Low-dimensional Representations of Semantic Context in Language and the Brain

Kiran Vodrahalli

Advised by Professors Sanjeev Arora and Ken Norman

Department of Mathematics

Princeton University

May 2, 2016

A senior thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Arts in Mathematics at Princeton University

This thesis represents my own work in accordance with University regulations.

$\sim$ Kiran Vodrahalli

Abstract

We study the problem of finding low-dimensional shared representations of meaning for natural language and brain response modalities for multiple-subject narrative story datasets (a portion of an episode of the *Sherlock* television program and a chapter of a *Harry Potter* book). These datasets have paired fMRI responses and textual descriptions. Our first goal is to determine if any fMRI space can be learned across subjects that correlates well with semantic context vectors derived from recent, unsupervised methods in natural language understanding for embedding word meaning in $\mathbb{R}^n$. Can distributed, low-dimensional representations of narrative context predict voxels? Our second goal is to determine if a shared space between the fMRI voxels and the semantic word embeddings exists which can be purposed to decode brain states into coherent textual representations of thought.

First, we were able to construct a fine-grained 300-dimensional embedding of the semantic context induced by a scene annotation dataset for *Sherlock*. Our primary positive result in this thesis is that the multi-view Shared Response Model produces a semantically relevant 20-dimensional space using views of multiple subjects watching *Sherlock*. This low-dimensional shared fMRI space is able to match fMRI responses to scenes with performance considerably above chance. Using the fMRI shared space over individual fMRI responses brings a large improvement in reconstructing voxels from semantic vectors, and suggests that other recent work in this area may benefit from applying the Shared Response Model.

Acknowledgements

First and foremost, I would like to thank my two advisors Professors Sanjeev Arora and Ken Norman for the vast amount of time they put into this project, for helping direct my research throughout the course of this year, and for providing ideas and generally making themselves available as a resource. I'm excited to continue working with both of you in the upcoming summer and year. I'd also like to extend a big thank you to Professor Zeev Dvir for being my second reader.

I am indebted to many helpful discussions with Cameron Chen, who met with me every week to discuss strategies and next steps, and also for the use of his well-documented and efficient Shared Response Model code. I must also thank Yingyu Liang (whose expertise at constructing word embeddings was invaluable), Chris Baldassano, and Janice Chen. Without Janice I would not have a (good) dataset to use. Without Chris I might still be stuck in the deep depths of fMRI preprocessing hell. I look forward to continuing to work with all of you.

Finally, I would be remiss to not acknowledge the role my friends and family have played in supporting my thesis struggles throughout the year. Thank you all.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Several researchers have attempted to find relationships between word featurizations and fMRI activations in the brain. One popular method due to Mitchell et al. (2008) [34] gathers fMRI data across several subjects corresponding to text stimuli: individual nouns [34], a set of words [38], and even a story [46]. In this thesis, we quantitatively investigate the interface between mind and language, with the central goal of giving explicit maps between measurements of neural activity and the words describing the thoughts the brain experiences. Particularly, we investigate two functional Magnetic Resonance Imaging (fMRI) datasets, both of which record multiple subjects perceiving a story and which are paired with textual annotations describing the stimulus.

The Sherlock Dataset

The Sherlock dataset (due to Chen et al. (2016) [8]) consists of fMRI recordings X_i of $i = 1, \ldots, 17$ people watching the British television program *Sherlock* for 48 minutes. This data is split into two fMRI collection periods of 25 and 23 minutes respectively. fMRI images are measured at a rate of one image every 1.5 seconds (TR = 1.5). All $70,000$ voxels are available for analysis, and several region-of-interest (ROI) masks are presented with number of voxels on the order of 100 or 1000. An important attribute of this dataset is that it combines both audio and visual stimuli with a narrative, and thus the resulting

fMRI responses are similar to a more natural brain states, allowing us to study the brain's representation of meaning in a real world setting [8].

In addition to the fMRI information, we use externally annotated, sub-second-resolution, English text scene annotations of the program. These annotations give descriptions of the story setting, characters' emotions, dialogue, plot points, and dialogue occurring in the scene. An example annotation of a scene where detective characters Donovan and Lestrade report on a string of "serial suicides" to reporters from various British news outlets is as follows: "Donovan looks up at the reporters and continues: 'Preliminary investigations...' Lestrade looks distressed. Donovan continues: '... suggest that this was suicide. We can confirm that this...".

The Harry Potter Dataset

The Harry Potter dataset records the fMRI BOLD response of 9 subjects as they read chapter 9 of *Harry Potter and the Philosopher's Stone* by J.K. Rowling [40], presented as a visual reading experience akin to some speed-reading apps. All subjects were familiar with the Harry Potter series and its characters before data collection. Before being presented with the Harry Potter stimulus, subjects were given an unrelated story to practice reading to become accustomed to the mode of information presentation. The collection of this dataset is due to Wehbe et al. (2014) and is divided into 4 collection periods of 11 minutes each. fMRI images are measured at a rate of one image every 2 seconds (TR = 2), and a word is presented every 0.5 seconds, leaving us with 4 words per TR. Different numbers of voxels are presented for the 9 subjects, ranging between roughly $24000 - 34000$ voxels. AAL Atlas labels are maintained for ROI recovery, but the voxels in each ROI are not anatomically aligned [46].

In addition to the text made available by Wehbe et al. (2014), we preprocess the entire set of *Harry Potter* books $1 - 7$ for the purposes of creating a reasonably-sized corpus on the order of 10^6 words long. We use this corpus to construct semantic word embeddings with which the fMRI data are matched.

Goals

We have multiple goals and questions we pursue throughout this thesis:

- How might we best featurize the raw fMRI data to reflect semantic meaning across time, given multiple subjects and the assumption that they are experiencing the same story stimulus?
- Given a textual description of a story, what is an accurate way to represent the story context as it changes over time? How can we adapt word vectors to the problem of encoding stories? Do these word embeddings identify semantically similar time points?
- How redundant are the voxel patterns in our brain? That is, how low-dimensional is the semantic information encoded in our brains?
- Is it possible to find a shared low-dimensional space which encodes both brain and story features that also induces a map between mind and semantic context which generalizes across multiple people?
- To what extent are mental representations of story scenes common across people, and can these representations (essentially, voxel activations) be explained with semantic vectors derived from unsupervised methods based on the distributional hypothesis of meaning?
- To which regions of the brain is it possible to fit a map from semantic context embeddings which attains good voxel reconstruction capability?
- Can the mental representations of stories presented via natural visual-audio stimuli (i.e. movies) also be explained by distributional word and context embeddings?
- Can we decode fMRI images into text? In other words, can we automate the transcription of thoughts?

However, among these varied questions, we have two principal goals: Our first objective is to determine if any fMRI space can be learned across subjects that correlates well with semantic context embeddings. Can distributed, low-dimensional representations of narrative context predict voxels? Our second objective is to determine if a shared space between the fMRI voxels and the semantic word embeddings exists which can be purposed to decode brain states into coherent textual representations of thought.

