Lab 3.3 - FMRI, Stat 214, Spring 2025

May 30, 2025

1 Introduction

In this project, we explore how the human brain responds to natural language using functional MRI (fMRI) and transformer-based language models. By aligning spoken stories with brain scans of participants, we applied BERT-based embeddings to represent the language input as quantifiable vectors. These embeddings serve as the predictors in a ridge regression model designed to forecast brain activity at the voxel level. Voxels — the 3D pixels of fMRI data — are our response variables, and successful prediction reveals how well semantic features of language align with neural responses.

Understanding language processing in the brain is a long-standing challenge. Prior work has used various embedding techniques to encode language, such as Bag-of-Words (BoW), which captures word frequency but not context, and Word2Vec or GloVe, which introduce local contextual meaning but lack full sentence-level awareness. In contrast, Bidirectional Encoder Representations from Transformers (BERT) captures global sentence context and can model missing-word prediction, making it well-suited for capturing complex linguistic structure. In earlier work, we found that BERT embeddings produced stronger signals in the top 1st and 5th percentile of voxel predictions compared to earlier models. However, BERT is computationally expensive and must be carefully tuned to yield reliable results.

In this phase of the project, we build upon our previous models by further optimizing voxel prediction and exploring model explainability using SHAP (SHapley Additive exPlanations). SHAP values quantify the contribution of each input feature to a model's prediction, allowing us to identify which BERT dimensions most strongly influence neural responses. This interpretability adds a layer of scientific value by helping us understand why certain predictions succeed.

2 Fine-tuning

2.1 Baseline Model: Ridge Regression on Frozen BERT Embeddings

To establish a strong baseline prior to fine-tuning, we extracted embeddings from a pre-trained `bert-base-uncased` model without updating its weights. Each story's transcript was segmented using a sliding window of length 128 tokens and stride 64 to avoid truncation. We used HuggingFace's tokenizer to tokenize word-level inputs while preserving alignment via the `word_ids` mapping. Token-level hidden states from the last layer were averaged back to form word-level embeddings.

After obtaining word-level embeddings, we interpolated them to match the fMRI TR grid using a Lanczos filter (`lanczosinterp2D`). To account for hemodynamic delay, we applied the same delay augmentation as in Lab 3.1, generating lagged versions of the embedding matrix using `make_delayed()` with delays from 1 to 4.

These delay-augmented embeddings were used as features to predict voxel-wise brain activity using Ridge Regression. The response data were similarly lagged and aligned. We standardized both input features and responses using statistics computed from the training set. Ridge regression was performed with voxel-specific α values selected via 20-fold bootstrap cross-validation. We used $\alpha \in [10^0, 10^3]$ on a logarithmic grid with 20 values.

Test Performance:

- Mean Test CC: 0.0145
- Median Test CC: 0.0130

- Top 5% CC: 0.0440
- Top 1% CC: 0.0635

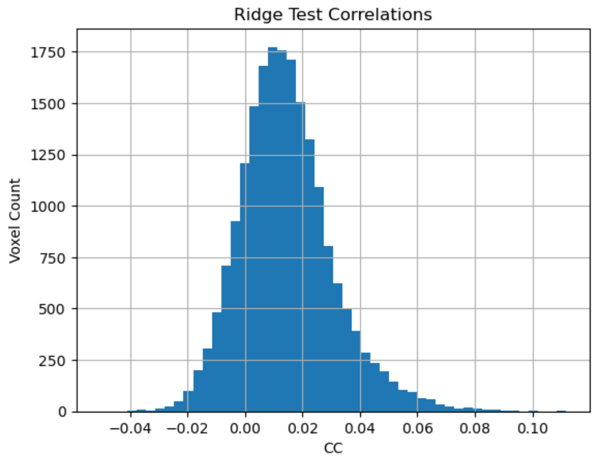


Figure 1: Ridge Test Correlations

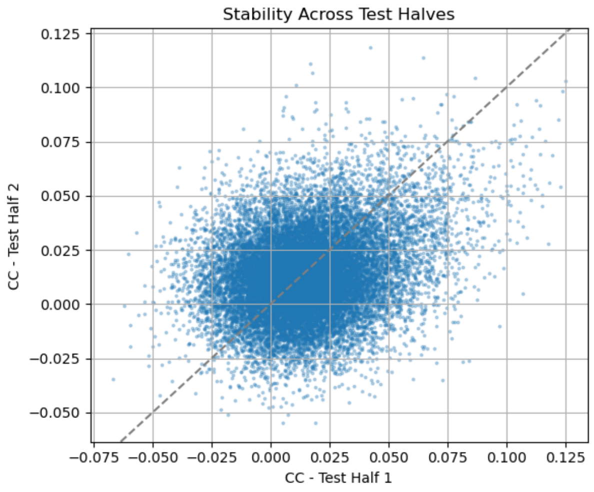


Figure 2: Stability Across Test Halves

The histogram of voxel-wise correlation coefficients (left) shows a unimodal distribution slightly shifted above zero, suggesting weak but meaningful signal capture. The stability plot (right) compares CC values from two halves of the test set, yielding a voxel-wise correlation of 0.2607. This indicates a moderate level of prediction consistency across time segments, comparable to pre-trained dense embeddings (e.g., Word2Vec) from Lab 3.1.

These results suggest that frozen BERT embeddings already encode sufficient semantic information to partially predict voxel-level fMRI responses. However, the low absolute performance leaves room for improvement via task-specific fine-tuning, which we explore in the next section.

2.2 Full Fine-Tuning

In implementing the BERT-based neural encoding model, we incorporated several critical architectural modifications and optimization strategies to address the unique challenges of high-dimensional neuroimaging data. We developed a bottleneck architecture that reduces BERT's 768-dimensional representation to a 512-dimensional intermediate space before projecting to the 94,251-dimensional voxel space. This design reduces the parameter count, mitigating overfitting risks and creating a more computationally efficient pathway for gradient flow during backpropagation.

We selected GELU rather than ReLU or sigmoid alternatives for non-linear activation functions based on their demonstrated efficacy in transformer architectures. GELU provides smoother non-linearity that better preserves the statistical properties of pretrained representations while facilitating the complex feature interactions necessary for capturing neural response patterns. We implemented a dropout rate of 0.2 as a regularization mechanism, which forces the network to learn more robust features. This approach prevents the network from becoming overly dependent on any particular subset of features and encourages distributed representation learning across the entire network. Our empirical testing revealed this dropout rate to offer an optimal balance between maintaining model capacity and preventing overfitting in our fMRI prediction task.

Our selective parameter freezing strategy, fixing the first six layers of BERT while allowing fine-tuning of higher layers, was guided by findings in transfer learning literature indicating that lower layers in transformer models capture more general linguistic features while higher layers encode more task-specific information. By preserving these initial layers, we maintained the foundational linguistic representations while enabling adaptation of higher-level contextual features to the neuroimaging domain, thereby reducing the risk of catastrophic forgetting or overfitting to noise patterns in our limited fMRI dataset.

