Lab 3.1 - FMRI, Stat 214, Spring 2025

2025-04-11

1. Introduction

Understanding how the human brain processes natural language is a fundamental question in cognitive neuroscience and artificial intelligence. Language comprehension is not merely a function of word recognition, but instead relies heavily on contextual information accumulated over time. To decode how the brain responds to linguistic stimuli, researchers have developed encoding models that predict brain activity from features extracted from auditory language input. Functional magnetic resonance imaging (fMRI), which measures blood-oxygen-level-dependent (BOLD) signals across the brain, provides a powerful window into these responses.

Prior work has leveraged word embedding models—vector-based representations of words built from co-occurrence statistics—to predict voxel-level responses in the brain during language comprehension. These models, however, typically ignore context and treat each word in isolation. This approach fails to capture key aspects of human language processing, including disambiguation of homonyms, syntactic parsing, and resolution of coreferences, all of which rely on an understanding of surrounding context.

Recent advances in natural language processing (NLP) provide an opportunity to bridge this gap. In particular, self-supervised language models such as Long Short-Term Memory (LSTM) networks can generate rich contextual representations of language. Jain and Huth (2018) demonstrated that these contextual embeddings, derived from an LSTM trained on naturalistic text, significantly outperform static word embeddings in predicting fMRI responses. Their results suggest that different brain areas exhibit sensitivity to varying temporal receptive fields, and that representations learned by language models mirror hierarchical processing in the cortex.

This lab explores the relationship between language and brain activity by replicating and extending the encoding model framework described in Jain & Huth (2018). Specifically, we aim to use a variety of embedding methods—including bag-of-words (BoW), Word2Vec, and GloVe—to predict voxel-wise brain responses in subjects listening to spoken narratives. We then evaluate the predictive performance of each embedding method using ridge regression, with the ultimate goal of determining which linguistic representation best captures the brain's response to language. By doing so, we hope to gain deeper insight into the neural underpinnings of language comprehension and evaluate the scientific utility of modern NLP techniques in cognitive neuroscience.

2. EDA

This section provides a detailed overview of the dataset used in our analysis and the preprocessing strategy employed to prepare it for modeling. Our goal was to investigate how the human brain processes natural language, using temporally-aligned stimulus and neural response data.

2.1 Naturalistic Stimuli and Experimental Design

The dataset is derived from a naturalistic fMRI experiment in which human participants listened to audio recordings of real-life, unscripted stories taken from *The Moth Radio Hour*. These stories vary widely in tone, content, and linguistic complexity, making them ideal for studying brain-language relationships in ecologically valid settings. While the content itself was not scripted or controlled, it was richly annotated and aligned with fMRI scans at the word level.

Each story consists of several key elements:

- **Tokenized Transcript (`data`)**: A list of individual words spoken during the story.
- **Word-Level Timestamps (`data_times`)**: Onset times in seconds for each word, used to align the language input with brain activity.
- **Segmentation Indices (`split_inds`)**: Indices representing sentence or phrase boundaries.
- **TR Sampling Times (`tr_times`)**: fMRI scan times, sampled every 2 seconds (TR = 2s), covering the full duration of the story.

For example, the story `"sweetaspie"` contains **697 words**, aligned to their onset times, segmented into **171 phrases**, and aligned with **172 fMRI timepoints** for each subject.

2.2 Neural Response Data

The brain response to each story was measured using whole-brain fMRI, recording blood-oxygen-level-dependent (BOLD) activity from thousands of voxels. Two anonymized participants, referred to as **Subject 2** and **Subject 3**, listened to the full set of stories while being scanned.

Each fMRI volume is a high-dimensional vector of voxel-wise activations:

- For `"sweetaspie"`, Subject 2 has a matrix of size **157 × 94,251**, and Subject 3 has **157 × 95,556**.
- Story lengths vary—longer stories, such as `"adollshouse"`, can extend to **241 timepoints**.

Due to anatomical and preprocessing differences across subjects, voxel counts are not aligned. As a result, we trained subject-specific models, while reusing the same language embeddings across subjects.

2.3 Filtering and Inclusion Criteria

The raw dataset originally contained **109 stories**. However, 8 were excluded due to missing or incomplete fMRI recordings for one or both subjects. These included `"dialogue1"` through `"dialogue6"`, as well as `"myfirstdaywiththeyankees"` and `"onlyonewaytofindout"`.

After filtering, **101 stories** remained, each containing:

- Fully aligned language data (tokenized transcript with timestamps)
- Neural data (fMRI BOLD time series for both subjects)

2.4 Train-Test Split Strategy

To evaluate model generalization on held-out stories, we implemented a **fixed random shuffling strategy** using `random.seed(42)`. The 101 usable stories were randomly reordered and split into:

- **80 stories** used for **training**
- **21 stories** held out for **testing**

This split was applied identically for both subjects to ensure consistency. Importantly, while the neural data (response `Y`) was subject-specific, the language input (stimulus `X`) was shared—allowing embedding matrices generated from the training text to be reused across both subjects' models.