Methodology for Creating Semantic Context Embeddings

In order to featurize the descriptions of these datasets (the Sherlock annotations and the text of *Harry Potter* Book One Chapter 9), we use distributed word embeddings. We apply unsupervised learning methods to a large corpus like Wikipedia (and perhaps an additional corpus like the seven *Harry Potter* books) to construct semantic context vectors Y using global co-occurrence matrix factorization and sparse coding [37, 4, 5]. The matrix factorization step gives us low-rank semantic vectors whose geometry clusters similar words together and creates linear algebraic analogy relationships ("king" - "man"+ "woman" = "queen") as in Arora et al. (2015) [4]. Then by applying sparse coding to these distributed word embeddings, we get fine-grained 300-dimensional representations of specific word senses [5]. We also employ an empirical transfer-learning procedure from the atoms learned in the Wikipedia corpus to select atoms for use in the featurization of the Sherlock descriptions. We use a quality-thresholding method to identify which atoms to use in a given context, and calculate a weighted average to create a context vector for each time step.

Methodology for Analyzing the fMRI Datasets

We focus most of our attention on the **Sherlock** fMRI dataset [8], since the data has considerably more signal than the Harry Potter dataset, as we demonstrate in this thesis.

Our main algorithm is the unsupervised Shared Response Model (SRM) [9], which can construct a shared embedding space S_{fMRI} across the fMRI responses X_i for eight distinct brain region-of-interest (ROI) masks corresponding to various areas of the Default Mode

Network (DMN), Visual Cortex, and Auditory Cortex. We can apply SRM again between S_{fMRI} and Y to create a joint shared space S_{joint} between fMRI voxels and word embeddings (see Figure 1.1).

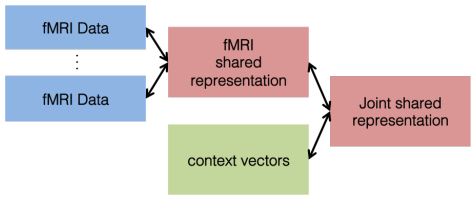

Figure 1.1: Two-Layer Semantic Shared Response Model

We can also apply ℓ_2-regularized linear regression (ridge regression) to fit linear maps between the original fMRI responses for individual subjects X_i and the semantic context vectors Y, or between the shared fMRI space S_{fMRI} and Y.

These models are validated with three procedures: context vector quality assessment, fMRI scene classification (also known as mystery segment classification), and fMRI reconstruction. We manually inspect properties of the context vectors after creation to determine whether they have any quality. Other than that, the context vectors are implicitly tested in all the models in which they take part. The scene classification task maps a scene from a held-out view of the stimulus into some shared space, and evaluates the top-1 correlation rank over all other scenes. Notably, this task is a harder generalization of the binary classification task of Mitchell et al. (2008) [34, 38, 46]. The reconstruction experiment evaluates how well our model predicts the actual fMRI response on a heldout set of time points. Of these tasks, the reconstruction task is most difficult because it is measured directly in terms of reconstruction instead of via a proxy where *enough* correlation will allow a model to do reasonably well.

The present work is similar in some ways to that of the recently published work of Huth et al. (2016) [24], which also seeks to map text embeddings from narrative stimuli to fMRI data. Our approaches to fMRI analysis primarily differ due to our use of the Shared Response Model to construct a shared fMRI space. Our semantic context vector construction creates embeddings into $\mathbb{R}^n$ which are both lower-dimensional and semantically

finer-grained than their 985-dimensional word vectors since we have a dictionary of atoms $\in \mathbb{R}^{300}$ which correspond to specific word senses. Moreover, our methods for semantic vector construction are theoretically justifiable. An additional difference is at the dataset level: We use Sherlock, our primary dataset, to analyze fMRI responses to an audio-visual movie with annotations describing unvoiced aspects of the scenes. In contrast, Huth et al. (2016) [24] analyze fMRI responses to auditory narratives for which the spoken text corresponds identically with the word embedding representations.

We construct similar spaces and maps using the Harry Potter dataset of Wehbe et al. (2014) [46], but as the experiments mostly failed on that dataset, the analysis of the dataset tends to focus on what properties of the Harry Potter dataset caused our methods to fail.

Results

Ultimately, our central positive result is the finding that it is possible to identify a semantically-relevant shared representation of fMRI response in an unsupervised fashion using views of multiple subjects watching the same natural movie stimulus. Using the shared response S_{fMRI} instead of individual subject responses X_i, we are able to significantly improve the prediction of voxel values from semantic word vectors Y which represent descriptions of the audio-visual movie *Sherlock*, as well as perform a mystery segment matching task with reasonable above-chance accuracy.

We also provide concrete evidence towards the hypothesis made in [24] regarding the existence of a shared fMRI representation across multiple subjects which correlates significantly with fine-grained semantic context vectors derived via statistical word co-occurrence approaches. Our use of multiple subject views of the movie data plays a great role in boosting the performance of our model and suggests that if the model in Huth et al. (2016) [24] was applied using multiple-subject SRM, their results would also improve. Since we use only semantic vectors to featurize a movie stimulus dataset, our work provides additional support for the notion that the distributional hypothesis of word meaning may extend to real life multi-sensory stimuli.

Chapter 2

Background Work

2.1 A History of Context

Many psychologists and neuroscientists have studied **context**, which is generally defined as slowly drifting information which persists over large time scales in the human brain. We can think of experiencing information at two levels: The stimulus, which is the new information presented at a given point in time, and the context, which represents an aggregation of previous experiences (perhaps within a certain time window). Previous models have attempted to explicitly model context as a running average of various features, which may be turned on or off with some probability at a given moment in time. Furthermore, the strengths of these associations between feature space and context space may be adjusted up or down depending on the coocurrence of features with similar context states. We can describe this view of information processing in the human brain in terms of time scales. The **time scale** is the duration of time over which a given set of information is "active." With this idea in mind, we see that the stimulus is presented under a short time scale (perhaps every second, we receive a new set of stimuli), while the overarching context persists at a long time scale [31]. This interaction may be described with a recurrent neural network applying a variant of Hebbian learning [23]. Various other approaches to explicitly modeling contextual drift are similar. Validation of these models often occurs via free-recall experiments, which

present subjects with lists of words which the subjects then attempt to recall [31].

A more data-rich environment in which to study context involves using immersive stories (in video, audio, and textual formats) as stimuli. Previous work by Hasson et. al. defined the **temporal receptive window**, which measures the dependence on the past of the neural activity in a given region of the brain [18]. Suppose we have a movie, which subjects watch in order. We observe the brain responses for each scene in a movie. Let us say we are investigating a particular scene S_t which occurs at time t. Now, we present the scenes of the movie in a different order. Suppose we keep the previous k scenes $S_{t-k}, \cdots, S_{t-1}$ fixed except for some scene S_{t-i} with $i < k$ which we swap with a different scene in the movie, R. Then, if the brain reponse at time t changes as a result, we know that S_{t-i} does contain contextual information about S_t, whereas if this is not the case, then we can conclude that S_{t-i} was not relevant in S_t's context. The largest i for which S_{t-i} is relevant is the temporal receptive window, or TRW [31]. The size of the TRW naturally provides a measure for the time-scale of various brain regions. Large TRWs correspond with long time scales, and short TRWs are essentially discrete stimuli. Given that stories have overarching narratives with recurring characters, one might hypothesize that regions of the brain with large TRW are important for processing the semantic context of a story. Various previous work suggests this hypothesis is true [18, 21, 39, 2, 47, 8, 41], and we further study the question in this thesis.