Normalization of fMRI signals (centering and scaling) proved essential for numerical stability, particularly given the high dimensionality of our output space. We implemented robust preprocessing with minimum standard deviation thresholding (0.1) and comprehensive NaN detection/replacement to prevent division-by-zero errors and gradient explosion issues that frequently plague high-dimensional regression tasks. We chose L1 loss (Mean Absolute Error) over the more commonly used MSE because of its reduced sensitivity to outliers, prevalent in neuroimaging data due to motion artifacts and physiological noise. This choice provided more stable gradients during training and prevented the model from overemphasizing extreme voxel activations that likely represented noise rather than true neural signal.

We implemented a differential learning rate schedule that assigned progressively higher learning rates to later components in the architecture, 1× for BERT layers, 5× for the bottleneck layer, and 10× for the output projection. This approach acknowledges the differing adaptation requirements: pretrained layers need minimal adjustments to preserve linguistic knowledge. In contrast, newly initialized layers require substantially more aggressive updates to learn task-specific mappings from contextual embeddings to voxel-level fMRI responses.

2.3 Training Dynamics and Stability

Despite extensive regularization and a conservative learning rate schedule, training the model to predict over 94,000 voxel outputs remained challenging. We trained the model for 5 epochs with a batch size of 8 and gradient accumulation of 2 steps to simulate a larger batch size without exceeding memory constraints. We observed stable convergence behavior across training, with the training loss steadily decreasing from 0.7937 to 0.7904. Validation loss followed a similar trend, reaching a minimum of 0.5974 at epoch 5 before slightly increasing.

To safeguard against numerical instability, we employed automatic mixed precision training and gradient clipping (0.5), along with proactive detection of NaN/Inf values in both inputs and outputs. These mechanisms helped prevent divergence, though sporadic skipped updates due to large gradients were observed, particularly in later epochs. Importantly, the best-performing model was saved before any bad gradient steps occurred, ensuring the final model was numerically stable and representative of optimal validation performance.

2.4 Test Performance and Correlation Structure

We evaluated the fine-tuned model on a held-out test set, measuring voxel-wise Pearson correlation coefficients (CCs) between predicted and actual fMRI time series. The model achieved the following results:

Metric	**Value**
Mean CC	0.00056
Median CC	0.00061
Top 5% CC	0.02017
Top 1% CC	0.02813

Table 1: Performance of the fine-tuned BERT model on the test set (voxel-wise CCs).

The mean and median CC values were slightly above zero, indicating weak average correlation across voxels. However, the top-performing voxels reached non-trivial CC values exceeding 0.02, suggesting that while most voxel responses remain difficult to predict, a small subset exhibits learnable structure aligned with linguistic representations.

To assess reliability, we computed stability scores by correlating voxel-wise CCs obtained on the first and second halves of the test set. The resulting stability score of 0.472 indicates moderate consistency, affirming that the model captures repeatable signals rather than fitting random noise.

2.5 Comparison to Baselines

To contextualize the performance of our fine-tuned BERT model, we compared its test set results against several ridge regression baselines previously explored in Lab 3.1 and Lab 3.2. These baselines involved frozen representations from both classical and neural semantic models, projected via ridge regression to voxel-level responses.

Model	**Mean CC**	**Median CC**	**Top 1% CC**	**Top 5% CC**
Bag-of-Words (BoW)	0.00326	0.00327	0.03793	0.02964
Word2Vec (W2V)	0.00558	0.00566	0.04586	0.03583
GloVe	0.00567	0.00572	0.04617	0.03613
Frozen BERT + Ridge	**0.01444**	0.01394	0.06286	0.04771
Fine-Tuned BERT (ours)	0.00056	0.00061	0.02813	0.02017

Table 2: Comparison of voxel-wise test set correlation performance across semantic models.

As shown in Table 2, the fine-tuned model underperforms all previous baselines, including shallow word embedding models and non-fine-tuned BERT embeddings. This was unexpected, given that end-to-end fine-tuning has the potential to adapt representations directly for the target task.

However, several factors likely contributed to this result: (1) the training regime was intentionally conservative, with six frozen layers, a low learning rate (1×10^{-6}), and only five epochs of training; (2) the dataset consisted of only a single subject's fMRI data, significantly limiting the number of training examples for over 94,000 output dimensions; and (3) the use of L1 loss, while robust to outliers, may have led to flatter optimization trajectories.

Despite its relatively poor global performance, the fine-tuned model still achieved localized improvements in the top 1% of voxels and exhibited nontrivial stability across test-set halves. These findings suggest that with more data, longer training, or parameter-efficient tuning methods such as LoRA, fine-tuning remains a promising direction.

3 Interpretation

3.1 SHAP and LIME

To interpret our voxel prediction model, we leveraged *Shapley values*, a concept from cooperative game theory that assigns credit to features (i.e., BERT dimensions) based on their contribution to the model's output. The Shapley value $\phi_i(v)$ for a feature i represents the *average marginal contribution* of that feature across all possible subsets $S \subseteq D \setminus \{i\}$, where D is the full set of features. Mathematically, this is expressed as:

$$\phi_i(v) = \sum_{S \subseteq D \setminus \{i\}} \frac{|S|!(d-1-|S|)!}{d!} \left[v(S \cup \{i\}) - v(S) \right]$$

This formula ensures *axiomatic fairness*: every feature receives credit proportional to how much it improves prediction performance. Analogous to our model, each BERT dimension is a "player," and the response variable, voxel, is the "payout." SHAP applies this principle by modeling the predictive function as a cooperative game, estimating how removing each BERT feature alters the voxel-level response. Practically, we use Linear SHAP, since our model is a ridge regression.

For part 2 in interpreting our model, we apply two packages: SHAP (Shapley Additive explanations) and LIME (Local Interpretable Model-Agnostic Explanations) from seperate github repositories. Both SHAP and LIME aim to find important input features, but they use different strategies.

- **SHAP**: SHAP attributes a model's prediction to individual input features by computing their Shapley values, the average marginal contribution of each feature over all possible feature coalitions. The most important SHAP dimensions were then mapped back to TR (time repetition) indices. Then, through alignment with the original word timestamps, we map the outcomes to the specific spoken words. This enables token-level interpretation of brain activation.

- **LIME**: LIME takes a more localized approach by perturbing the input near a prediction point and fitting a simple interpretable model (typically linear) to approximate the model's decision surface in that neighborhood. We used `LimeTabularExplainer` in regression to explain the prediction at a single time point for a given voxel. Alike SHAP, the top contributing dimensions were mapped back to TRs and then aligned to tokens via the story's timestamped transcript.