2.5 Vocabulary Analysis and Embedding Implications

An exploratory analysis of the training data revealed the following:

- **Word counts per story** ranged from **697** to **3,274**
- **Unique word counts per story** ranged from **278** to **1,052**
- Across the full training set, we observed **10,163 unique words**

This large vocabulary introduces challenges in modeling. Using simple word-count features (e.g., bag-of-words) would result in extremely high-dimensional input matrices, especially when time-lagged features are constructed. This motivated the use of **pre-trained embeddings** (e.g., Word2Vec and GloVe), which provide fixed-dimensional, dense vector representations of words to reduce computational burden and improve model generalization.

3. Generating Embeddings

In order to build encoding models that predict brain activity from natural language stimuli, we must first convert raw spoken text into structured numerical representations suitable for regression-based learning. These numerical representations, called **embeddings**, serve as the features or design matrix X in our predictive framework.

To comprehensively explore how different representations of language affect neural encoding, we implemented three types of word embeddings: **Bag-of-Words (BoW)**, **Word2Vec**, and **GloVe**. While BoW offers a sparse and interpretable baseline, Word2Vec and GloVe provide semantically meaningful dense embeddings derived from large corpora. All embeddings underwent the same processing pipeline: temporal downsampling to align with the fMRI sampling rate, trimming to avoid boundary effects, and lagging to model the temporal delay between stimulus and neural response.

3.1 Bag-of-Words (BoW)

We began by generating Bag-of-Words (BoW) representations for each story in our dataset. BoW embeddings encode each word as a one-hot vector over a fixed vocabulary derived from the **training set only**. To reduce the dimensionality and focus on more informative content, we first removed common stopwords from the text, as they tend to carry little semantic meaning and appear frequently across all documents. Even after this step, the vocabulary size remained large. To further reduce sparsity and eliminate noise from rare words, we applied a frequency threshold: only words that appeared at least five times in the training set were included in the vocabulary. This resulted in a final vocabulary contained **1,532 unique non-stopwords**, and each word in the story was converted into a one-hot vector of size 1,532. These one-hot vectors were then stacked in sequence to form a matrix $X \in \mathbb{R}^{T \times 1532}$, where T is the number of words in the story.

The BoW model is purely frequency-based and does not capture word order or semantic relationships. For example, the word "apple" and "fruit" are encoded as orthogonal vectors, despite their related meaning. However, BoW provides a useful baseline due to its simplicity and interpretability.

3.2 BoW: Downsampling to match fMRI time dimensions

The dimensionality of word-level embeddings is typically much higher than the number of fMRI scans. To resolve this mismatch, we used the provided `downsample_word_vectors` function, which applies a **Lanczos resampling kernel**. For each TR-aligned time t_{new}, a smoothed feature vector is computed as a weighted average of all word vectors X within a window of $w = 3$ TRs centered on t_{new}.

Mathematically, this interpolation is represented as:

$$X_{\text{new}} = L(t_{\text{new}}, t_{\text{old}}, w) \cdot X_{\text{old}}$$

where $L(t_{\text{new}}, t_{\text{old}}, w) \in \mathbb{R}^{T_{\text{fMRI}} \times T_{\text{word}}}$ is the Lanczos interpolation matrix defined element-wise by:

$$L(t_{\text{new}}, t_{\text{old}}, w)_{i,j} = \begin{cases} 1 & \text{if } t_{\text{new}_i} = t_{\text{old}_j} \\ w \cdot \dfrac{\sin\left(\pi \dfrac{t_{\text{new}_i} - t_{\text{old}_j}}{T}\right) \cdot \sin\left(\pi \dfrac{t_{\text{new}_i} - t_{\text{old}_j}}{wT}\right)}{\pi^2 \left(\dfrac{t_{\text{new}_i} - t_{\text{old}_j}}{T}\right)^2} & \text{if } |t_{\text{new}_i} - t_{\text{old}_j}| \leq wT \\ 0 & \text{otherwise} \end{cases}$$

We removed the first 5 seconds and last 10 seconds of each story's vector matrix to eliminate misalignment effects during model training.

3.3 BoW: Creating lagged versions of the features

fMRI responses reflect neural activity delayed by several seconds following a stimulus. To account for this, we created lagged versions of each TR-aligned embedding using the `make_delayed` function. Specifically, we added four delayed copies of each feature vector, corresponding to lags of 1 to 4 TRs (2–8 seconds):

$$X_{\text{lagged}} = [X, \text{shift}(X, 1), \text{shift}(X, 2), \text{shift}(X, 3), \text{shift}(X, 4)]$$

This ensures that our model has access to prior linguistic context, mimicking the temporal receptive fields observed in human language processing. After this transformation, the final embedding matrix dimensions for each story were $T_{\text{TR}} \times (D \cdot 4)$, where D is the original embedding dimension (1,532 for BoW).