2.2 Identifying Regions of Interest and Pattern Analysis with fMRI Data

How can scientists study the living brain? Circa 2016, it is scarcely possible to measure every neuronal impulse in a living human being. For starters, in order to study human thought in natural settings (as natural as is possible inside a laboratory), it is necessary to avoid invasive measurements. Therefore, the methods by which we can attempt to study the living brain are limited to measurements of internal processes which allow for detection

outside the skull. Thankfully, these exist.

In fact, there are several such approaches. Electroencephalography (EEG) and magnetoencephalography (MEG) focus on deciphering the electrical and magnetic signals emitted by the brain, and have the desirable property of high temporal resolution (measurements are on the order of milliseconds). Unfortunately, the methods of measurement (placing various nodes on the human scalp a couple of centimeters apart) limits the spatial resolution of the data. Functional magnetic resonance imaging (fMRI), on the other hand, has high spatial resolution (down to the $1 \times 1 \times 1\text{mm}^3$ level is possible, though $3 \times 3 \times 3\text{mm}^3$ is more typical), but low temporal resolution (typically on the order of 2 seconds per brain image, or 2 seconds per repetition time (TR)). Thus we see a tradeoff between temporal and spatial resolution. The modality of measurement is often chosen to suit the question of interest, where usually either time resolution or spatial resolution is more important. Extremely high frequency response stimuli experiments will typically use EEG or MEG, while fMRI is suited for slow-moving stimuli and responses [30]. Of particular interest to this thesis is fMRI experiments where the stimulus is a natural movie. By analyzing inter-subject correlations for a temporally synchronized natural story view, it becomes possible to study brain behaviors which are typically not accessible in a lab setting [19]. Natural stimuli like audio-visual movies thereby enable neuroscientists and psychologists to study for the bridge between mental representations of semantic meaning and the way information is presented in the real world.

2.2.1 The Basics of fMRI

fMRI data is nontrivial to collect: The data consists of "a sequence of magnetic resonance images (MRI), each consisting of a number of uniformly spaces volume elements, or voxels, that partition the brain into equally sized boxes" [30]. Typical brain scans have on the order of around 10^5 voxels. A lot of noise is present due to the machinery used to collect fMRI data (hardware reasons), head motion artifacts, and other background noise due to scanner instabilities. Furthermore, it is necessary to consider the lag between stimulus and

response. Functional magnetic resonance imaging most often uses blood oxygenation level-dependent (BOLD) contrast to examine changes in the concentration of deoxyhemoglobin in the brain [30]. Essentially, changes in the flow of oxygen in the brain's blood induce different magnetic properties, and the different states also produce different local magnetic fields.

One prominent way of analyzing fMRI data is multi-voxel pattern analysis (MVPA) [36], which is essentially the idea that we can think of fMRI brain states as patterns to match to stimuli. It is therefore reasonable to apply traditional pattern-classification techniques. Benefits of the approach include a boost in sensitivity by looking at the contributions of multiple voxels. Of course, it is necessary that the voxels with signal be identified, or that voxel space be transformed into a space which has high signal. Therefore, identifying relevant subspaces of voxel space is important. In order to study the mental context with which humans view stories, it is essential to apply learning methods to determine the relationship between stimulus and fMRI space: Otherwise, there is nothing linking the information. In this thesis, we take the view that predictive power on held-out testing sets is indicative of signal.

2.2.2 The Default Mode Network

Another approach to identifying where to look for patterns in fMRI data is via regions of interest (ROI). ROIs are typically anatomically demarcated regions of the brain, identified by some brain atlas (a voxel map where each voxel is identified with a number corresponding to a certain ROI). In this thesis, we are particularly interested in regions of the brain which may be related to semantic context, and which are activated while participating in story comprehension. The default mode network (DMN) was identified in the 2000s as several small regions of the brain which correlated with each other. Fox et al. (2005) [12] were one of the first to identify the DMN as a brain network routinely exhibiting task deactivations. Fox et al. identify the posterior cingulate, medial and lateral parietal, and medial prefrontal cortex as being part of this network. The default mode network (refered to as the "task-

negative network" in [12]) was so named because it was observed that activation in this state occurred when nothing was happening: It is a resting state activation state not due to any low-level task like eye movements, or presence or absence of visual input [12].

Several experiments by the Hasson Lab revealed that the DMN and its subcomponents have the longest temporal receptive windows, and therefore have the longest temporal dynamics, at around the 1-2 minute level [18]. Successive works by Honey et al. (2012) and Regev et al. (2013) demonstrate that the DMN response is not due to low-level stimulus features (for instance, in the task of viewing a single word, a portion of the stimulus response may be dedicated to observing the curvy shape of the letter "o", which has nothing to do with the semantic meaning of "tool") [21, 39]. Ames et al. (2014) studies when the DMN response is reliable, and finds that when fMRI subjects have appropriate context for observing a stimulus, two substructures in the DMN network, the posterior cingular cortex (PCC) and medial prefrontal cortex (pFC) become more aligned [2]. They also conclude that when the stimulus is not understood in context, the DMN response is unreliable [2].

Simony et al. (2016) provides further support for the notion that the network configurations of the DMN are locked to particular narrative stories [41]. fMRI BOLD signal is recorded as subjects listen to a story read aloud. The strength of a particular network configuration is assessed via inter-subject functional correlation (ISFC), which looks at the correlation between different brain regions across different brains. The results indicate that scrambling the order of the narrative significantly decreases the reliability of finding the same network configurations across groups of people, and furthermore that the strength of DMN configuration during a given story scene predicts the memorability of that scene, as assessed by a memory test subjects took after the fMRI scan [41]. In this thesis, we use several DMN voxel-masks developed by Simony et al. (2016) for analyzing whether semantic embeddings of words can accurately predict the activity of the DMN and other related areas.

Yeshurun et al. (2016) demonstrates that if there are multiple interpretations of a narrative, it is possible to use the regions of the DMN to distinguish people following one

variant of the narrative from people following a different narrative of the story [47]. As we have seen, work prior to Yeshurun et al (2016) suggests that the higher-order DMN responses across individuals are similiar when people are exposed to the same natural narrative stimuli. However, the sensory parts of the brain also react similarly to these stimuli (since the raw images and sounds are processed by all subjects as well). If the DMN truly is related to the meaning of narrative, it should be possible to create different responses in the DMN if the implied meaning of the narrative stimulus is changed, while keeping the raw images and sounds constant. They hypothesize that introducing a context which could change a person's interpretation of a story should produce this effect [47]. To do that, they use as stimulus an auditory rendition of a 12-minute short story by J.D. Salinger, "Pretty mouth and green my eyes", which was designed by Salinger to be ambiguous with two completely plausible and yet drastically different interpretations. Yeshurun et al. (2016) separate the test subjects into two groups, where different background information (context) is provided to each group, intentionally conditioning each group for a different perception of the narrative. The results indicate that the magnitude of the difference in neural response in the regions of the DMN significantly correlates with the extent to which a subject interpreted a story as assessed by a post-experiment questionnaire assessing the subject's understanding of the story [47].