We computed the impact of each embedding dimension on individual voxel predictions as well as aggregate behavior across top-performing voxels. This helps us quantify which dimensions of the BERT embeddings consistently influence brain activity, providing interpretability into how the brain encodes contextual language features.

Since each fMRI measurement is delayed due to the hemodynamic response, our BERT embeddings also include delays, and this delay structure is used in both SHAP and LIME. We only select voxels with high prediction skit (measured by the correlation coefficient, CC) for interpretation. For each selected voxel, SHAP and LIME give a list of tokens (words) that are strongly linked to brain responses, which will help us interpret how language is processed in the brain.

SHAP gives a global explanation by averaging over all possible feature subsets. LIME focuses on the local area near one input and explains the prediction there. These differences can lead to different explanations, which we explore in later sections.

3.2 Story 1: exorcism

For story 1, we selected *Exorcism*. We chose the top 5 voxels and stratified the analysis using two approaches: (1) we applied SHAP and LIME to assess feature importance and interpret our model at the single-voxel level; and (2) we aggregated the top 5 voxels to identify important BERT embedding features that consistently contributed to voxel response. This aggregated approach should help with revealing global influential features and associated words that may evoke higher voxel response.

To identify the top 5 voxels in Exorcism, we used Pearson correlation coefficients (CC), model weights, BERT embedding features, and voxel responses to map back the most predictable voxels from our ridge regression model.

Overall, this has been a challenging research problem with very noisy results. This is evident in the extremely low CC values calculated at the beginning of the lab—0.0440 and 0.0635 for the top 5 % and top 1 % of CCs, respectively. Meanwhile, the average and median CCs hovered around 0.01. If there's any indication that our model is capturing real signal amidst all this noise, it would be in those upper percentiles.

This rationale is exactly why we decided to focus our interpretation on the top 5 voxels—those were the points where our model picked up the strongest signal. Lastly, working with voxels that contained the strongest signal is indicative of our response variable and could further inform on the predictability of our model. This approach falls under the PCS framework where interpretability is limited to the components of the model that perform well. Based on this argument and results from part 1, the top voxel ids we chose to analyze from *Exorcism* are **73611, 34693, 79693, 73701, 28701**.

3.2.1 Story 1: SHAP and LIME on Top Voxel

The SHAP summary Figure 3 represents the top contributing BERT embedding dimensions for voxel **28701**, one of the top 5 most predictive voxels. Each row shows a BERT feature (dimension), with dots representing individual timepoints (TRs). The position along the x-axis reflects the **magnitude and direction of the SHAP value**, i.e., how much that feature contributed to increasing or decreasing the voxel's predicted activation.

Color indicates the **feature value** at each timepoint, where blue represents lower values and red higher values. For example, features like `dim_1760`, `dim_1604`, and `dim_1890` had consistently strong negative SHAP values, suggesting that **higher values of these features tended to suppress voxel activity**, while other dimensions (e.g., `dim_1536`, `dim_1951`) occasionally pushed the prediction in the positive direction.This plot tells us not just which features matter, but also how their presence or absence interacts with voxel response.

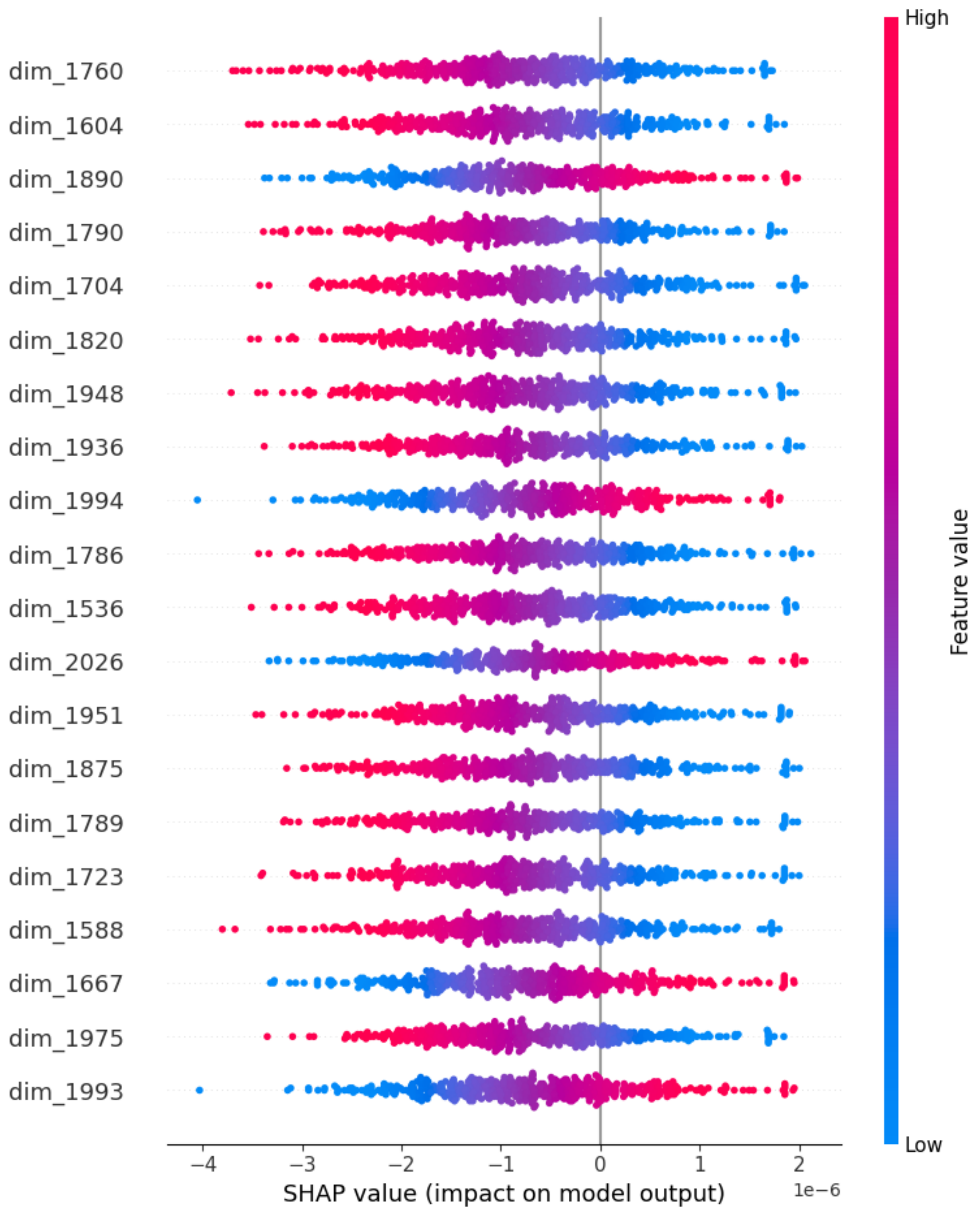
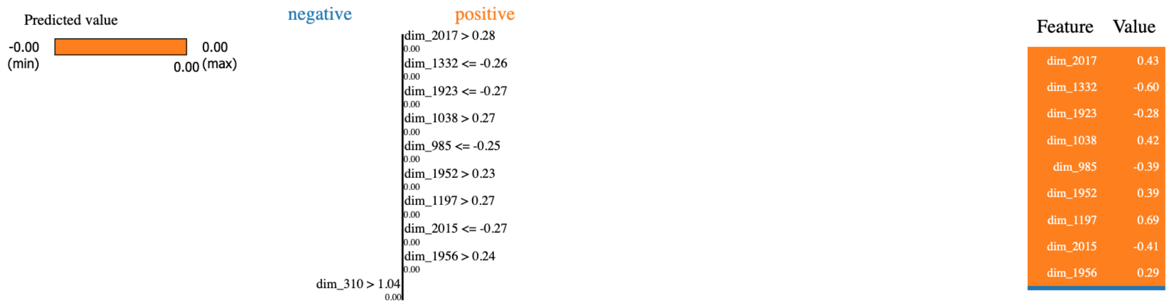


