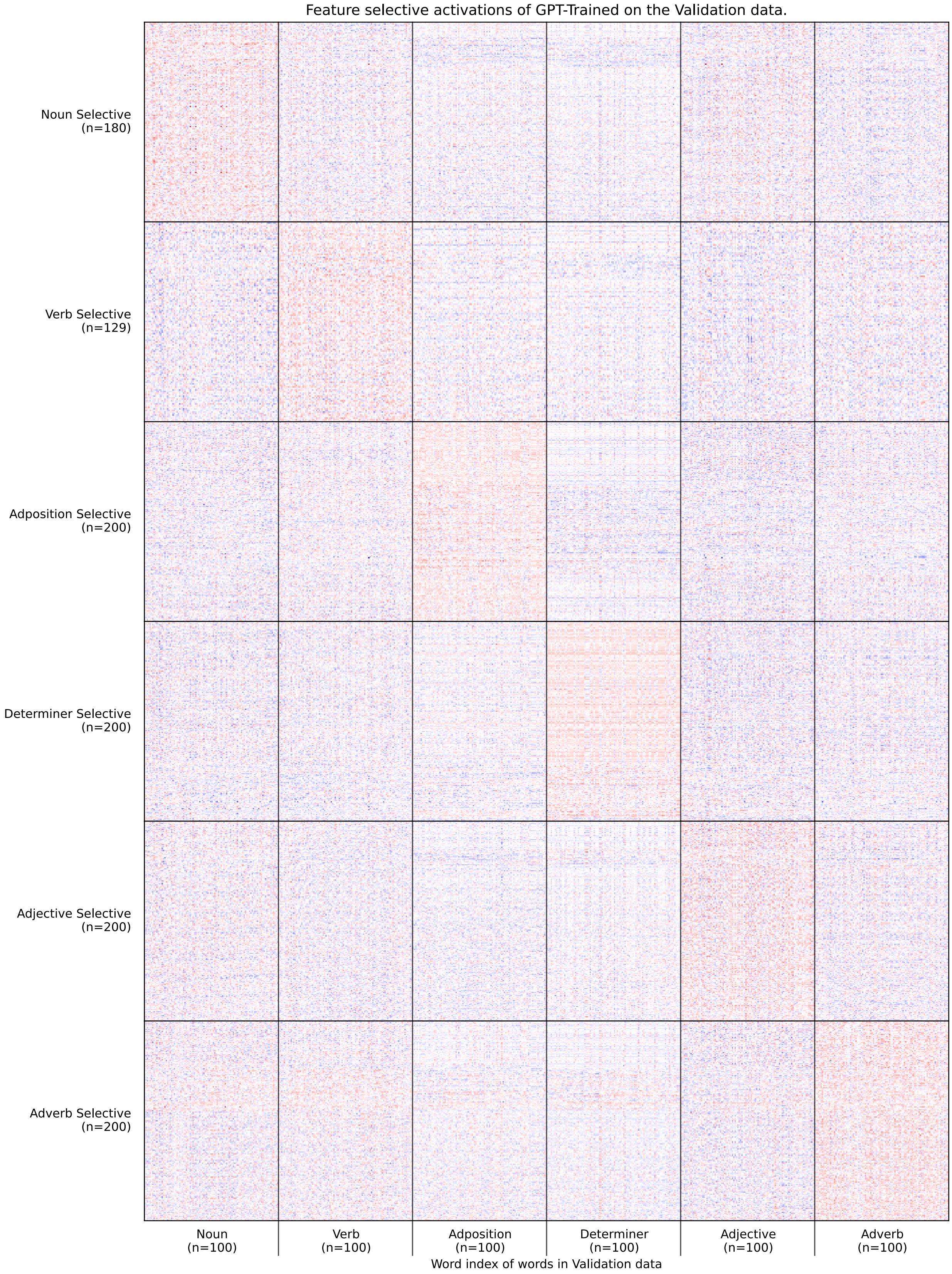

Figure 4.4: A visual representation of GPT-Trained's POS-selective neurons activations over the validation data. The units are arbitrary with white for near 0 activations, more red for greater than 0, more blue for less than 0. This confirm that this methodology, trained models do have neurons that can be directly interpreted as POS-selective selective, and serves as an exemplar for what would be desired in the untrained model.