Chen et al. (2016) study the correlation between the fMRI representation of the experience and spoken recall of scenes from *Sherlock*, the BBC television show. The 48-minute story was divided into 50 distinct scenes. Many of these scenes in several of the DMN brain regions had particularly strong within-subject correlation between the original pattern formed by the experience of the scene and the pattern re-formed at the scene's later reinstatement, when the subject recalled the scene. Furthermore, activation patterns *across* subjects also had high correlation, suggesting that the DMN representations of the events of the *Sherlock* video were to some extent subject-independent [8]. The quantitative experiment was a two-group matching experiment: The test subjects were divided into two groups A, B of sizes 8 and 9 respectively, and the average PMC region of interest (a subset of the DMN) was calculated for all 50 scenes. Then, pairwise correlations between the group aver-

ages were calculated for all 50 scenes. For each scene s, if the correlation between the group A view s was most highly correlated with the group B view of s, the matching task was counted as a success. The overall accuracy was calculated as the proportion of scenes out of 50 marked correct. Then, this accuracy was cross-validated over all possible partitions of the subjects (A, B). Overall classification accuracy was 38.4% with $p < 0.001$ with chance at $1/50 = 2\%$ [8]. These results suggest that the DMN encodes semantic information about stories, and can be used for the purpose of decoding story narratives into text.

Given this history of results connecting the DMN to contextual and semantic meaning, this thesis primarily focuses on this region in its analysis.

2.3 The Representation of Language in the Brain

Several researchers have already attempted to find a relationship between word featurizations and fMRI activation in the brain. One popular method, started by Mitchell et al. (2008) [34], is to gather fMRI data across several subjects corresponding to stimuli related to some text: a noun, or perhaps a set of words, or even a story. Then, a linear map is learned from the word vectors to the fMRI activations on a training portion of the data. To test whether or not there is a significant relationship between the word embeddings and the fMRI voxels, a binary classification task is designed where two fMRI responses and their associated word embeddings from the testing data are held out: The classification task is to correctly match the word embeddings to the fMRI responses, which has 50% guess rate. Typically, the researchers also supply an accuracy rate at which the p-value is significant by which to compare the attained result [34, 45]. Notably, this task does not require a high degree of correlation in terms of actually **reconstructing** voxels: Only a little correlation is needed in several voxels for the binary classification to succeed. From there, it is possible to also identify voxels which fit particularly well, and analyze the produced brain maps to see which voxels encode what information about language. Another interesting application of the framework is to reverse the direction of the linear map to produce a brain decoding algorithm which outputs text given an fMRI input [38].

In a more recent line of work by the Gallant Lab at U.C. Berkeley, the goal is **reconstruction** rather than simple binary classification. The goal is to use semantic word embeddings to *predict* voxel activation, a considerably harder task [25, 24].

For an overview of distributed word and context embeddings, see Appendix A.

2.3.1 Binary Classification Experiments for fMRI-Language Maps

In the seminal Mitchell paper [34], the main contribution is the presentation of a computational model which predicts fMRI responses for concrete nouns (words like "dog", "cat", "chair"). The theory underlying the model is that the neural semantic representation of concrete nouns is related to the distributional hypothesis of meaning: Basically, brain vectors for concrete nouns should behave similarly to word vectors for those same concrete nouns in a huge corpus. This assumption is basically positing that we learn word meaning based on reading. The model is trained on a trillion-word text corpus (the Google 5-gram corpus from English web pages) and fMRI data observed after viewing a 58 concrete nouns from 12 semantic categories. For testing, the model predicts fMRI activation for words on a held-out set of size 2 and achieves highly significant accuracies. They also train competing computational models with different features for encoding meaning of concrete objects in the brain. The best model predicts fMRI activity to the degree that it can match words to their previously unseen fMRI images with accuracy far above chance. Thus there exists a direct predictive relationship between word co-occurence statistics and fMRI patterns associated with thinking about the word. The three central assumptions made by Mitchel et al. (2008) are as follows:

1. The semantic features that distinguish meanings of concrete nouns are reflected by their statistics of their use in a very large text corpus (specifically, for the $n = 25$ co-occurences the authors chose to record).

2. Different spatial patterns of neural activity are associated with different semantic categories of pictures and words.

3. The brain activity observed when thinking about a concrete noun is a linear combination of semantic feature values.

The first assumption is generally known as the distributional hypothesis of meaning, though its use here is more restricted since the authors only use the co-occurences of each concrete noun w with 25 verbs. The authors justify the second assumption by arguing that many linear models are used in the fMRI literature with the assumption that fMRI activation reflects a linear superposition of many sources. Furthermore this model allows the training data to determine the locations in the brain whose activity is affected by word meaning aspects, rather than making assumptions from neuroscience about which regions of the brain encode which aspects of meaning.

Training

Some notation first: Let n be the number of semantic features used to represent a word. Let m be the number of voxels in the brain. There are two steps to training. First, semantic features based on co-occurrence properties are computed from the very large text corpus. The second step learns weights for a linear combination of the semantic features to predict the activation at each voxel. Let $y(w)$ be the $m \times 1$ matrix of voxel activations for a given word w, C be an $m \times n$ matrix of coefficients to be learned, and $f(w)$ be the $n \times 1$ semantic feature encoding of word w. Then

$$y(w) = Cf(w) = \sum_{i=1}^{n} C_{*,i} f_i(w) \tag{2.1}$$

Here C is not dependent on a word w. We can interpret this equation in terms of the columns $C_{*,i}$ of C. By re-writing, we get that $\{C_{*,i}\}_{i=1}^{n}$ is a semantic image feature basis, with each image associated to a different semantic feature. In this paper, the semantic features are the co-occurence statistics of the input word w with 25 different verbs (accounting for different forms of the verb). The verbs correspond to basic sensory and motor activities, actions performed on objects, and actions involving changes to spatial relationships. For each voxel v, we learn the $1 \times n$ row vector $C_{v,*}$ of C with linear regression. Let the number of different

words be T. Let X be a $T \times n$ matrix where each row is $f(w_t)$ for $t \in [T]$. Let y_v be a $T \times 1$ vector where each entry $y_v(t)$ is the response for voxel v for word w_t. Then, for each $v \in [m]$ we find

$$\text{argmin}_{C_{v,*}} \|y_v - XC_{v,*}^T\|_2^2 + \lambda \|C_{v,*}^T\|_2^2 \tag{2.2}$$

which is solved by ridge regression. If the number of training examples is $< n = 25$, then there is no unique solution. In this case, adding l_2 regularization (i.e. ridge regression) gives a unique solution of least norm where $\lambda = 1$. After each $C_{v,*}$ is trained, we have the full predictor matrix C which given a word w and its featurization $f(w)$, we can use to predict the full fMRI response $y(w) = Cf(w)$.

Results

There were 60 randomly ordered stimuli (a picture of the object in white over black background) which came from 12 semantic categories (animals, body parts, buildings, etc.). There were only 9 human subjects, of college age. Each word-picture pair was presented 6 times. The representative fMRI response for each word was computed by averaging over the 6 presentations of word-picture pairs. The mean over all 60 presentations (one for each word-picture pair) was then subtracted from each presentation. A separate model was learned for each of the 9 participants.