Figure 3: SHAP summary plot for one top voxel in the story *Exorcism*

The LIME explanation in Figure 4 identifies the top contributing BERT embedding dimensions for a single timepoint in our top voxel. Features like `dim_2017`, `dim_1038`, and `dim_1952` had a strong positive influence on the model's prediction, while features such as `dim_1332` and `dim_1923` negatively impacted it. Unlike SHAP, which aggregates over timepoints and reflects consistent feature impact, LIME provides a more granular snapshot, emphasizing the immediate influence of feature values on individual predictions.



Feature	Value
dim_2017	0.43
dim_1332	-0.60
dim_1923	-0.28
dim_1038	0.42
dim_985	-0.39
dim_1952	0.39
dim_1197	0.69
dim_2015	-0.41
dim_1956	0.29

Figure 4: LIME feature importance plot for the same voxel in *Exorcism*

Figure 5 summarizes the same voxel's overall SHAP activity by summing the absolute SHAP values across all features for each TR. Peaks in this line plot indicate moments in the story when the model found certain features to be particularly important in predicting the voxel's BOLD response. These peaks can be used to identify the most influential time windows, which can be mapped back to the corresponding text tokens for interpretability.

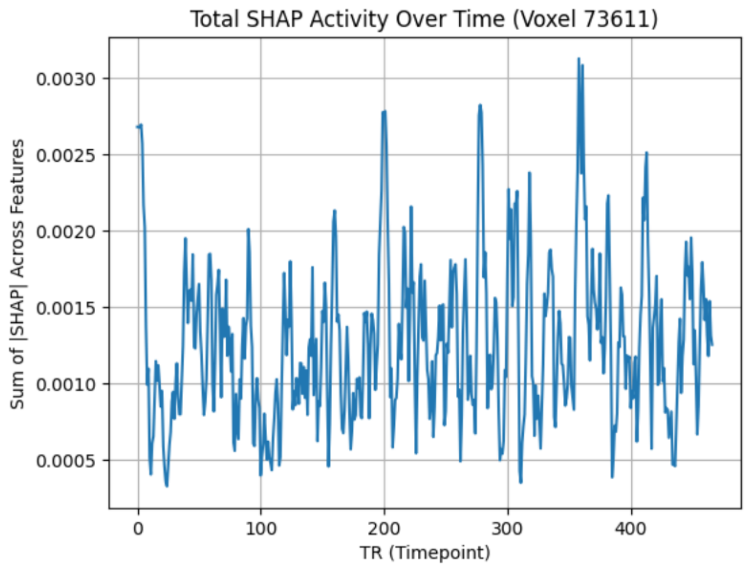


Figure 5: Total SHAP activity across all features over time

Figure 6 visualizes the SHAP values over time for Voxel 73611 using a heatmap. Each row corresponds to a TR (timepoint), and each column represents a BERT embedding feature. The color indicates the magnitude and direction of each feature's SHAP value, with red showing positive contributions and blue showing negative ones. This visualization highlights how different features dynamically influence voxel activation over time. For example, the banding patterns in the heatmap suggest temporal clusters of high-impact feature activations.