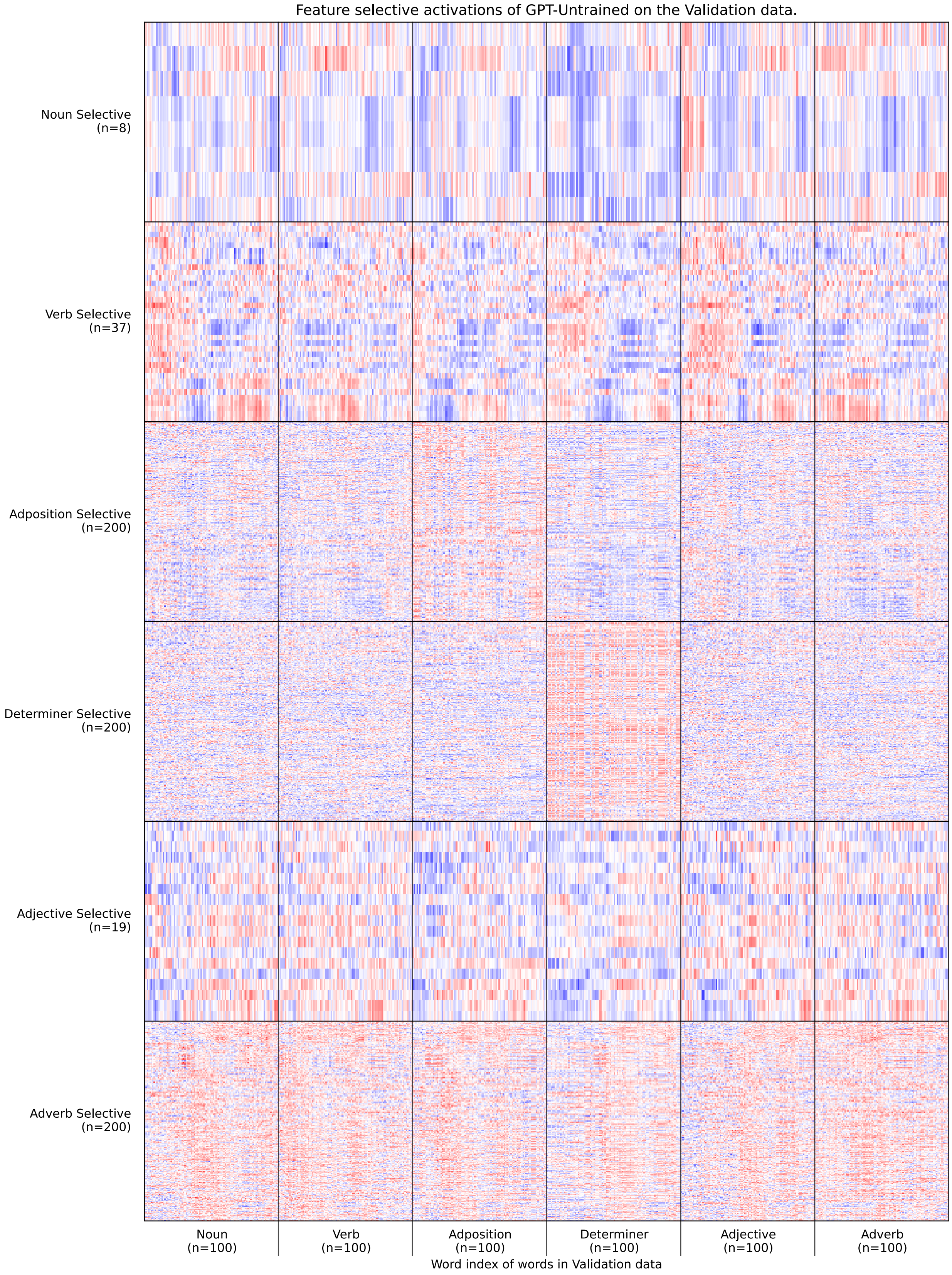

Figure 4.5: A visual representation of GPT-Untrained's POS-selective neurons activations over the validation data. The units are arbitrary with white for near 0 activations, more red for greater than 0, more blue for less than 0. Note that the determiner selective activations and adposition selective neurons are clearly more active on the validation data, whereas its not very clear if the POS-selectively holds on the validation data others are not.

Table 4.1: Prevalence of feature selective neurons per feature/model/layer.

		Noun	Verb	Adposition	Determiner	Adjective	Adverb
Model	Layer						
GPT-Trained	0	0.26%	1.56%	6.38%	8.59%	1.17%	1.04%
	1	0.78%	1.3%	3.78%	8.72%	1.95%	2.86%
	2	1.04%	1.3%	4.56%	9.77%	1.82%	3.39%
	3	1.43%	1.17%	4.17%	8.2%	2.47%	2.99%
	4	2.08%	1.3%	3.26%	6.25%	2.34%	3.26%
	5	1.69%	1.69%	2.99%	5.86%	2.6%	2.99%
	6	2.47%	1.56%	3.26%	4.95%	2.86%	2.73%
	7	2.6%	1.95%	3.26%	4.95%	3.26%	3.78%
	8	2.86%	1.95%	3.52%	4.43%	2.21%	3.91%
	9	2.34%	1.04%	4.17%	4.95%	2.47%	3.65%
	10	2.47%	1.3%	5.08%	5.21%	1.95%	3.39%
	11	2.21%	1.04%	5.6%	4.69%	1.04%	2.34%
	12	1.43%	1.17%	5.99%	4.17%	1.04%	3.12%
GPT-Untrained	0	0%	0.14%	4.85%	20.38%	0.06%	0.23%
	1	0.01%	0.13%	3.05%	16.57%	0.04%	0.82%
	2	0%	0.03%	1.88%	14.92%	0.06%	1.07%
	3	0.03%	0.03%	1.43%	13.63%	0%	1.07%
	4	0%	0.04%	1.35%	12.73%	0.03%	1.3%
	5	0.03%	0.04%	1.27%	11.86%	0%	0.94%
	6	0%	0.09%	0.98%	11.39%	0%	0.88%
	7	0%	0%	0.77%	11.24%	0.03%	1.1%
	8	0%	0.03%	0.84%	10.87%	0.03%	0.93%
	9	0%	0.04%	0.84%	10.27%	0.03%	1.01%
	10	0.01%	0.04%	0.84%	9.74%	0%	0.91%
	11	0.01%	0.03%	0.78%	9.62%	0.03%	1.04%
	12	0.01%	0.03%	0.62%	9.26%	0.03%	0.85%
GPT-Guassian	0	0%	0.14%	4.85%	20.38%	0.06%	0.23%
	1	0%	0%	0.2%	7.51%	0.01%	0.27%
	2	0%	0%	0.2%	7.47%	0.01%	0.33%
	3	0%	0%	0.23%	7.48%	0.01%	0.32%
	4	0%	0%	0.2%	7.38%	0.01%	0.33%
	5	0%	0%	0.2%	7.44%	0.01%	0.33%
	6	0%	0%	0.22%	7.31%	0.01%	0.32%
	7	0%	0%	0.23%	7.26%	0.01%	0.32%
	8	0%	0%	0.25%	7.22%	0.01%	0.32%
	9	0%	0%	0.25%	7.35%	0.01%	0.32%
	10	0%	0%	0.25%	7.26%	0.01%	0.29%
	11	0%	0%	0.23%	7.44%	0.01%	0.32%
	12	0%	0%	0.27%	7.41%	0.01%	0.32%