Evaluation was performed with leave-two-out cross validation. That is, the model was repeatedly trained with 58 out of 60 word-fMRI image pairs, and tested on the remaining two. For testing, first a prediction of the fMRI image was generated for each of the two words, then these predicted fMRI images had to be matched to the correct fMRI image. This task was executed by comparing cosine similarity of the fMRI image vectors (where only a subset of the voxels were used). The subset of voxels was decided by calculating stability scores for each voxel: For each of the 6×58 presentations shown, there is a given fMRI voxel matrix. Then they calculated pairwise correlation across the 6 rows in the 6×58 matrix for each voxel, which assigns higher scores to voxels which exhibit consistent variation across the 58 images presented. Cross validated accuracies for each of the 9 models had a

mean of 77% accuracy, which is above chance (they claim an accuracy of 62% is statistically significant based on empirical accuracy distributions for null models). Another evaluation was performed to test whether the model could distinguish among a more diverse range of words. Here, the model was tested using a leave-one-out test where the model for each individual was trained on 59 words. Then, for 1000 additional words and the held-out word, an fMRI image was predicted. The 1001 words were then ranked by cosine similarity of their predicted fMRI to the true fMRI data for the held-out word. The average percentile rank was 0.72 across participants.

Mitchell et al. (2008) also manually examined the semantic feature signatures (think of $C_{*,i}$ for semantic feature $f_i(w)$): i.e., whether the predicted activations for various verbs matches the associations. They saw that activity in the gustatory cortex co-occurs with the verb 'eat', activity in motor areas co-occurs with 'push', strong activation in somatosensory cortex co-occurs with 'touch', and 'listen' co-occurs with activation in the language processing regions of the brain.

The authors also checked how accuracy varied over different feature sets. They tested 115 feature sets of 25 randomly drawn words from the 5000 most frequent words in the text corpus excluding the 60 stimulus words and the 500 most frequent words (i.e. containing 'the' and 'have'). The minimum and maximum accuracies of these random feature sets was 0.46 and 0.68, with the average of 0.60 and a standard deviation of 0.04. These results suggest that the hand-picked features do rather better than random. The success of the 25 sensory-motor specific verbs as a feature set suggests that neural representations of concrete nouns are in part related to sensory-motor features.

In 2014, Wehbe et al. [45] use a similar approach to decode arbitrary text passages in a chapter of a Harry Potter book [40]. This newer paper removes some of the early assumptions of [34] and attempts to generalize from concrete nouns to sentences with story structure. Their model is able to distinguish between which of two story segments (as opposed to which of two concrete nouns) is being read with 74% accuracy over 50% chance accuracy.

The Model and its Features

This model follows the same broad strokes as the model from the 2008 paper [34]. The setup is as follows: For nine individuals, fMRI activity was collected while each individual read the 9^{th} chapter of the first Harry Potter book. Reading was performed by having a single word appear at the center of the screen every 0.5 seconds (this format is known as rapid serial visual format (RSVF)). Note that each of the subjects was familiar with the Harry Potter story, had been recently updated on the contents of chapter nine, and had practiced RSVF on an unrelated story to the point where reading in this fashion was considered 'comfortable.' fMRI activity was collected every two seconds. Thus we have two time series, one of words and one of fMRI activity for every individual. To match the time series up, the word time series was chunked into groups of four words per TR for a time resolution of two seconds.

In the original paper [34], we essentially thought of the words w_t as a list of concrete noun examples presented in some order which did not matter. In this paper, the order in which the words presented does matter, and the authors take this into account in their model. Furthermore, each time step now consists of four words rather than one (so that text and fMRI time series are aligned). Thus, features $f_i(w_t)$ are now transformed into features $f_i(\{w_1, w_2, w_3, w_4\}_t)$ since features can be a property of each four-word chunk. For convenience we will refer to this as $f_i(t)$, the feature $i \in [n]$ at time $t \in [T]$. In the original paper [34], the features were word co-occurrences with 25 different verbs relating to sensory-motor activities. In this paper, the story features attempt to address multiple levels of representation. The types of features can be divided into four categories: visual features, semantic features, syntactic features, and discourse features:

1. Visual features are just the average word length in each TR and the word length variance in each TR.

2. Syntactic features are derived using an automated parser to get parts of speech for each word as well as dependency roles for each word inferred from a parse tree. There

are 28 part-of-speech relationships and 17 dependency relationships for a total of 45 binary features indicating if a given part-of-speech or a dependency occurred in a 4-word TR. An additional 46^{th} feature is the average position of the words in the TR in the sentence they belong in, numbered starting at 1.

3. Discourse features are derived from manual annotations going through story text. Pronouns are annotated with the character they refer to, and binary features are created for whether or not a certain character (one of 10) shows up in a TR. Frequent physical motions were chosen as well: These come with two values, a binary feature representing the start of the motion and a binary 'sticky' feature representing whether the motion is currently ongoing. Similarly, speech between characters is represented by a feature representing which character is speaking and a sticky binary feature indicating speech by a specific character is ongoing. There are also features for when emotion is mentioned and a corresponding sticky feature indicating an emotion is ongoing. For non-motion verbs (*hearing, knowing, seeing*), only a binary feature indicating that the verb took place is used, since these verbs typically do not last long enough to necessitate a sticky version of the feature.

4. Semantic features are most closely related to the features from [34]. They use non-negative sparse embedding (NNSE) to learn semantic vectors from a massive web corpora on which various dependency and document co-occurrence counts are computed. There are two co-occurrence matrices with different definitions of 'context': document counts are the number of mentions a word has in a particular document, and dependency counts are the number of times a word is in a given dependency parse link (e.g. word u is the subject of the verb "eat"). These dependencies are primarily verb- and adjective-related [13]. The co-occurrence matrices are factored using NNSE. to produce 1000 features of which this paper uses the top 100: these are essentially 100-dimensional word vectors. Since each TR has four words, they need a way to compose these word vectors: Their approach is to simply sum the features within each TR.

The semantic features in the model are derived from the Non-Negative Sparse Embedding (NNSE) algorithm [14]. Let $X \in \mathbb{R}^{w \times c}$ be made from c corpus statistics for w words (i.e. X is a word-context matrix). Then, NNSE produces a low-dimensional, sparse, non-negative latent representation using matrix factorization. The idea behind non-negativity is that you typically describe an object or concept by its positive facets; i.e. you say "an apple is a fruit" and not "an apple is not a tool". Sparsity is common to encourage only the most important features to have high weights. The NNSE objective is given by

$$\begin{aligned} &\text{argmin}_{A,D} \sum_{i=1}^{w} \|X_{i,*} - A_{i,*}D\|_2^2 + \lambda\|A\|_1 \\ &\text{s.t} \\ &D_{i,*}D_{i,*}^T \leq 1 \text{ for all } 1 \leq i \leq r \\ &A_{i,j} \geq 0 \text{ for all } 1 \leq i \leq w, 1 \leq j \leq r \end{aligned} \tag{2.3}$$

where the algorithm outputs the solution $A \in \mathbb{R}^{w \times r}$ that represents word semantics in r-dimensional space while being sparse and non-negative. $D \in \mathbb{R}^{r \times c}$, and note that $D_{i,*}$ and $A_{i,*}$ are row vectors of dimension $1 \times r$. Thus this program factors X to minimize reconstruction error using l_1 regularization for sparsity. This objective is not convex due to the fact that we are learning both A and D.