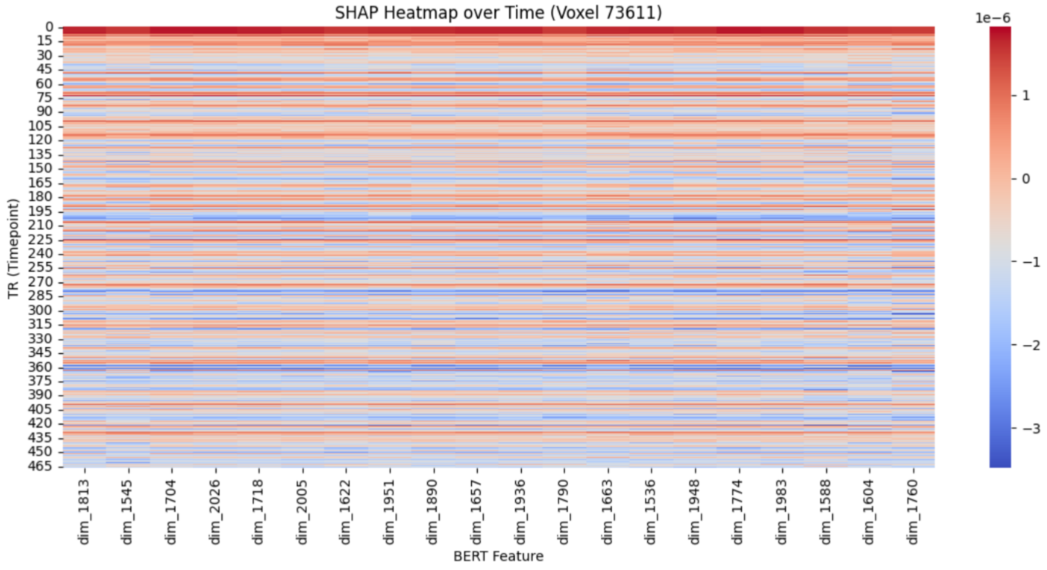


Figure 6: SHAP heatmap: Feature importance over time across BERT dimensions

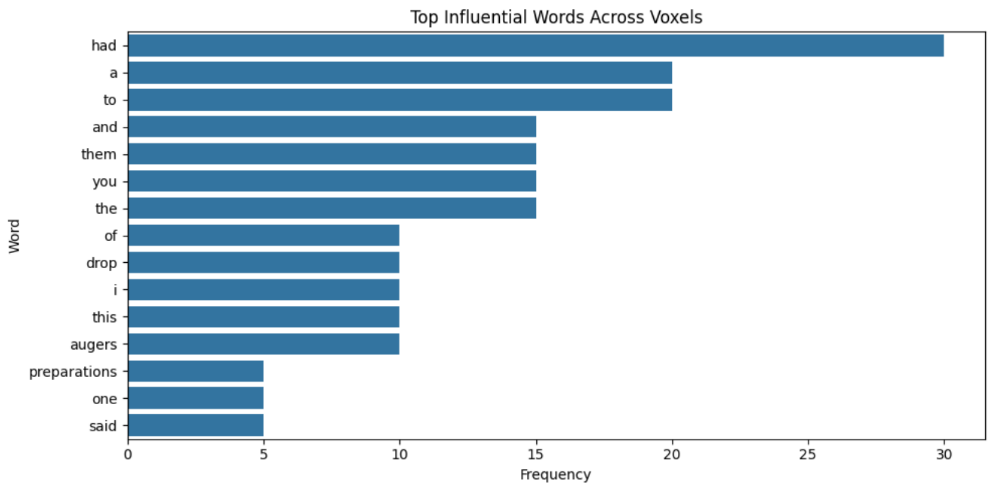


Figure 7: Top influential words identified using SHAP and LIME

The bar chart above displays the most frequently occurring words associated with high SHAP values across the top 5 voxels in Exorcism. Notably, common function words like “had”, “a”, “to”, and “and” appear most often. This suggests that the model’s predictions for voxel responses are influenced by syntactically essential but semantically broad words. Though these tokens may not carry strong content meaning on their own, their frequency was recognized and picked up on by the model. Nonetheless, some more content-relevant words like “augers”, “preparations”, and “drop” also made the list. The mix of these token types implies that SHAP is highlighting more context-based embeddings - even if the meaning of those tokens may seem somehewat abstract.

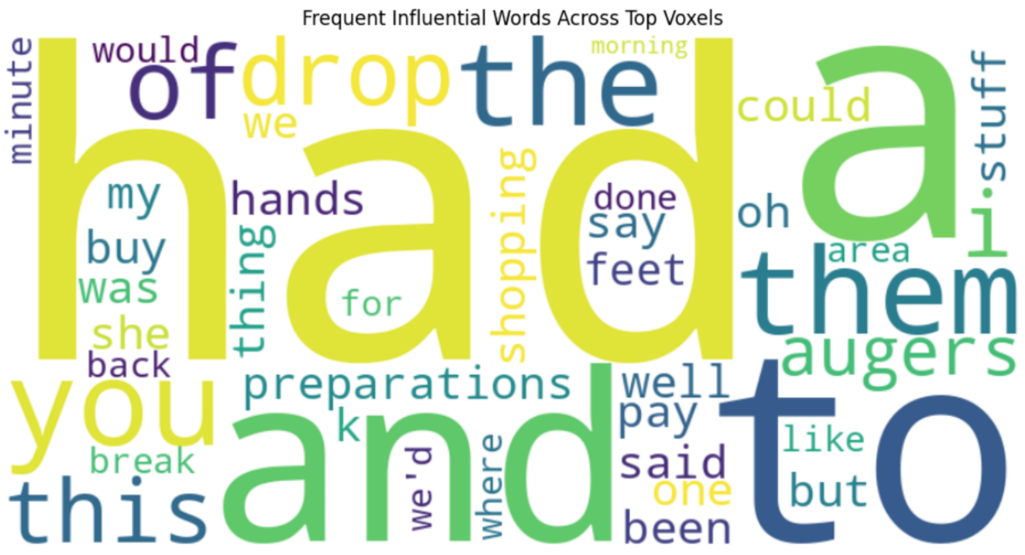


Figure 8: Token frequency comparison across SHAP and LIME

The word cloud above visualizes the most frequently identified influential words across the top 5 voxels in Exorcism. These words may seem recognlizable to figure 7, representing like "had", "a", "to", and "and." This word cloud from SHAP dominates and reflects their consistent association with high SHAPley values. Again, just like in figure 7, some content-specific terms like "shopping", "drop", "preparations", and "augers" also appear.

3.3 Story 2: tetris

3.3.1 Top Voxel Selection

To begin our interpretation analysis, we selected the test story named *tetris*, and examined which voxels had the highest prediction accuracy from our BERT model–prediction performance is measured using the Pearson correlation coefficient (CC) between the predicted and actual BOLD signal for each voxel.

We can first take a general view of the story. The first 520 characters for this story are as follows:

let's start with tetris in june of nineteen eighty four the computer programmer alexey pajitnov was working for the soviet academy of sciences in moscow he was twenty eight years old and a bit of a puzzle nerd to test the capabilities of the computers he worked on pajitnov liked to create simple games and that month he developed a game in which the player had to fit various kinds of tetraminoes together a tetramino being a shape composed of four connected squares tetraminoes come in seven possible configurations

Based on the result of part 1, we ranked all voxels based on test CC values and selected the top 5 voxels with the highest CC. These voxels are: **16841, 19609, 6465, 13744, 16789**.

In the following steps, all of our work will focus only on the top 5 voxels. This decision align with the PCS framework: interpretability should be limited to regions where the model performs well. If the model cannot accurately predict a voxel's activity, any explanation would be scientifically unreliable. We restrict SHAP and LIME analysis to these top-performing voxels, which ensures meaningful and trustworthy interpretation results.

3.3.2 SHAP and LIME on Top Voxel

For the five voxels with the highest correlation coefficients, we applied SHAP and LIME to identify the most influential embedding.

SHAP:

- Used the `KernelExplainer` from the SHAP library to interpret.

- Selected the first 100 time points as background samples.
- Explained the predictions for the first 20 time points.
- Computed SHAP values for all embedding dimensions, then aggregated to identify the top 10 most important features
- Mapped back to TR indices and aligned with original words.

LIME:

- Used the `LimeTabularExplainer` to generate local explanations.
- Selected a single time point (TR 0) as the target for explanation.
- Applied perturbation-based sampling to create a local neighborhood around the input.
- Fitted a linear surrogate model to approximate the prediction.
- Identified the top 10 contributing features from the local model.
- Mapped back to TR indices and aligned with original words.

After the two methods, we generate a SHAP summary plot and a LIME plot for each voxel, and they show similiar pattern. Figure 9 and Figure 10 show the SHAP and LIME explanations for voxel 16841 in the story *tetris*, which was identified as the most predictable voxel.