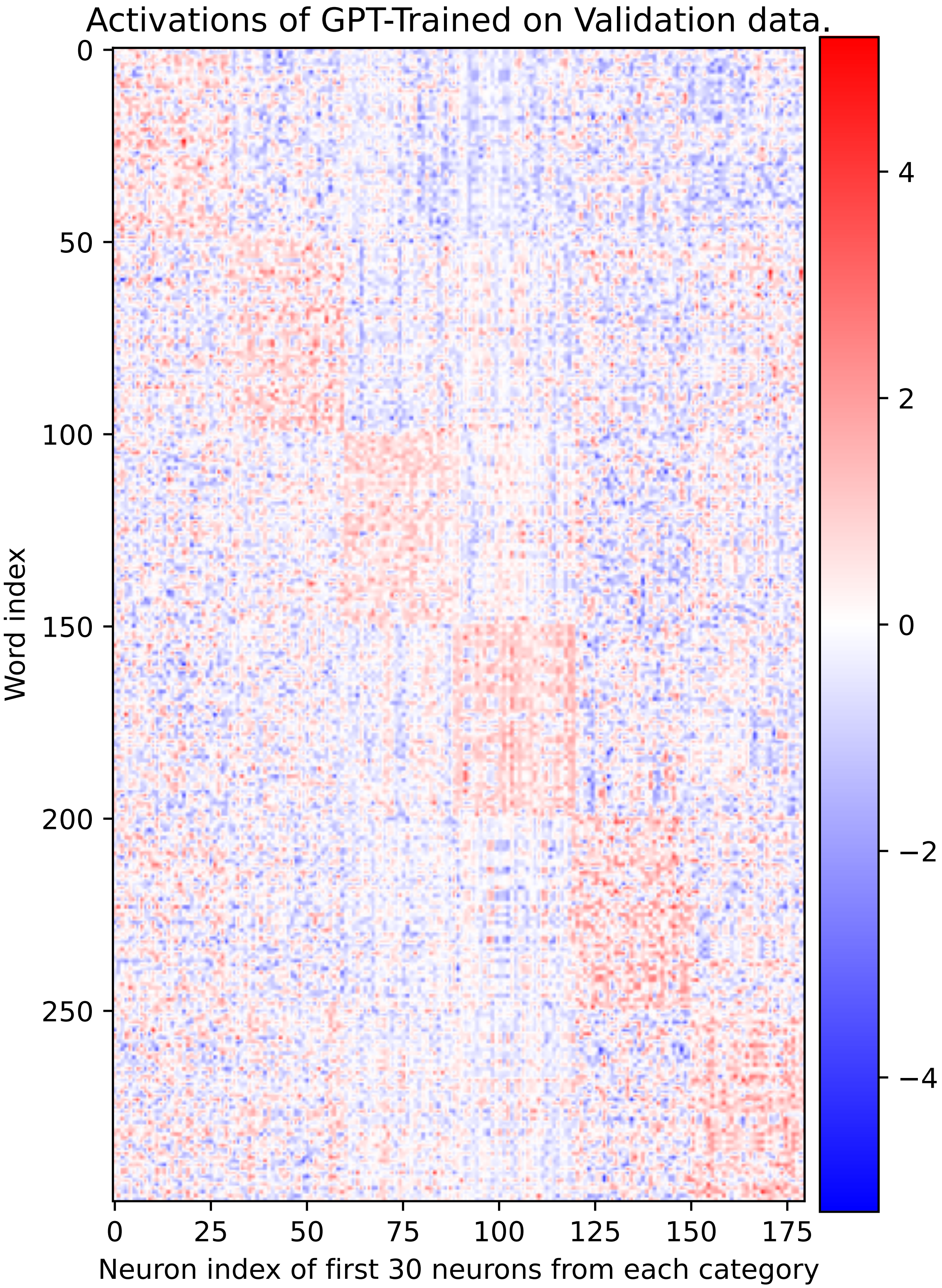

Figure 4.6: Activations of feature-selective neurons on held out data. This figure shows that the determination of selectivity does not overfit the training data. The neuron index corresponds to the first 30 neurons that were determined to be feature-selective for the 6 categories. (e.g., first 30 noun-selective neurons, the first 30 verb-selective neurons...) The word index corresponds to the first 50 words of each category (e.g., first 50 nouns, then first 50 verbs...)