The authors make the following assumptions in the model:

1. They assume that each feature has a signature activity in each voxel which is consistently repeated every time the brain encounters this feature (and if a voxel does not encode this feature, the weight is 0).

2. The signature activity is scaled by the value of the feature at the time the feature is presented.

3. Total activation of a voxel is a linear combination of the feature values.

4. There is spherical Gaussian error in voxels with a different variance for each voxel. However, the variance for each voxel remains fixed over time.

5. The activity created by the feature is the convolution of the response signature with the time course of the feature. This convolution makes sense in the context of the hemodynamic response function (HRF) of the BOLD (blood-oxygen level dependent) signal, which fMRI measures. The HRF gives the activation curve for each voxel. While canonical HRFs do exist, literature has shown that the HDF is not necessarily uniform across the brain nor is it uniform across people. Thus learning weights for different time dependencies makes sense. Inspecting the weights over $k = 1, 2, 3, 4$ after training for each feature reveals these resemble the characteristic shape of the HRF at different points of the HRF.

6. Putting the last two assumptions together, the activity of voxel v at time t is given by

$$y_v(t) = \sum_{j=1}^{F} \sum_{k=1}^{4} f_j(t-k) c_k^{vj} \tag{2.4}$$

   Here we adopt the convenient notation that $F = n/4$, and c_k^{vj} is a special indexing of feature coefficients where we let k range from 1 to 4. In practice we stack on 4 additional feature columns per feature to our coefficient matrix C to represent the weights $c_1^{vj}, c_2^{vj}, c_3^{vj}, c_4^{vj}$.

Training

First we write down the training objective as for Mitchell et al. (2008): Note that the definitions are almost identical. Let X be the $T \times n$ matrix such that each row is the featurization of a different time step. Let C be an $m \times n$ matrix where m is the number of voxels in an fMRI scan and $n = 4 \times 195$ is the number of features, where the factor of 4 comes from time shifts. Then each row $C_{v,*}$ is the $n \times 1$ weight vector we learn for a given TR. Let y_v be the $T \times 1$ vector such that each entry is the response of voxel v at each TR t. Let $y(t)$ be the $m \times 1$ vector denoting activation at each voxel at TR t. Let $f_i(t)$ be the i^{th} feature at TR t and let $f(t)$ denote the feature vector of the four words at TR t.

This paper's model adds noise to the fMRI voxels. Let $\epsilon_v \sim \mathcal{N}\left(0, \sigma_v^2 I_T\right)$ be spherical

Gaussian noise with zero mean. Let ϵ be the $m \times 1$ random variable of Gaussian variables $[\epsilon_1, \cdots, \epsilon_m]$. As before, we learn a different C for each subject, and thus every parameter is learned differently for each subject. The only constant parameter is X, which represents the featurization of the Harry Potter text over time.

Thus, the equation to predict fMRI from the four words in a given TR t is

$$y(t) = Cf(t) + \epsilon \tag{2.5}$$

and the least squares regression objective is

$$\text{argmin}_{C_{v,*}} \|y_v - XC_{v,*}^T\|_2^2 + \lambda_v \|C_{v,*}^T\|_2^2 \tag{2.6}$$

Training separately for each row $C_{v,*}$ gives us the full matrix C. Note that this is simply ridge regression again, for which we are guaranteed to get a unique solution (the solution of least norm when $\lambda_v = 1$). l_2 regularization gives us the MAP estimator for least squares when we assume Gaussian errors. In practice, cross-validation is used on the training data to find the correct λ_vs. Ridge regression also results in effective automatic voxel selection since it learns high penalties for noisy voxels and small penalties for good voxels.

Results

They show that the predictions of the trained model are sufficient to distinguish between which of two previously unseen short passages is being read, given only observed fMRI activity. The first test task is analagous to the task from Mitchell et al. (2008)[34]. The trained model predicts the fMRI time series for two held-out story passages. Then it selects the passage such that the predicted fMRI time series is most similar in l_2 norm to the held out real fMRI time series. The results are cross-validated across all choices of the two held-out story passages. Random performance on this task is 50%. They attain an accuracy of 74%, which is significant with $p < 10^{-8}$. (p-values are determined by assuming the null hypothesis is 50% and then generating sample data and looking at the distribution of predictions for random weights). Notably, the authors also study to what extent the different features contributed to the accuracy. Without additional cross validation and voxel

selection methods, the semantic features (100 dimensions due to NNSE) only performs at 58% accuracy over chance, while the hand-crafted discourse features have a 84% success rate over chance [45]. These results suggest that a lot of the signal being captured is not due to the distributed embedding at all, but rather the hand-designed story-specific features.

The second test the authors run is to identify what type of information is processed by various regions of the brain. First, they effectively partition the brain into $15 \times 15 \times 15$ mm cubes corresponding to $5 \times 5 \times 5$ voxels (note there are typically on the order of 10^5 voxels, so this is about 10^{-3} of the full volume). They test every type of feature at every cube location to determine in which brain regions (the cubes) which types of features yield high classification accuracy. They found that the occipital cortex of the brain were strongly associated with the visual features (word length), as expected.As some examples of other results they find, they also see that the right temporo-parietal cortex is related to sentence length and the presence of dialog. Interestingly, the right temporo-parietal cortex has previously been shown to be more activated for better readers and is related to verbal working memory processes. The imagined physical motion of the story characters is found to activate in the posterior temporal cortex and angular gyrus, which agrees with neuroscientific knowledge. The identity of story characters is distinguishable by activity in the right posterior superior/middle temporal region, a region that has been found to encode facial identity. They also suggest they have found a partial answer to the question of whether semantic and syntactic properties of language are represented in different locations in the brain: For the semantic and syntactic features they use, there is a large overlap in some areas. They also find regions selectively processing syntax and semantics and that syntactic information is more widely and strongly represented (though this just may be due to the quality of the semantic features versus the synatactic features).

2.3.2 Decoding fMRI Stimuli Into Language

Pereira et al. (2011) [38] propose the inverse problem of the one solved by Mitchell et al. (2008) [34]. Instead of predicting fMRI given a word, given an fMRI response, they generate

text using a generative model (Latent Dirichlet Allocation, or LDA). They can generate a probability distribution over words pertaining to left-out novel brain images and that the quality of this distribution is measured quantitatively via a classification task that matches brain images to Wikipedia articles. The authors use the dataset from Mitchell et al. (2008) [34], which is fMRI data while subjects looked at both a picture and a word representing a concrete noun (e.g. *house*). From these fMRI images, the authors generated words pertaining to the relevant concept (e.g., *door, window, home*). Then the generated words are matched to corresponding articles from Wikipedia, providing a way of quantitatively analyzing the results.