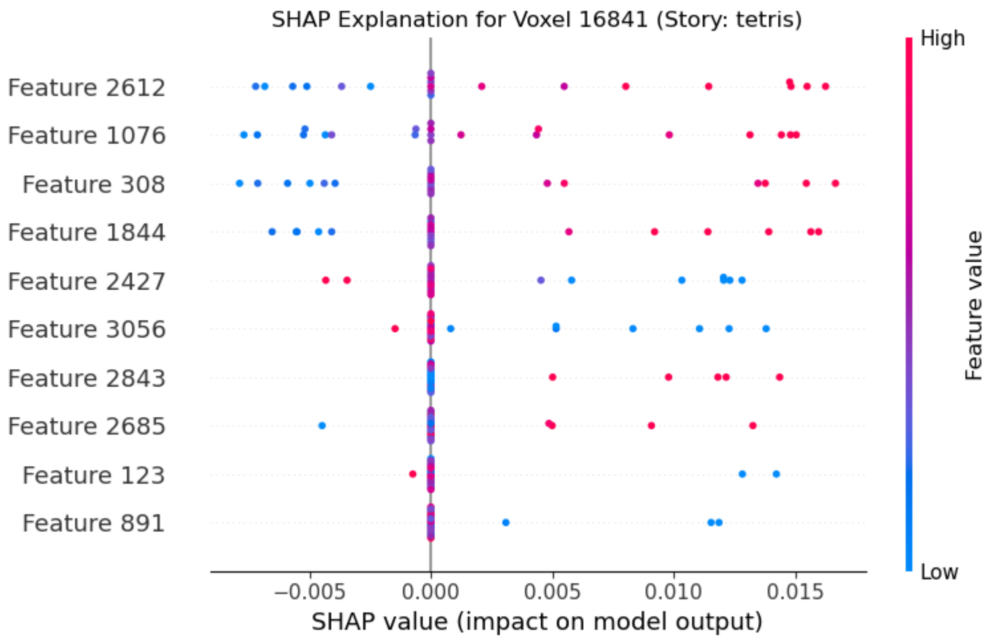


Figure 9: SHAP plot, voxel 16841, story tetris

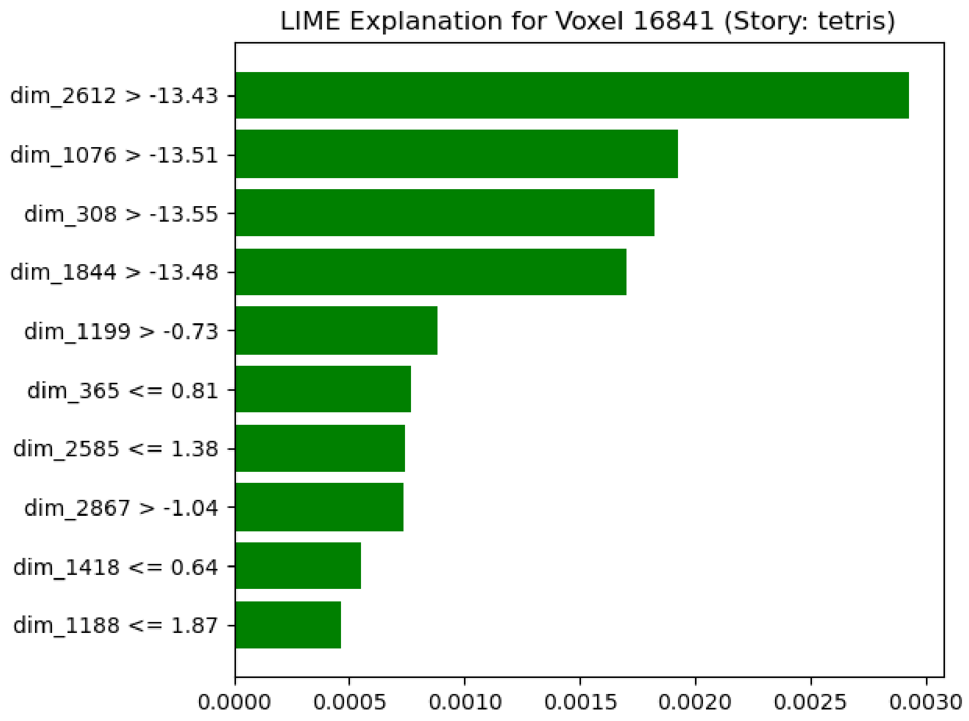


Figure 10: LIME plot, voxel 16841, story tetris

In the SHAP plot, each dot represents a time point (TR), and its position on the x-axis indicates how much a specific embedding feature contributed to the model's prediction. Red dots indicate higher feature values, and blue dots indicate lower values. Features like 2612, 1076, and 308 show strong positive contributions across multiple time points.

In the LIME plot, the bars represent the importance of the top 10 embedding features at a single time point (TR 0). Each bar reflects how much a specific feature influenced the prediction in that local neighborhood. Features such as dim_2612 and dim_1076 also appear as the most influential in LIME, showing some agreement with SHAP.

Overall, while SHAP and LIME use different technique, both methods highlight similar high-impact features for voxel 16841.

3.3.3 Token Intepretation and Comparison

After identifying the top 10 embedding dimensions, we have to map the embedding back to words in the raw text file and gain the final results for the influential words. For **voxel 16841**, the words list is as below. We retain all mapped words, including repeated ones, to reflect word frequency in the explanations.

- **SHAP tokens**: would, go, on, to, infect, time, and, space, come, in, seven, possible, and, tomb, raider, received, no, royalties, until, he, fell, in, the, darkness, genuinely, cannot, afford, to, the, soviet, union, tetris, remained, enduringly, and, dreams, of, millions
- **LIME tokens**: brief, miraculous, flicker, of, during, rare, jaunts, from, the, house, he, worked, on, pajitnov, liked, the, soviet, government, then, formed, an, tetramino, being, a, shape, composed, because, he, was, afraid, not, to, the, same, way, no, matter, how, expertly, piece, and, then, the, thrill, and, myself, from, feeling, four, the, computer,programmer

Likewise, we extract the most influential words from both SHAP and LIME across the 5 voxels. By aggregating the results from each voxel, we obtain two comprehensive lists of tokens, each with their corresponding frequency counts. For the aggregated unique token list, we have 64 unique token in SHAP and 104 unique token in LIME. This indicates that compared with SHAP, LIME tends to produce a more diverse set of tokens, possibly due to its local explanation nature.

- **SHAP–LIME Intersection**: 22 tokens
- **SHAP-only**: 42 tokens
- **LIME-only Tokens**: 82 tokens

Considering the frequncy of words in the full list, below are the top 4 most frequent tokens for each method:

- **SHAP:** to (15), and (14), in (8), he (6)
- **LIME:** the (17), and (11), of (7), from (7), then (7), to (6)

To further compare the tokens discovered by SHAP and LIME, we count the frequency of each method then combined these results. We have the top 15 overall frequency analyzes, as Figure 11 shows a side-by-side bar plot of these tokens, with separate bars for SHAP and LIME.