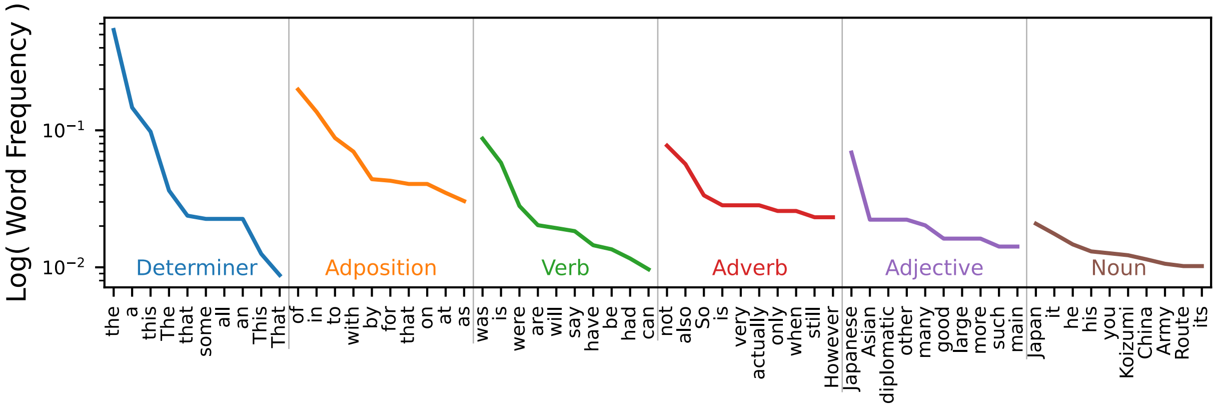

Figure 4.7: Word frequencies of the 10 most common words in each of the 6 POS categories over the first 500 sentences of the corpus. These words accounted for more than 10% of their categories occurrences: 'the' accounts for 54% of determiners, 'of' accounts for 20% of adpositions, and 'in' accounts for 14% of adpositions. These categories are also the categories for which the untrained models contained a significant number of neurons that were feature selective. This could mean that neurons are selecting for these common words, rather than for linguistic properties of the category.

4.2 Experiment 2 ***Examining individual neuron's POS selectivity.***

Whereas Experiment 1 explored the relationship of a word feature to a *layer* of activations, this experiment explored the relationship of a word feature to a *single neuron* activations. To follow Baek's statistical method as closely as possible, I implemented my experiment with the following specifications:

- z-score neurons
- 6 classes
- 200 training examples per class
- two-sided rank-sum test with a confidence threshold of $P < 0.001$

4.2.1 Method: Predicting word features from layer activations

First, the data instances were partitioned by their POS-12 class, and the noun and pronoun partitions were combined. Then, all but the six classes with the largest populations in the full corpus were discarded, resulting in the following classes.

1. Noun (includes Pronouns)
2. Verb
3. Adposition
4. Determiner
5. Adjective
6. Adverb

Since the smallest partition had 387 instances, each partition was randomly sampled for 200 training instances, 100 validation instances, and 75 training instances for a total of 2250 words of data. The activations of each unit were normalized by z-scoreing over all 2250 words. For each neuron, the class with the highest mean activations was compared with the others using a two-sided rank sum test ($P < 0.001$) to determine whether the difference in the mean of two distributions is statistically significant. If the difference was significant against all other categories, then that neuron was categorized as selective for that class.

4.2.2 Results

The percentage of neurons that are feature selective in each layer is shown in Table 4.1. The reported values for GPT-Untrained and GPT-Guassian are the means of the nine models. Interestingly, in the untrained network, the number of feature-selective neurons also appears to decrease asymptotically in subsequent layers. This seems to mirror the predictiveness of the layers for POS categories.

For example, perhaps the neurons are instead selective of determiners because ‘the’ accounts for 54% of determiners. To explore this, the word frequencies for the most common words of each category are shown in Figure 4.7.

Baek’s experiment used a network with 5 layers of convolution and 2 regular feedforward layers over an extremely high-dimensional image input. The LMs that I am using are much deeper and more sophisticated with a low-dimensional sequence of words input.

CHAPTER 5

DISCUSSION

Experiment 1 focused on predicting word features from a linear combination of neuron activations in each layer, while Experiment 2 essentially predicted linguistic features from individual neurons.

5.1 What did my research find about the trained models?

The trained LM activations were found to be able to model word features, as expected. For trained models on the layer-based prediction:

- Relative to current word targets, the peak predictiveness layer for the base GPT-2 model occurred in a deeper layer for next word targets and in a shallower layer for previous word targets. This could suggest hierarchical processing with layer depth.
- For all word frequency targets, the predictiveness of the trained GPT-2-XL model tracked the trained base model.
- In contrast, for all POS targets, the predictiveness of the trained GPT-2-XL model differed significantly in the middle layers, showing a minimum predictiveness about 1/3 of the way through the layers, with improvement through the second half of the layers. This was the only major result that seems to conform to the expectations of the fMRI prediction data.

For trained models in the experiment on single-neuron feature selectivity for 6 classes of POS:

- Trained models contained POS-selective neurons, which was expected.
- In contrast to the untrained models, the number of POS-selective neurons did not appear to asymptotically decay over progressive layers.
- The number of neurons for each part of speech was fairly constant for each category in deeper layers, about 2% of neurons in each category.

The last result is fairly interesting, as one would expect that the quantity of selective neurons may track how frequently that category appears.