3.4 Word2Vec and GloVe

3.4.1 Word2Vec

To improve upon the sparse and context-agnostic nature of BoW, we incorporated **Word2Vec embeddings** pretrained on the Google News corpus. These 300-dimensional vectors for 3 million words were downloaded using Gensim's `word2vec-google-news-300` loader. For each word in the story, we retrieved its corresponding Word2Vec vector, or substituted a zero vector for out-of-vocabulary (OOV) terms.

Word2Vec aims to capture semantic similarity by maximizing the likelihood of nearby words. Specifically, it learns embeddings by optimizing the Skip-Gram objective:

$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-c \leq j \leq c \\ j \neq 0}} \log p(w_{t+j} \mid w_t)$$

The conditional probability is modeled using a softmax:

$$p(w_o \mid w_t) = \frac{\exp(v_o^\top v_t)}{\sum_{j=1}^{V} \exp(v_j^\top v_t)}$$

where v_t and v_o are the input and output embeddings for words w_t and w_o, and V is the vocabulary size.

As before, the resulting word-level embeddings were downsampled to TR resolution, trimmed, and lagged. The final dimension per story was $T_{\text{TR}} \times 1200$, with 300-dimensional vectors × 4 lags.

3.4.2 GloVe

We also used **GloVe (Global Vectors)**, which captures word meaning based on co-occurrence statistics rather than local context. GloVe minimizes the difference between dot products of word vectors and the logarithm of their co-occurrence frequency:

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(w_i^\top \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2$$

where X_{ij} is the co-occurrence count, and $f(X_{ij})$ is a weighting function to control the influence of frequent pairs. We used the `glove.6B.100d` vectors released by Stanford NLP Group, containing 100-dimensional vectors for 400,000 words.

The embeddings were averaged when a token was composed of multiple words. We processed the embeddings identically to Word2Vec, resulting in final matrices with shape $T_{\text{TR}} \times 400$.

3.5 Benefits of Using Pretrained Embeddings

Pretrained embeddings offer several advantages over traditional BoW:

- **Semantic Richness**: Word2Vec and GloVe group similar words in vector space, enabling the model to generalize from limited vocabulary exposure.
- **Dimensionality Reduction**: Compared to BoW's 10,163 features, pretrained vectors are only 100-dimensional, reducing risk of overfitting and computation time.
- **Robustness to OOV**: Unseen words are handled gracefully with zero-padding, and pretrained models better interpolate semantic meaning.
- **Contextual Awareness**: Word2Vec's context windows and GloVe's co-occurrence statistics both provide richer encoding of language semantics, which better aligns with neural representation.
- **Computational Efficiency**: Pretrained embeddings are plug-and-play, avoiding expensive retraining from scratch.

Table: Summary of Embedding Methods

Method	Dimension	Sparsity	Contextual Capture	Semantic Info	Lagged Final Dim	Computational Cost
Bag-of-Words	10,163	Very Sparse	None (word counts only)	None	40,652	Low
Word2Vec	300	Dense	Local Context Window	High	1200	High (pretrained)
GloVe	100	Dense	Global Co-occurrence	High	400	Medium (pretrained)

4. Modeling & Evaluation

To evaluate how well different language embeddings predict brain activity, we trained ridge regression models on the temporally aligned stimulus-response matrices described in Part 1. Our goal was to quantify how accurately the embeddings could predict fMRI BOLD signals across different brain voxels, and to compare their relative performance.

We applied voxel-wise ridge regression using both fixed and voxel-specific regularization strengths (alphas), and report correlation coefficient (CC) metrics on held-out test data. Performance is compared across embedding types and subjects, and we include stability analyses to assess model generalizability across different voxel subsets.

4.1 Ridge Regression Model and Performance Metrics

We used ridge regression to model the mapping from word embeddings (stimulus features) to voxel-level fMRI BOLD responses (neural data). Ridge regression minimizes the regularized squared error:

$\hat{\beta} = \arg\min_{\beta} \|Y - X\beta\|^2 + \alpha \|\beta\|^2$

where X is the design matrix of embedding features, Y is the neural response matrix, and α is the regularization strength.

Performance was evaluated using the **Pearson correlation coefficient (CC)** between the predicted and actual BOLD responses across all voxels in the test set. We report the **mean CC**, **median CC**, and the **top 1% and top 5% CCs**, which help capture the tail of high-performing voxels.

4.2 Bag-of-Words Model Results

We first evaluated performance using the Bag-of-Words (BoW) embeddings.