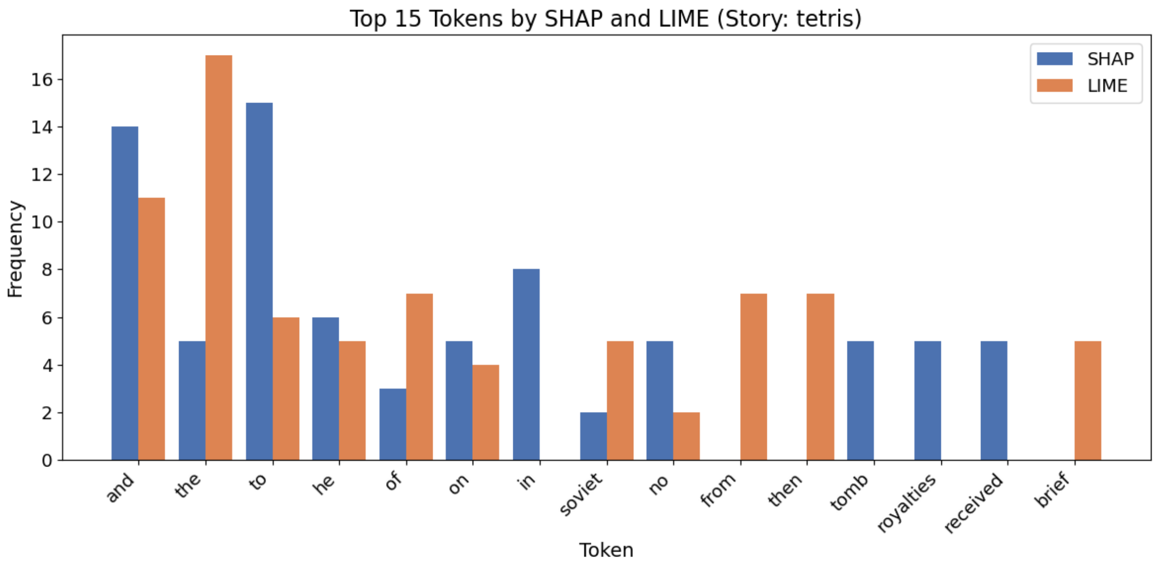


Figure 11: Top 15 Most Overall Frequent Tokens by SHAP and LIME

The frequency comparison shows that both SHAP and LIME highlight high-frequency function words such as *and*, *to*, and *the*. Notably, SHAP tends to emphasize *and*, *to*, and *in*, while LIME tends to emphasize *the*, *of*, and *from*,

Also, LIME emphasizes a broader variety of content-specific tokens such as *from*, *then*, and *brief*, which do not appear in SHAP's top list. LIME tends to produce more diverse and localized explanations, while SHAP focuses on features with more global and consistent influence across time points and voxels.

To visualize the full results, we generate separate word clouds based on token frequency for SHAP and LIME. Figure 12 and Figure 13 present the word clouds generated from SHAP and LIME explanations, aggregated over the top 5 voxels for the story *tetris*. Larger words indicate higher frequency, which suggests stronger model attribution.

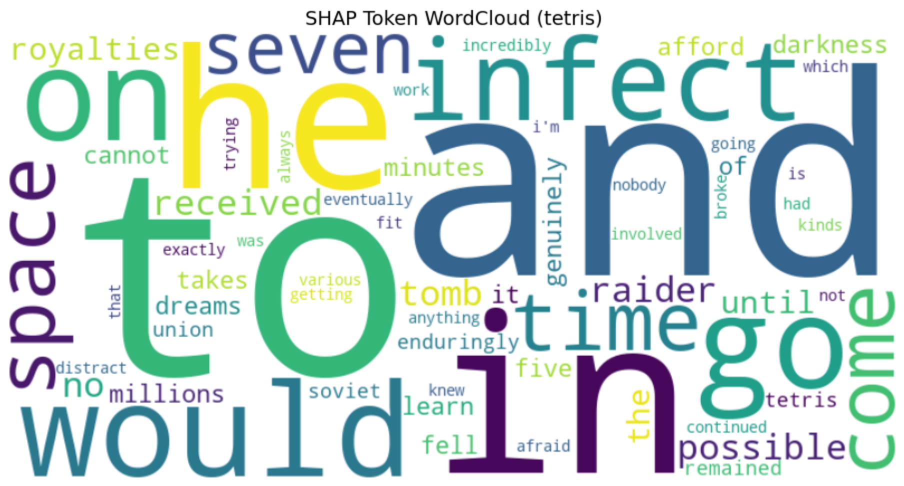


Figure 12: SHAP Wordcloud Plot, Story tetris

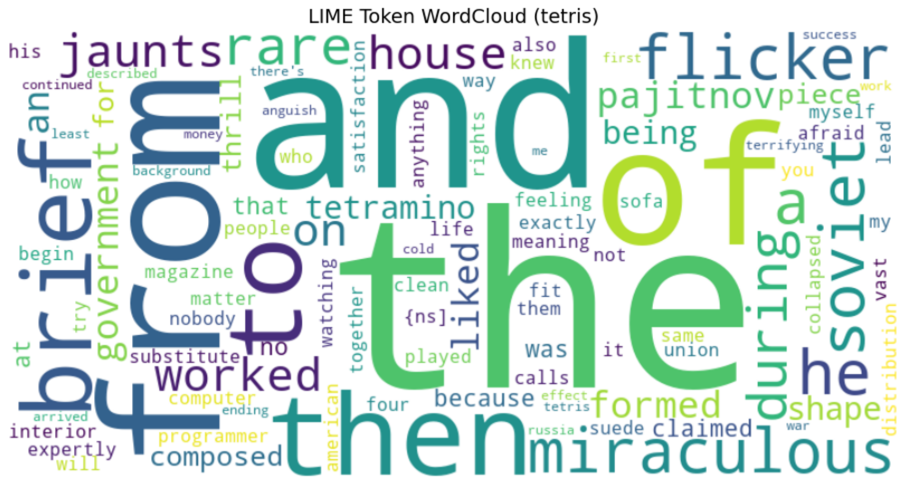


Figure 13: LIME Wordcloud Plot, Story tetris

Both methods identify common function words such as *to*, *and*, and *the*. SHAP also highlights words like *infect*, *tomb*, and *space*, which are semantically rich and possibly linked to the story content. In contrast, LIME gives more attention to context-specific terms such as *brief*, *flicker*, *government*, and *miraculous*, which might reflect localized activations at certain time points.

These observations suggest that SHAP tends to emphasize globally consistent tokens, while LIME captures more locally relevant words. The overlap between the two indicates shared model understanding, but the differences also show the complementary nature of the two methods.

Comparison across Voxels

Former comparison is mainly between the two methods SHAP and LIME. Next we will aggregate the token in the two methods, compare the token list for the 5 voxels and study the voxel-level pattern.

Below are the top 10 most frequent tokens for each voxel, based on combined SHAP and LIME outputs.