5.2 What did my research find about the untrained models?

Untrained models were found to be predictive of linguistic targets providing evidence for my theory that their prediction of fMRI data was by virtue for their prediction of linguistic features. For untrained models:

- They were reasonably predictive of POS
- They were very predictive of the current words word frequency
- They were not very predictive of higher orders of ngram frequencies for the previous word.
- The untrained GPT-2-XL models were less predictive of all linguistic targets than the untrained base GPT-2.
- Across linguistic targets, the performance of the untrained model appeared to decrease asymptotically over progressive layers, approaching a constant value.
- Across all models, this asymptotic decrease was also observed in single-neuron feature selectivity.

In general, the performance of the untrained model seems to directly contradict their performance in predicting fMRI data. In contrast to the prediction of the linguistic targets:

- The untrained GPT-2-XL models were more predictive of fMRI targets than the base GPT-2.
- The performance of the untrained models on the fMRI data was constant or slightly increased over progressive layers.

5.2.1 Critically evaluate assumptions and limitations in light of the research.

This research was carried out under the assumption that the activations of the untrained model model fMRI data because these activations model linguistic features. It could be that the activations are modeling features not directly related to the brain's processing of the language itself, and instead the untrained model just happens to be a naturally good feature extractor for those nonlinguistic features, such as audio features caused by certain syllables.

However, there is an alternative way to model this that consistently explains the observed results. It is rather interesting, though not tested.

1. Each feature has an explicit representation and an embedded representation.
2. The explicit representation is the representation that is predictive of the feature directly (perhaps via a linearly separable model).
3. The embedded representation is the representation that predicts the fMRI data (perhaps through a linearly separable model).
4. The model architecture is such that on average in a random computation, the computation deconstructs the explicit representation into its embedded form.
5. Likewise, the model architecture is such that, on average in a random computation, the computation reconstructs the embedded representation into its explicit form.
6. Finally, assume that the networks are such that in net, they preserve information so that an increase in one is balanced by a decrease in the other.

This would result in the following explanation for the base GPT-2 models: the 0th layer embedding of the input (an explicit word) happens to randomly extract features with a more explicit representation than an embedded representation. Then, as the layers progress, the explicit representations are converted into implicit representations, and vice versa. This leads to an asymptotic approach towards equilibrium. This approach results in progressively more embedded representation which increases fMRI predictiveness, and vice versa, the approach results in progressively less explicit representation and hence progressively less predictiveness of word features.

This theory contains many assumptions to validate. Perhaps one way to test this would be to replace n activations in one layer directly with explicit feature information and observe the prediction of that feature downstream. If this theory is accurate, then one should be able to fit an equilibrium equation to solve for the previous quantity of explicitly represented features, and the previous quantity of embedded information, both in terms of n. And if this prediction holds, then this method of explicit feature injection could be a more direct way to evaluate the model's sensitivity to that feature's embedded information.

5.3 Future research: Where to go from here?

Given the complexity and number of choices involved in each of the data sets, and the inherent opaqueness of both the brain and neural networks, there seems to be a vast number of ways to continue research into understanding how the untrained models can predict neural data.

For example, different linguistic features could be examined. The experiments can be scaled up in a number of ways, such as using more data points from the OntoNotes.

Future research along this line could look at other word-level features such as named entities, semantic role labels, and co-reference features.

As noted in the literature, Pasquiou [2] found opposite results to Schrimpf in untrained networks, so it seems that one would want to examine the fMRI pre-processing and / or prediction procedures more closely.

A more practical direction for the single neuron study would be automating the selection of features to examine, in line with the style of a recent study published by OpenAI in which they automated the process of interpreting individual neuron activations with the latest version of Chat-GPT [4]. This would be especially useful for comparing different architectures.

CHAPTER 6

CONCLUSION

My research explored the language predictiveness of untrained neural language models to better understand how their activations were predictive of neural data. Using the code of Schrimpf *et al.*, I reproduced their results for the GPT-2 and GPT-2-XL LMs on the Blank2014 and Pereira2018 datasets, and described discrepancies that occurred when analysing the Blank2014 dataset. As expected, pretrianed model activations were able to predict word features above chance, especially POS and ngram features. In the context of the fMRI prediction methodology, untrained LM activations were found to be predictive of many word features, especially POS and Ngram frequencies, although less so than their pretrained counterparts. Untrained LMs were also found to be more predictive of word features for the next word than previous word targets. When looking at the performance of individual layers, whereas untrained LM predictions of fMRI data appear to increase asymptotically with increasing depth, their predictions of word features appear to decrease asymptotically with increasing depth.

The larger untrained model was significantly more predictive of fMRI data than the smaller untrained model, but significantly less predictive of all linguistic targets. For trained models, the results were similar but more nuanced. Relative to the smaller model, the larger trained model again outperformed on fMRI prediction, again unperformed on POS, word order, and syntactic tree depth predictions, but had very similar performance on ngram probability prediction. These results held over these features for the current and previous words as well.

Ultimately, although the untrained LM prediction of fMRI data might be partially explained by the LM's prediction of word features, further explanation is required to explain the degree of performance in fMRI prediction. Finally, possible reasons are explored to reconcile some of the more contradictory results.

REFERENCES

[1] M. Schrimpf *et al.*, “The neural architecture of language: Integrative modeling converges on predictive processing”, *Proceedings of the National Academy of Sciences*, vol. 118, no. 45, e2105646118, Nov. 2021.

[2] A. Pasquiou, Y. Lakretz, J. Hale, B. Thirion, and C. Pallier, “Neural Language Models are not Born Equal to Fit Brain Data, but Training Helps”, *ArXiv e-prints*, Jul. 7, 2022. eprint: 2207.03380.