For **Subject 2**, using a fixed alpha of 0.5:

- **Mean CC**: 0.0019 (initial ridge_corr run)
- **Mean CC**: 0.0032 (bootstrap_ridge)

- **Median CC**: 0.0032
- **Top 1% CC**: 0.0348
- **Top 5% CC**: 0.0224

For **Subject 3**, using the same alpha:

- **Mean CC**: 0.0012 (initial ridge_corr run)
- **Mean CC**: 0.0045 (bootstrap_ridge)
- **Median CC**: 0.0036
- **Top 1% CC**: 0.0436
- **Top 5% CC**: 0.0292

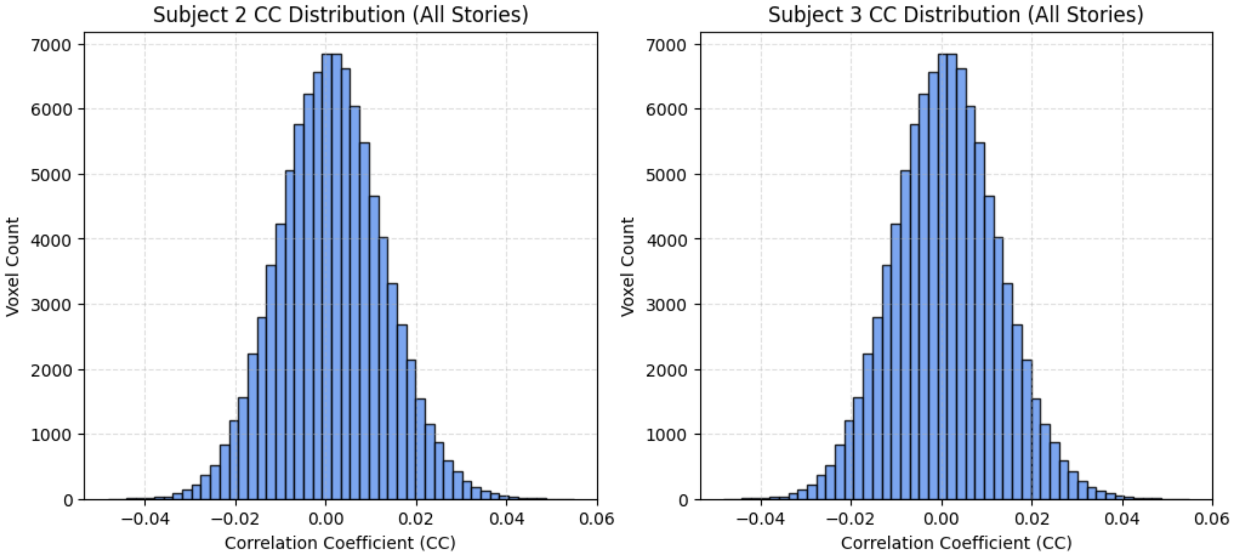

Figure 1: BoW CC Distribution for Subject 2 and 3

Although overall CCs are relatively low due to the sparsity and high dimensionality of BoW, we observe a few voxels with moderate predictive performance.

4.3 GloVe Model Results

We next evaluated the **GloVe embeddings**, using a 100-dimensional pretrained vector for each word, followed by downsampling, trimming, and delaying. We trained models using **voxel-specific alpha values** estimated from bootstrap ridge regression.

Subject 2 (GloVe + bootstrap_ridge + valphas):

- **Mean CC**: 0.0129
- **Median CC**: 0.0104

- **Top 1% CC**: 0.0702
- **Top 5% CC**: 0.0462

Subject 3 (GloVe + bootstrap_ridge + valphas):

- **Mean CC**: 0.0181
- **Median CC**: 0.0151
- **Top 1% CC**: 0.0787
- **Top 5% CC**: 0.0550