A Brief Description of LDA

LDA (Latent Dirichlet Allocation) operates on a word-document co-occurrence matrix, placing documents in low dimensional space by taking advantage of sets of words which appear in multiple documents. Each dimension corresponds to a co-occurrence pattern (a topic word probability distribution). LDA is a generative model and allows you to interpret topic probabilities as the probability that a word came from the distribution of a particular topic. LDA models each Wikipedia article representing concept w as coming from a process where the number of words N and the probabilities of each topic being present θ_w are drawn. Each word u is drawn by selecting topic z according to probabilities θ_w, and then drawing from $\mathbb{P}\{u|z\}$, the distribution over words given topic z. θ_w is the featurization of each concept; i.e. $f(w) = \theta_w$. Since LDA places the topics in a simplex, the presence of some topics and detract from the presence of others.

Given a concept w, we can also induce a probability distribution for words u in w:

$$\mathbb{P}\{u|\theta\} = \sum_{i=1}^{|\text{topics in } w|} \mathbb{P}\{u|z_i\}\theta_w^{(i)} \tag{2.7}$$

Training

Pereira et al. use the following series of steps to arrive at their generative text model:

Table 1 | The top 10 most probable words according to each topic in the 40 topic model used in Figure 2A (topic ordering is slightly different).

Topic	Top 10 words
1	Plant fruit seed grow leaf flower tree sugar produce species
2	Color green light red white blue skin pigment black eye
3	Light drink lamp wine beer bottle water produce valve pipe
4	Drug chemical acid opium cocaine alcohol substance produce form reaction
5	School university student child education college degree state train Unite
6	Animal species cat wolf breed hunt dog male wild human
7	Water metal form temperature carbon process air element iron salt
8	Vehicle wheel gear car aircraft passenger speed drive truck design
9	Market party state country price government political trade people economic
10	Water ice rock river surface form sea ocean wind soil
11	Species bird egg fish insect female ant live feed bee
12	Language book write art century form story character word publish
13	War military force army weapon service submarine soviet world train
14	Blood cause disease patient treatment infection health risk increase pain
15	Church bishop pope catholic priest roman soap cardinal religious time
16	Cell muscle body brain form tissue human organism bone animal
17	Ship fish boat water vessel sail design build ski bridge
18	Iron blade steel handle head cut hair metal tool nail
19	Film image camera digital shotgun movie lens magazine rifle gun
20	Wear horse woman clothe saddle century dress fashion ride trail
21	Law state court legal police crime person act Unite criminal
22	Smoke chocolate light tobacco sign speed cigaret cigar state traffic
23	key lock switch machine needle tube bicycle type knit design
24	Card record information service company product datum process program credit
25	State cross head salute plate model symbol portrait scale circus
26	Love sexual god woman people pyramid death sex religion evil
27	Coin gold silver issue currency stamp state dollar value bank
28	Game play player ball team sport rule football hit league
29	Fuel engine gas energy power oil hydrogen heat rocket produce
30	Woman marriage god word christian child term jesus family gender
31	Fiber sheep wool cotton fabric weave hamlet pig produce silk
32	City build house store street town state home road bus
33	Tea tooth pearl kite shoe culture wear tattoo jewelry form
34	Earth sun star planet moon solar time orbit day comet
35	Material wood paint build wall structure construction design size window
36	Human social study people culture theory individual nature behavior term
37	Power station train signal line locomotive radio steam electric frequency
38	Food diamond cook meat bread coffee sauce chicken kitchen eat
39	Measure scale angle [formula theory object unit energy line property
40	Music instrument play string band bass sound note player guitar

Figure 2.1: A list of topics and their 10 most likely words [38]

1. First, from a corpus of 3500 Wikipedia articles about concepts deemed concrete or imageable (including 60 concepts from [34]), the authors created a topic model (latent factor representation using LDA) of each article, which represents the concept the article is about. This topic model effectively takes the place of the semantic features from [34] as an approximation of the mental representation of the concept. The authors ran LDA with the number of topics allowed ranging from 10 to 100 in increments of 10. The result is a representation of each of the 3500 Wikipedia articles in terms of the probabilities of each topic being present: We call these latent factor loadings. Each topic is a probability distribution over words.

2. They use ridge regression to learn a mapping from each topic/concept to a corresponding pattern of brain activation: This is equivalent to learning C before. The only difference from the formulation established from Mitchell et. al (2008) [34] is that the definition of $f(w)$ changes. Instead of using the hand-chosen verb co-occurences, Pereira et. al. use the topic probabilities describing the Wikipedia articles corre-

sponding to the concept w, which are the concrete nouns from Mitchell et. al (2008) [34]. The regression inputs are $f(w)$ and output is voxel activation y_v. The number of subsampled voxels used here is 1000 as opposed to 500 in [34]. As before, the fMRI images can be decomposed into a set of topic-specific basis images ($\{C_{*,i}\}_{i=1}^{n}$, corresponding to the semantic feature signatures in [34]). At this point we are still predicting fMRI image from featurized words. Now, from before, a probability distribution over a set of topics representing a concept induces a probability distribution on words for that concept, $\mathbb{P}\{u|\theta_w\} = \mathbb{P}\{u|f(w)\}$.

3. For brain images in a test set, the mapping can be used to infer a weighting over latent factors. The generative model from the first step can then be inverted to map from latent factors to text.

   In Mitchell et al. (2008), learning $f(w)$ given C ($m \times n$ matrix where m is number of voxels and n features) and $y(w)$ would not have allowed us to generate text, since $f(w)$ was just co-occurrences with certain verbs. In principle it could be possible to derive a probability distribution from these co-occurences: The authors would need to tabulate for each of the 25 verbs which words occur within a 5-word window of the verbs; this vector could then be normalized into a probability distribution. In this paper, $f(w)$ is a probability distribution over topics θ_w which **induces** a probability distribution over words u. Thus we can use fMRI images of concepts to produce words from that concept. We simply need to solve the convex optimization problem

$$\begin{aligned} &\text{argmin}_\theta \|y - C\theta\|_2^2 \\ &\text{s.t.} \\ &\theta_i \geq 0 \text{ for all } i \in [n] \\ &\sum_i \theta_i = 1 \end{aligned} \tag{2.8}$$

   C is fixed from the ridge regression and its columns are the basis fMRI images for each concept, y is the new image we want to infer the topic distribution for, and θ is

the distribution to infer. Recall that the number of features n is the number of topics. Let θ_y be the optimal topic probability distribution for a given novel fMRI image y. Now recall that for each topic z_i for $i \in [n]$, we have $\mathbb{P}\{u|z_i\}$ for word u learned from LDA.

$$\mathbb{P}\{u|y\} = \sum_{i=1}^{n} \mathbb{P}\{u|z_i\}\theta_y^{(i)} \tag{2.9}$$

Note the similarity to Equation (3): The difference here is that θ_y is inferred. Thus, by solving this convex program, we have inverted our map to estimate topic probability distribution θ from new unseen concept fMRI images y.

Results

First we note that the topic representations of the 3500 articles are sparse with respect to topics (though there are multiple topics in the representation of most articles). There is a link online to browse the 3500 concepts and topic distributions in detail. To objectively evaluate the quality of the generated text, 58 of the concepts are used for training and they test on the held-out 2 concepts. For the 2 concepts, we get the fMRI images and infer the topic probability distributions, $\theta_{y_1}, \theta_{y_2}$. The topic distributions are then matched with corresponding Wikipedia pages by using $\mathbb{P}_{y_1}\{u|\theta_{y_i}\}$ to determine which Wikipedia article is most probable. Note that random chance as accuracy 50% since we are pairing two fMRI images with the two held-out Wikipedia articles. In the majority of cases, classification was accurate.