[3] S. Baek, M. Song, J. Jang, G. Kim, and S.-B. Paik, “Face detection in untrained deep neural networks”, *Nature Communications*, vol. 12, no. 7328, pp. 1–15, Dec. 2021.

[4] S. Bills *et al.*, *Language models can explain neurons in language models*, https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, 2023.

[5] E. A. Hosseini, M. Schrimpf, Y. Zhang, S. Bowman, N. Zaslavsky, and E. Fedorenko, “Artificial neural network language models align neurally and behaviorally with humans even after a developmentally realistic amount of training”, *bioRxiv*, p. 2022.10.04.510681, Oct. 5, 2022. eprint: 2022.10.04.510681.

[6] T. Wolf *et al.*, “Transformers: State-of-the-art natural language processing”, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[7] I. Blank, N. Kanwisher, and E. Fedorenko, “A functional dissociation between language and multiple-demand systems revealed in patterns of BOLD signal fluctuations”, *Journal of Neurophysiology*, Sep. 1, 2014.

[8] E. Fedorenko *et al.*, “Neural correlate of the construction of sentence meaning”, *Proceedings of the National Academy of Sciences*, vol. 113, no. 41, E6256–E6262, Oct. 11, 2016.

[9] F. Pereira *et al.*, “Toward a universal decoder of linguistic meaning from brain activation”, *Nature Communications*, vol. 9, no. 963, pp. 1–13, Mar. 6, 2018.

[10] S. Pradhan *et al.*, “Towards robust linguistic analysis using OntoNotes”, in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 143–152.

[11] D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, “Performance-optimized hierarchical models predict neural responses in higher visual cor-

tex", *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8619–8624, Jun. 10, 2014.

[12] C. Zhuang, J. Kubilius, M. J. Hartmann, and D. L. Yamins, "Toward goal-driven neural network models for the rodent whisker-trigeminal system", in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[13] A. J. E. Kell, D. L. K. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott, "A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy", *Neuron*, vol. 98, no. 3, pp. 630–64 416, May 2, 2018. eprint: 29681533.

[14] J. T. Hale, L. Campanelli, J. Li, S. Bhattasali, C. Pallier, and J. R. Brennan, "Neurocomputational Models of Language Processing", *Annu. Rev. Linguist.*, vol. 8, no. 1, pp. 427–446, Jan. 2022.

[15] J. Frankle and M. Carbin, "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks", *OpenReview*, Dec. 21, 2018.

[16] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, and M. Rastegari, "What's Hidden in a Randomly Weighted Neural Network?", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 13, 2020, pp. 11 890–11 899.

[17] P. L. Aparicio, E. B. Issa, and J. J. DiCarlo, "Neurophysiological Organization of the Middle Face Patch in Macaque Inferior Temporal Cortex", *Journal of Neuroscience*, vol. 36, no. 50, p. 12 729, Dec. 12, 2016.

[18] F. Xavier, "ontonotes-conll2012", *Mendeley Data*, vol. 2, Mar. 14, 2022.

[19] S. Petrov, D. Das, and R. McDonald, "A Universal Part-of-Speech Tagset", *ArXiv e-prints*, Apr. 11, 2011. eprint: 1104.2086.

[20] D. H. Howes and R. L. Solomon, "Visual duration threshold as a function of word-probability", *J. Exp. Psychol.*, vol. 41, no. 6, pp. 401–410, Jun. 1951.

[21] K. I. Forster and S. M. Chambers, "Lexical access and naming time", *Journal of Verbal Learning and Verbal Behavior*, vol. 12, no. 6, pp. 627–635, 1973.

[22] H. Rubenstein, S. S. Lewis, and M. A. Rubenstein, "Homographic entries in the internal lexicon: Effects of systematicity and relative frequency of meanings", *Journal of Verbal Learning and Verbal Behavior*, vol. 10, no. 1, pp. 57–62, Feb. 1971.

[23] C. P. Whaley, “Word—nonword classification time”, *Journal of Verbal Learning and Verbal Behavior*, vol. 17, no. 2, pp. 143–154, Apr. 1978.

[24] J. I. Chumbley and D. A. Balota, “A word’s meaning affects the decision in lexical decision”, *Memory & Cognition*, vol. 12, no. 6, pp. 590–606, Nov. 1984.

[25] D. Howes, “On the relation between the probability of a word as an association and in general linguistic usage”, *J. Abnorm. Psychol.*, vol. 54, no. 1, pp. 75–85, Jan. 1957.

[26] H. B. Savin, *Word-Frequency Effect and Errors in the Perception of Speech*, [Online; accessed 17. Nov. 2019], 1963.

[27] D. Jurafsky, *Probabilistic Modeling in Psycholinguistics: Linguistic Comprehension and Production*, [Online; accessed 7. Nov. 2019], 2006.

[28] C. Gerloff, B. Corwell, R. Chen, M. Hallett, and L. G. Cohen, “The role of the human motor cortex in the control of complex and simple finger movement sequences.”, *Brain*, vol. 121, no. 9, pp. 1695–1709, Sep. 1, 1998.

Appendices

APPENDIX A

A.1 Method: Predicting fMRI from layers activations.