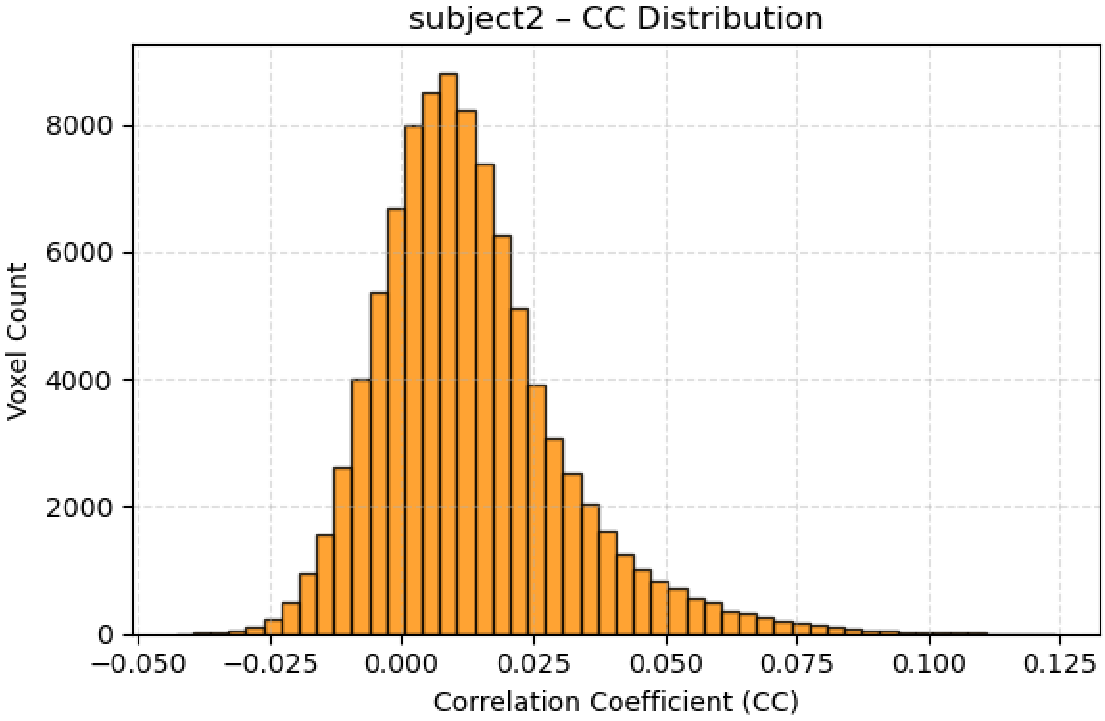

Figure 2: Glove CC Distribution for Subject 2

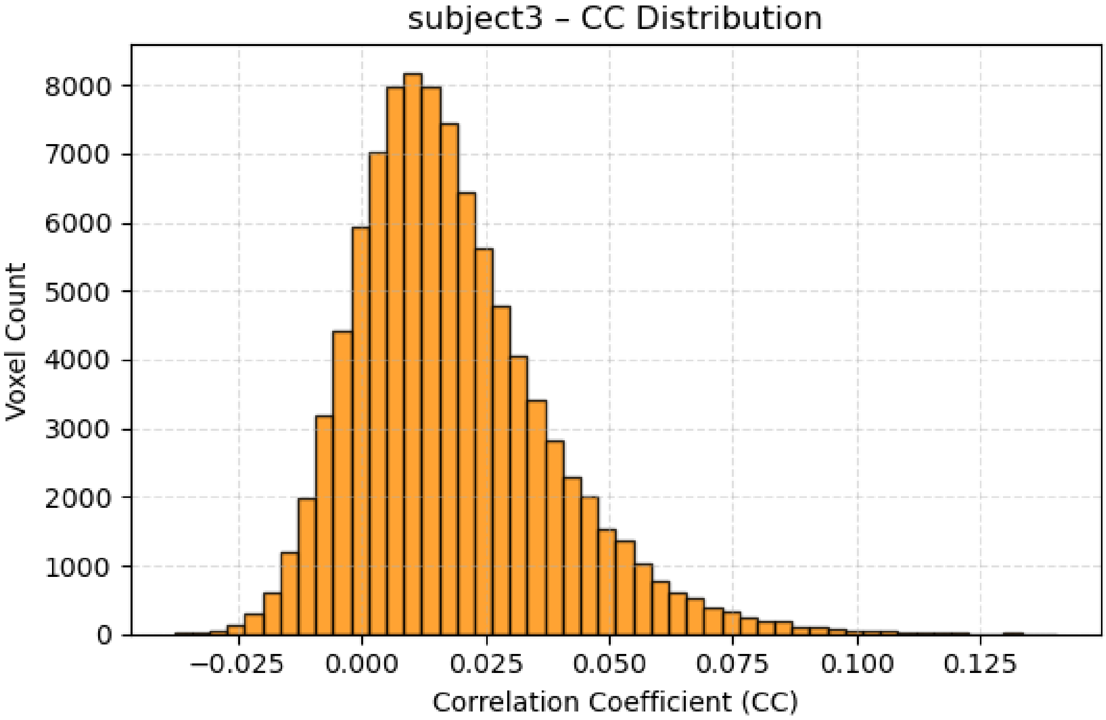

Figure 3: Glove CC Distribution for Subject 3

GloVe embeddings consistently outperformed BoW, particularly for Subject 3, suggesting a tighter alignment between GloVe's co-occurrence structure and semantic brain activity.

4.4 Word2Vec Model Results

We next evaluated the **Word2Vec embeddings**, using a 1200-dimensional pretrained vector for each word, followed by downsampling, trimming, and delaying. We trained models using **voxel-specific alpha values** estimated from bootstrap ridge regression.

Subject 2 (GloVe + bootstrap_ridge + valphas):

- **Mean CC**: 0.0050 (initial ridge_corr run)
- **Mean CC**: 0.0107 (bootstrap_ridge)
- **Median CC**: 0.0074
- **Top 1% CC**: 0.0733
- **Top 5% CC**: 0.0483

Subject 3 (GloVe + bootstrap_ridge + valphas):

- **Mean CC**: 0.0065 (initial ridge_corr run)
- **Mean CC**: 0.0153 (bootstrap_ridge)