- **Voxel 16841:** the (7), and (5), to (3), he (3), on (2), in (2), no (2), soviet (2), of (2), from (2)
- **Voxel 19609:** to (4), and (4), the (4), on (2), in (2), he (2), it (2), nobody (2), knew (2), exactly (2)

- **Voxel 6465:** and (5), to (4), a (3), anything (2), of (2), tetramino (2), being (2), shape (2), composed (2), would (1)
- **Voxel 13744:** to (6), and (5), the (4), he (3), of (3), from (3), on (2), no (2), that (2), soviet (2)
- **Voxel 16789:** the (7), and (6), to (4), in (2), on (2), he (2), of (2), soviet (2), then (2), come (1)

For the token distributions across the five voxels, function words such as *the*, *and*, and *to* appear frequently in all voxels. Some voxels show unique token preferences. For example, voxel 6465 highlights tokens like *tetramino*, *being*, and *shape*, and voxel 19609 includes tokens like *nobody*, *knew*, and *exactly*.

The word clouds for voxel 6465 and voxel 19609 further reveal distinct token patterns. Voxel 6465 shows higher frequencies of game-related words, which relate directly to the story's theme about Tetris. This suggests that this voxel may be sensitive to visual or object-based semantic content. In contrast, voxel 19609 highlights words are more abstract and reflective. These tokens may reflect mental state, reasoning, or emotional interpretation. This comparison supports the idea that different brain regions encode different types of linguistic or semantic information.

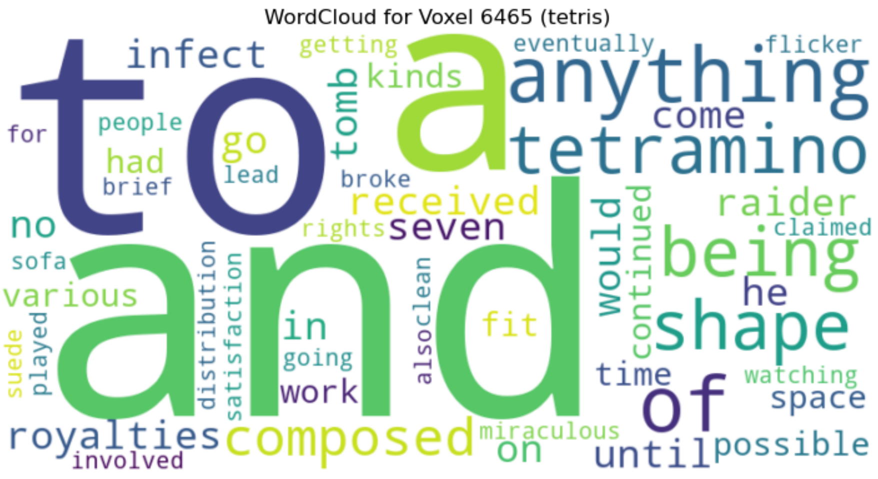


Figure 14: Voxel 6465 Wordcloud Plot, Story tetris

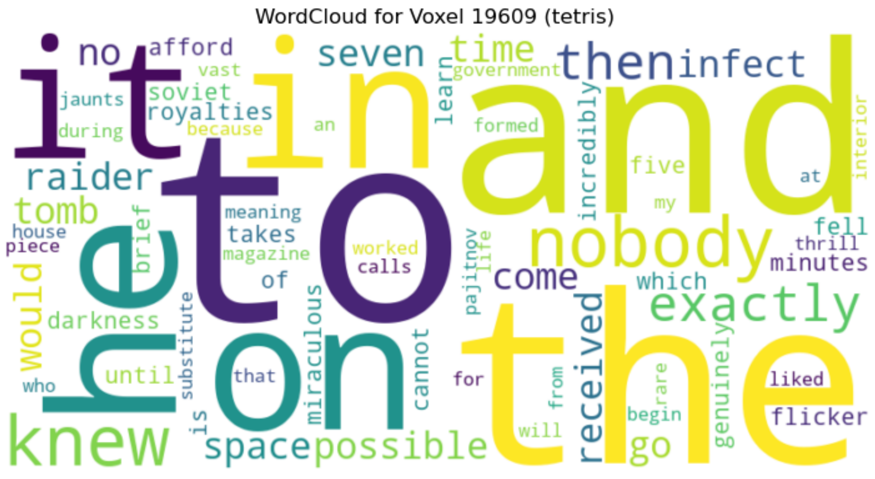


Figure 15: Voxel 19609 Wordcloud Plot, Story tetris

4 Discussion and Conclusion

Our results from Lab 3.3 reveal both the potential and limitations of fine-tuning transformer-based language models for predicting fMRI voxel responses. Despite our efforts to implement a full fine-tuning pipeline with architectural modifications, selective layer freezing, and rigorous regularization, the overall performance of the fine-tuned model underperformed relative to frozen BERT embeddings projected via ridge regression. While this was unexpected and unfortunate, it did not prevent us from conducting meaningful analysis on previous results and successfully applying SHAPley techniques to interpret the validity of our models. Applying SHAPLEy through the use of the SHAP and LIME modules will surely come in handy when working in industry. In a world that is becoming increasingly opening up and relying on AI to solve society's most pervasive problems, it is important to be able to explain and justify parts of our models and why they work they way they do.

Turning our attention back to the analysis, even though global performance declined, we still observed meaningful signal in the top 1% of voxels. This motivated our interpretability analysis in Part 2, where we focused on SHAP and LIME to uncover which BERT embedding features contributed most to voxel predictions. By restricting this analysis to well-performing voxels, we adhered to the PCS framework and avoided over-interpreting noisy or unreliable outputs.

Our findings suggest that SHAP and LIME provided complementary insights. SHAP tended to highlight features and tokens with global consistency, whereas LIME revealed more localized, time-point specific effects. Both methods frequently pointed to high-frequency function words (e.g., "to", "and", "the"), which may indicate that our model captured more structural aspects of language.

Overall, this lab reinforces the complexity of interpreting brain responses to language, especially in high-dimensional, noisy data settings. Additional exploration of model interpretability will also help bridge the gap between statistical signal and cognitive understanding. Despite challenges, our analysis highlights the potential of combining BERT embeddings with fMRI data to map the brains processing of the human language.

5 Bibliography

A Academic honesty

A.1 Statement

We make the academic integrity pledge here: All our work in this report are done independently. All sources we used are properly cited, whether from LLMs, papers, textbooks, or classmates.

Academic research honesty is necessary in the academy and also the foundation of our society. It guarantees fairness in research, stimulates research's activeness, and exerts a positive impact on the progress of science and technology. We should always adhere to the rules, which will create a more regulated and benign world.

A.2 LLM Usage

Coding

We use GitHub Copilot for several parts of debugging. We additionally used it to help with making graphs and signatures for the functions in the notebooks.

Writing

Grammarly was used to check for any spelling issues.