At the lowest level, consider the prediction for the values a particular voxel, for a particular subject, for a particular layer. Finally, consider a particular time value and consider just the first fold of cross-validation, partitioning the words into a training and held-out set. At this resolution, each word in the stimulus corresponds to one layer of activations and one fMRI data point. First, the time dimension is eliminated using the first fold to determine the optimal time point (the BOLD signal has a 4-8ish second delay). Then, the held-out data is set aside so that it does not influence the choice in time. A linear regression model is trained over the activations of all words in the training set to predict their corresponding fMRI values. The Pearson correlation coefficient is calculated between these predicted values and the actual values. This is repeated over all values in the time dimension, and the time that maximizes this prediction is chosen as the target time delay. Next, for all folds, the heldout values are predicted from the training values at that time delay, and the Pearson correlation coefficient is calculated. This results in a single correlation coefficient, representative of that voxel/subject/layer. This coefficient is normalized by dividing by the noise ceiling of that data set, which is taken to represent the maximum correlation coefficient that is possible: 0.32 (Pereira), 0.17 (Fedorenko), and 0.20 (Blank)[1] This normalized Pearson correlation coefficient is taken to represent a normalized predictivity for that combination of voxel, subject, and layer.

Finally, the coefficients are aggregated over voxels by discarding the lowest scoring 90% of voxels (which was reported to be a standard procedure) and computing the median over the remainder. Then these scores were aggregated over subjects, taking the median value over all subjects. This results in the final metric, Brain Score: an aggregated normalized Pearson correlation coefficient reported for each layer of the model. (When comparing a large number of models, the scores can be further aggregated over layers by selecting the value of the maximally performing layer only, which results in a single Brain Score per model.)

Schrimpf reported the standard deviation in the participant score as the error. [2]

A.2 Package Versioning details for replicating the results of Schrimpf's Research.

Schrimpf's code included the published results of the models on the Pereira2018, Fedorenko2016, and Blank2014 fMRI datasets, as well as their Python environment. Their code was unable to reproduce these results run as-is at the most current version; I had to manually downgrade the version of their code along with five of the code's dependencies, 'brainio', 'brainio_base', 'brainio_collection', and 'result_caching'. Additionally, I had to use the same version of tensorflow, pytorch, and cuda, otherwise the calculated Brain Score was different. I also attempted to run the code on the same

[1] The ceiling estimation involves using sub-sampling with a procedure that involves predicting the biological readings of one participant from a subset of the others, and extrapolating for the highest ceiling possible.

[2] Please see Schrimpf's supplemental appendix for more detail on these calculations[1].

versions of Ubuntu as the authors with a fresh install of Ubuntu 16.04, but got the same results as running the code on Ubuntu 22.04. Because the original version of CUDA was not technically supported on my operating system (Ubuntu 22.04), I also attempted to run the code on a fresh install of the same OS as the authors (Ubuntu 16.04), but the results did not vary. In the end, my code replicated almost the exact Pereira2018 results as Schrimpf published, with variation in the second decimal place, replicated the exact reported for all layer Brain Scores for the pretrained GPT-2 and GPT-2-XL model on the Blank2014 dataset except for the 0th layer, and would not run the Fedorenko2016 dataset without error.

A.3 Reconciling Tokenization Issues

There were two major issues with the pretokenization process. First, the GPT-2 tokenizer also tokenizes spaces and appends them to the following word. Secondly, the OntoNotes corpus is pretokenized, with its features reported at the token level, but the GPT-2 tokenizer has a different tokenization scheme. For example:

⟨ *Don't replace the can marked "Green Paint".* ⟩

tokenizes into

⟨ *(Don 't) (re place) (the) (can) (marked) (" Green) (Paint " .)* ⟩

where parentheses denote each space-delimited "word". The activation for each word is defined as the activations on the final token of each word. Additionally, it appends a space character to the front of the first token of each word, except for the first word in every sentence. The activations for each word are computed as the activations for the final token of the word. The OntoNotes corpus was already tokenized as

⟨ *Do n't replace the can marked " Green Paint " .* ⟩

If fed directly into the GPT-2 tokenizer, this would result in

⟨ *(Do) (n ' t) (re place) (the) (can) (marked) (") (Green) (Paint) (") (.)* ⟩

The GPT-2 tokenizer further tokenizes ⟨ *replace* ⟩ into ⟨ *re place* ⟩. The OntoNotes tokenization tokenizes a contraction into ⟨ *base n't* ⟩ whereas the GPT-2 tokenizer uses ⟨ *basen* + *'t* ⟩. The GPT-2 tokenizer incorrectly places spaces between tokens such as in front of ⟨ *Green* ⟩, the period, the end quote, and the ⟨ *n* ⟩ in ⟨ *don't* ⟩.

Since the corpus was annotated, this was solved in the following way: I moved the ⟨ *n* ⟩ in all ⟨ *n't* ⟩ to the previous word. And finally, I flagged all tokens to have a space in front of them, unless they met one of the following conditions:

- token was already a tokenized symbol like ⟨ *'t 's* ⟩
- token was a opening or closing parenthesis, or end of sentence tag
- token followed by a start parenthesis POS or start quote POS tag.
- token followed a hyphen POS tag
- token was the first token in the sentence.

The most major difference between my activation retrieval and Schrimpf's was that whereas my activation retrieval returned one set of layer activation's per token, Schrimpf's code returned one set of layer activation's per (space delimited) word. This was done by assigning to the word, the set of layer activation's per token. In my previous example, the last word is

⟨ *Paint".* ⟩

which is composed of 3 tokens:

⟨ *Paint " .* ⟩

So Schrimpf's code would assign to the full 3 token word, only the models activations corresponding to the period. This seems to be a necessary step for modeling an audio stimulus with a text-based model where punctuation has no explicit form but instead is applied through modifications of the nearby explicit words. But given that I was assigning activations to analyze the token level “word features”, I just assigned the token activations directly to the tokens. Therefore, in the context of my experiments, a word is defined as a token in the OntoNotes corpus, rather than space delineation as in the fMRI prediction in Schrimpf.