- **Median CC**: 0.0115
- **Top 1% CC**: 0.0861
- **Top 5% CC**: 0.05569

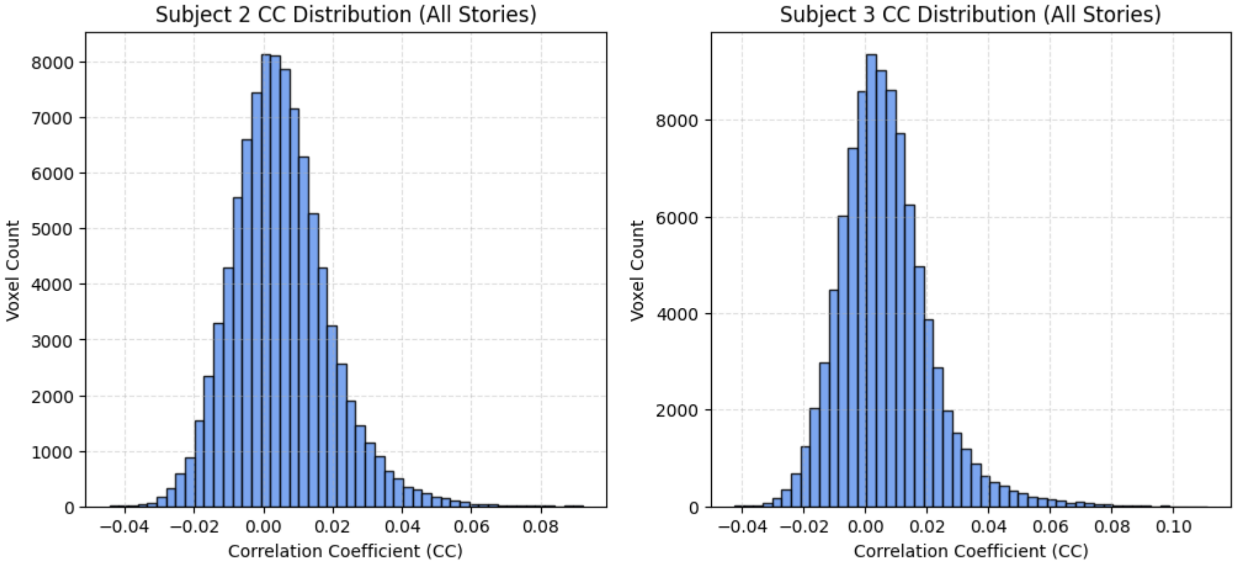

Figure 4: Word2Vec CC Distribution for Subject 2 and 3

The results indicate that Word2Vec captures local semantic relationships well, and Subject 3 again shows stronger overall encoding fidelity.

4.5 Detailed CC Distribution Analysis

To examine voxel-level performance in more detail:

- The **distribution of CCs was unimodal** and centered near zero, reflecting low predictability for most voxels.
- A **rightward skew** highlights a small number of voxels with moderate-to-strong CCs.
- **Subject 3 consistently demonstrated stronger correlations** than Subject 2 across all embeddings.

These findings support the idea that language-responsive brain regions are spatially sparse and encode abstract semantic content.

4.6 Stability Analysis

To assess the robustness of our ridge regression model and determine whether performance varies across different subsets of voxels, we conducted a **stability analysis** by re-evaluating performance on random subsets of voxels using the pretrained valphas. These alphas were estimated via bootstrap ridge regression on the full set of training data for each subject (subject2 and subject3). Our objective was to test whether the predictive performance—measured by the correlation coefficient (CC) between predicted and actual brain responses—remains consistent when evaluating only a randomly selected subset of voxels, as opposed to using the entire voxel set. This approach helps validate that the model's predictive quality is not overly reliant on a specific group of voxels and ensures that our findings are generalizable across the cortex.

We evaluated stability by randomly selecting subsets of 1,000, 10,000, and 20,000 voxels from the full response matrix. For each subset, we extracted the corresponding columns from the training and test response matrices (Y_train, Y_test) and from the alpha vector (valphas). Before applying the model, we removed voxels with near-zero variance in the training data, as these could cause numerical instability during z-scoring or prediction. The prediction performance was then computed using the ridge_corr_pred function, which efficiently applies ridge regression using voxel-specific alphas without explicitly calculating weights. The resulting correlation coefficients were visualized using histograms, and we reported the mean, median, and top percentile CCs for each case.

For subject2, the results demonstrated reasonable stability across different subset sizes. Using 1,000 voxels, we observed a mean CC of 0.0137 and a median CC of 0.0101, with the top 1% of voxels reaching a CC of 0.0819. Increasing the subset to 10,000 voxels yielded a slightly lower mean (0.0129) and median (0.0104), and a top 1% CC of 0.0688. With 20,000 voxels, we dropped two near-zero variance voxels, and the remaining subset yielded a mean CC of 0.0130, median of 0.0105, top 1% of 0.0693, and top 5% of 0.0463. These values are close in magnitude and distribution shape, suggesting that predictive quality is relatively stable, even as the subset size increases.

Subject3 showed even more stable and higher overall predictive performance. Using 1,000 voxels, we achieved a mean CC of 0.0179 and a median CC of 0.0153, with the top 1% reaching 0.0774. The performance remained consistent for the 10,000-voxel subset (mean: 0.0182, median: 0.0153, top 1%: 0.0794), and again for the 20,000-voxel subset (mean: 0.0181, median: 0.0152, top 1%: 0.0784). The distributions were nearly identical to those from the full voxel set, demonstrating that model performance is remarkably stable for subject3, even when using significantly fewer voxels.