APPENDIX B

OTHER RESULTS

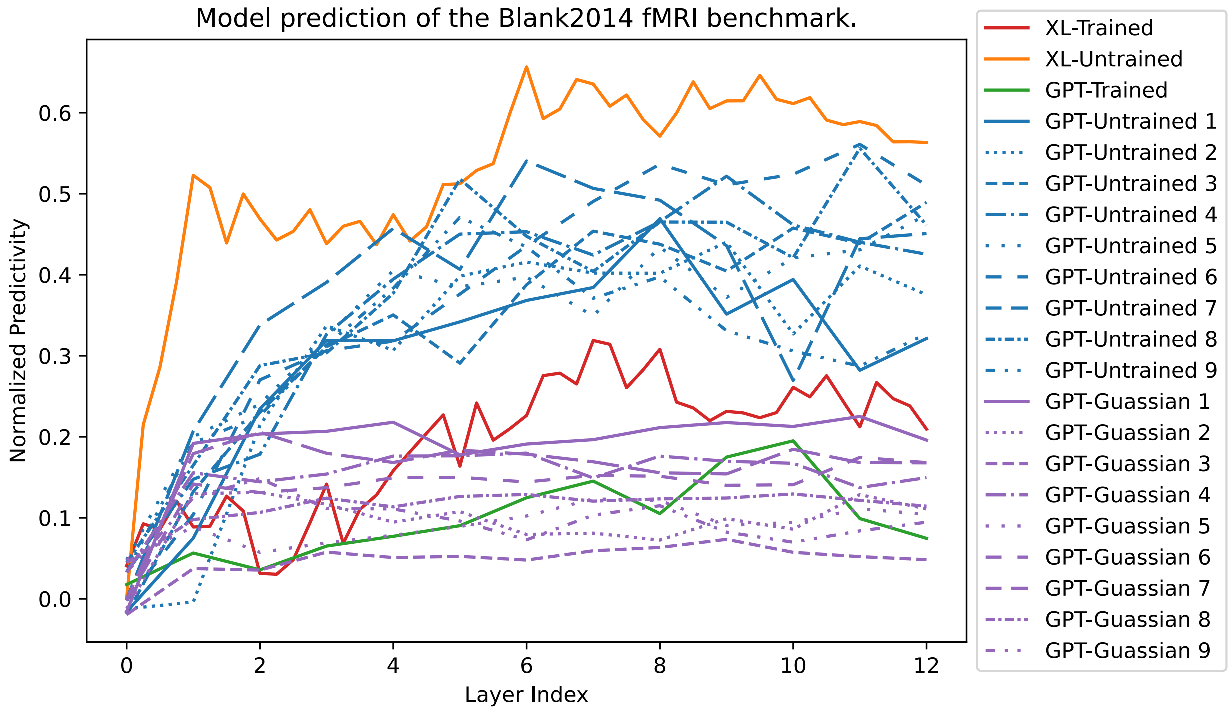

Figure B.1: Model prediction of the Blank2014 dataset. The scores of the trained models reproduced Schrimpf's published results. The layer index of the 49-layer XL models is scaled to align with the index of the 13 layers of the non-XL models. However, these results are suspect: the performance of the layers of the Gaussian models should be nearly uniform after the 0th layer, as is the case in all other results of this research. (This is because the models activations themselves are nearly uniform after the 0th layer.)

APPENDIX C

RAW DATA AND STATISTICS

Table C.1: Categorical statistics for POS-51 (n=2.74M)

POS	Proportion
NN	12.13
IN	10.32
DT	8.44
NNP	7.50
JJ	5.41
NNS	5.04
.	4.65
XX	4.62
,	4.23
RB	4.11
PRP	4.07
VB	3.35
VBD	3.12
CC	2.85
VBZ	2.28
VBP	1.97
VBN	1.92
CD	1.88
VBG	1.57
TO	1.41
MD	1.16
PRP$	1.06
HYPH	0.67
UH	0.62
POS	0.61
”	0.58
“	0.57
WP	0.46
WDT	0.45
RP	0.38
:	0.37
WRB	0.37
JJR	0.26
NNPS	0.26
VERB	0.19
$	0.17
EX	0.16
JJS	0.14
RBR	0.13
-RRB-	0.13
-LRB-	0.13
PDT	0.09
RBS	0.05
FW	0.04
NFP	0.02
SYM	0.02
WP$	0.01
LS	0.01
ADD	0.01
AFX	0.00
*	0.00

Table C.2: Categorical statistics for POS-12 (n=2.74M)

POS	Proportion
NOUN	24.93
VERB	15.38
.	10.82
ADP	10.32
DET	9.14
X	6.20
ADJ	5.81
PRON	5.61
ADV	4.67
CONJ	2.85
PRT	2.40
NUM	1.88

Table C.3: Categorical statistics for POS-7 (n=2.74M)

POS	Proportion
Noun	30.54
X	24.14
Verb	15.38
Adposition	10.32
Determiner	9.14
Adjective	5.81
Adverb	4.67

Table C.4: Categorical statistics for POS-6 (n=2.08M)

POS	Proportion
Noun	40.25
Verb	20.27
Adposition	13.61
Determiner	12.05
Adjective	7.66
Adverb	6.15