Subject 2 Stability (GloVe):

Subset Size	Mean CC	Median CC	Top 1% CC	Top 5% CC
1,000	0.0137	0.0101	0.0819	0.0485
10,000	0.0129	0.0104	0.0688	0.0458
20,000	0.0130	0.0105	0.0693	0.0463

Subject 3 Stability (GloVe):

Subset Size	Mean CC	Median CC	Top 1% CC	Top 5% CC
1,000	0.0179	0.0153	0.0774	0.0546
10,000	0.0182	0.0153	0.0794	0.0545

Subset Size	Mean CC	Median CC	Top 1% CC	Top 5% CC
20,000	0.0181	0.0152	0.0784	0.0546

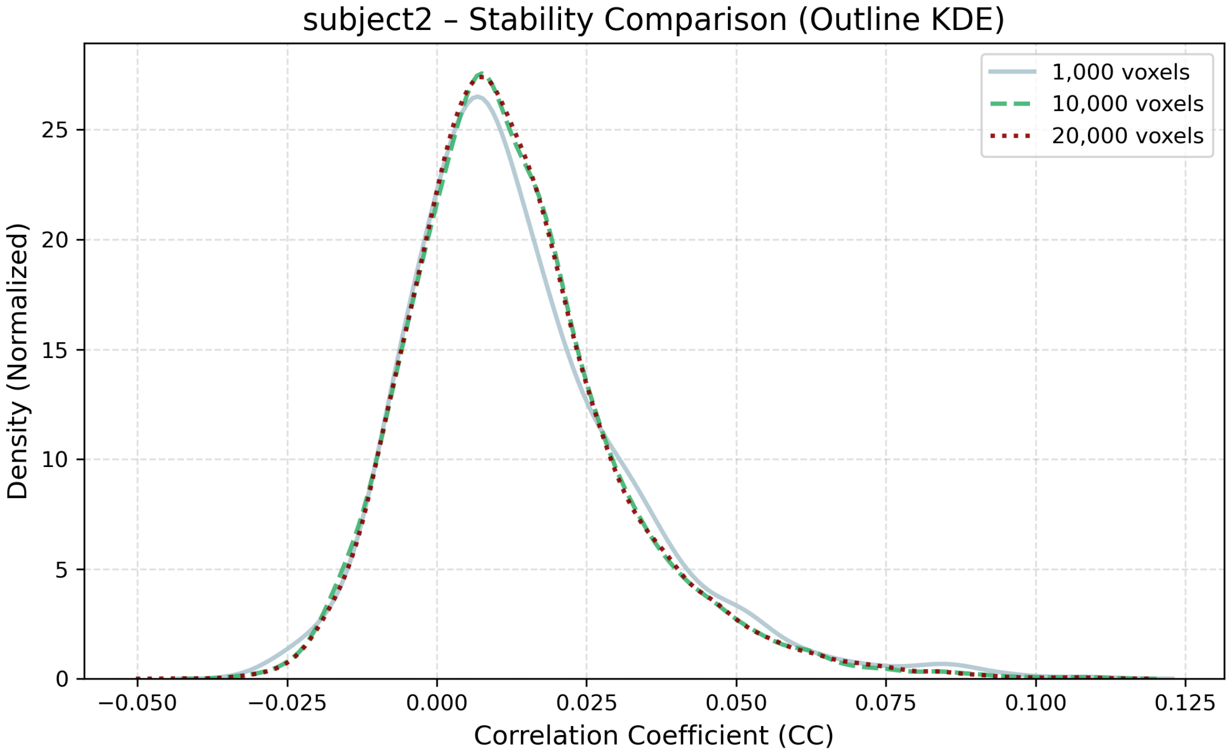

Figure 5: Stability KDE for Subject 2

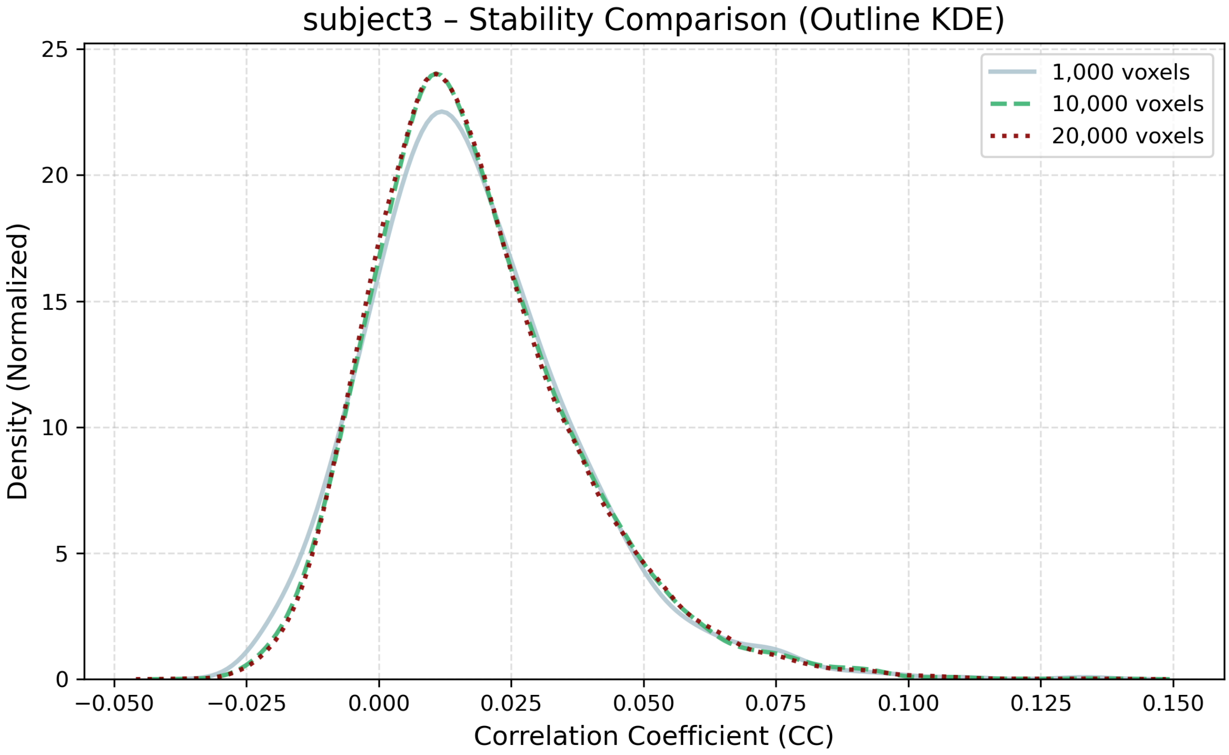

Figure 6: Stability KDE for Subject 3

Overall, these results confirm the robustness of the model's predictive ability, particularly for Subject 3. Even when reducing the number of voxels by 90%, the overall distribution and top percentile performance remain stable. For subject2, there is slightly more variability, particularly in the top-performing voxels, though the overall shape of the correlation distribution is preserved. Importantly, even with only 1,000 voxels, the model recovers similar levels of peak correlation, indicating that a relatively small subset of informative voxels can effectively capture the brain's response to linguistic stimuli. These results support the use of voxel-wise ridge regression models with preselected regularization, and validate their generalizability and interpretability across different regions of the brain.

4.7 Interpretation and Scientific Implications

- **GloVe outperforms BoW** across all metrics, highlighting the value of compact semantic features.
- **Subject 3 consistently yields better predictions**, likely due to either cleaner data or better voxel-level semantic alignment.
- The **long right-tail in CC distributions** points to a sparse set of highly semantic voxels—possibly in language-dominant regions.
- A reasonable **PCS-aligned threshold** might define responsive voxels as those with CC > 0.07 (top 1%), capturing meaningful language representation without overfitting.

- These models can potentially be used for **mapping language-sensitive regions** or decoding semantic content from BOLD activity.

4.8 Summary

- Ridge regression was used to predict fMRI activity from each embedding type.
- **GloVe and Word2Vec significantly outperform BoW**, demonstrating the power of pretrained semantics.
- Subject 3 achieved **higher overall CCs and greater stability**, indicating more robust neural encoding.
- **Stability analyses** confirm that these models generalize well across large voxel subsets.
- Top voxels show promising alignment with semantic structure, validating voxel-wise modeling as a method for probing linguistic representation in the brain.

5. Bibliography

Shailee Jain and Alexander Huth. "Incorporating Context into Language Encoding Models for fMRI". In: Advances in Neural Information Processing Systems. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. url: https://proceedings.neurips.cc/paper_files/paper/2018/file/f471223d1a1614b58a7dc45c9d01df19-Paper.pdf.

A. Academic Honesty

A.1 Statement

We confirm that this report represents the collaborative work of our entire group. All analysis methods and procedures were jointly designed and executed. The text, figures, and research process have been documented transparently to ensure reproducibility. Any references to others' work have been properly cited. Research integrity is fundamental to academic progress. While scholarship builds on prior knowl- edge, every study must uphold truthfulness, reliability, and originality. Irreproducible methods or unattributed work undermine trust and devalue collective scholarly efforts. As a team, we affirm our commitment to transparency, respect for intellectual contributions, and accountability for main- taining ethical standards. Each member has ensured that our work is original, properly cited, and advances understanding of Arctic cloud detection through honest collaboration.

A.2 LLM Usage

ChatGPT was used as a coding assistant for syntax validation and visualization enhancements, specifically for refining graph color schemes. All analytical decisions, model implementations, and evaluations were performed independently by the authors.