When the two held-out concepts were in different semantic categories (i.e. *vegetable* and *car*), accuracy was on average over the nine participants was around 0.8, with a max of around 0.9 and a min of around 0.65. When the two held-out concepts were in the same semantic category, average accuracy over the nine participants was around 0.55, with a max of around 0.6 and a min of around 0.48. Note that these values are averaged over using different numbers of topics from 10 to 100, for every possible pair of two held-out concepts.

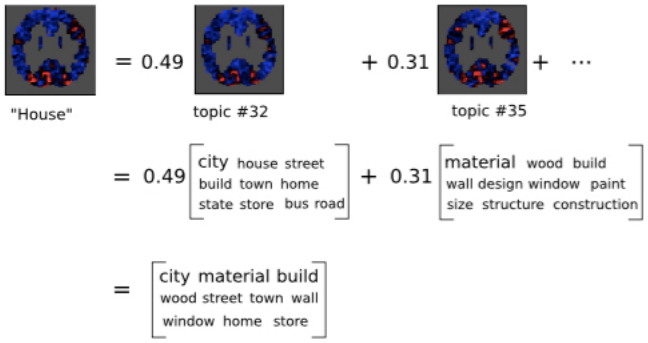

Figure 2.2: Visualization of weighted sums of latent factors [38]

The reason for the inaccuracy suggests text outputs for semantically related concepts are very similar, which is both good and bad: It suggests the model is not fine-grained enough (too bag-of-words-ish), or that the concept-representations (Wikipedia articles) are too similar intrinsically. They also saw that the voxels from the temporal and occiptal cortex voxels were the most stable across the 6 presentations of a concept to the fMRI subjects, suggesting that the learned fMRI basis associated with a topic is related to both semantic (word) and visual (picture) aspects of the topic.

2.3.3 A Joint Embedding Model for Language and fMRI

So far, we have discussed voxel reconstruction, a binary classification variant of brain decoding, and more genuine brain decoding into text [34, 45, 38]. We would also like to consider how we might find a shared space for both fMRI data and semantic word embeddings.

The Joint-NNSE algorithm intends to improve semantic word embeddings by utilizing information about the brain as input [14]. The purpose of the Joint-NNSE objective is simply to add an additional data source for a subset of the words in X. In the context of [14], the additional data is to be either fMRI or MEG data to encourage A to behave similarly in both the brain and word settings. Here, first re-order the rows of the corpus data X so that the first $1, \cdots, w'$ rows have associated brain recordings. Then let $Y \in \mathbb{R}^{w' \times v}$ be the data matrix of brain recordings, where v is the number of features associated with the brain data. Then let $D^c \in \mathbb{R}^{r \times c}, D^b \in \mathbb{R}^{r \times v}$. The JNNSE objective is given by

$$\begin{aligned}
&\text{argmin}_{A,D^c,D^b} \sum_{i=1}^{w} \|X_{i,*} - A_{i,*}D^c\|_2^2 + \|Y_{i,*} - A_{i,*}D^b\|_2^2 + \lambda\|A\|_1 \\
&\text{s.t.} \\
&D^c_{i,*}\left(D^c_{i,*}\right)^T \leq 1 \text{ for all } 1 \leq i \leq r \\
&D^b_{i,*}\left(D^b_{i,*}\right)^T \leq 1 \text{ for all } 1 \leq i \leq r \\
&A_{i,j} \geq 0 \text{ for all } 1 \leq i \leq w, 1 \leq j \leq r
\end{aligned} \tag{2.10}$$

where again we receive the output $A \in \mathbb{R}^{w \times r}$ is in a low-dimensional space while being sparse, non-negative, and representing word and brain semantics, since we have ensured that words represented in brain space must behave similarly by keeping A fixed across optimizations. Note that JNNSE can handle partially paired data, in comparison to Canonical Correlations Analysis (CCA) which requires fully paired data. In JNNSE, we only seek a solution keeping the transformed form fixed and maximally correlating the data reconstruction instead. In contrast, CCA maximally correlates the transformed form while keeping the input data fixed. This change allows the data to only be partially paired. The JNNSE objective is not convex due to the required alternating optimization. JNNSE also suffers from the weakness that there is only one other brain allowed in the model.

Fyshe et al. (2014) goes on to perform experiments using the data from [34] to demonstrate that JNNSE vectors are more consistent with independent samples of brain activity collected from different subjects for use as semantic features. Here, the authors train a linear predictor of semantic vectors given brain state vector, and use the predicted semantic vectors to see if the model can differentiate between two unseen words. This task inverts [34] and predicts a word from a brain state. Since semantic vectors have a sensible definition of similarity (unlike the verb co-occurrence semantic features from [34]), it makes sense to predict the word associated with the true semantic vector closest to the predicted one. On this prediction task with 50% accuracy, JNNSE achieves around 74% accuracy after cross-validating (this time only using 150 random pairs of hold-out words), which is on average 6% better than when using NNSE.

2.3.4 Assigning Semantic Meaning to Voxels

While the previous experiments primarily relied upon binary classification tasks to suggest that their results were statistically significant, the following work by the Gallant lab is able to demonstrate the ability to reconstruct voxels to a surprisingly high level of accuracy (namely, $0.3-0.5$ Pearson correlation; this can be interpreted as a cosine similarity measure) using semantic word embeddings. In this April 2016 work, Huth et al. have an fMRI dataset consisting of seven subjects who listen to over two hours of stories from *The Moth Radio Hour* as BOLD response was measured, with an entire 10-min heldout story for testing purposes [24]. The group takes a simplistic approach to word embedding, computing a word coocurrence matrix for the top 985 common English words (according to Wikipedia). For each word, they simply normalize the cooccurrence counts with these 985 common words to obtain a 985-dimensional semantic vector. Additional features to account for word rate and phonemes are added in the learning phase of the model, and are discarded after a map between the semantic vectors and the fMRI responses is learned. Temporally downsampling the word vectors at an appropriate rate allows for the pairing between words and fMRI images lined up by TR. A ridge regression extremely similar to the model used by [45] is then applied to learn the map between the vectors and the voxels. Then, they proceed to find a low-dimensional subspace by concatenating the maps they learned across subjects and performing PCA. They find that four dimensions explained a significant amount of variance, and project onto the word embeddings onto this space [24]. Then, they perform k-means clustering with $k = 12$ to identify distinct semantic categories, thus tiling the regions of the brain with these categories. They perform further refinement of their semantic voxel maps with a new algorithm called PrAGMATiC to create a "shared" atlas across people. Essentially, given the semantic map due to the principal component features for each person, PrAGMATiC shifts around the tiles across people to ensure the semantic tiles match up as good as possible.

This work is particularly interesting due to the very high voxel reconstruction performance directly from semantic vectors. However, the semantic vectors they use are rather

strange from the natural language processing perspective. LSA is a very old algorithm which has been shown to be outperformed in numerous papers using word embeddings. Moreover, the particular variant of LSA the authors appear to be using does not apply a dynamic range transformation of any sort, which has been previously shown to improve semantic embeddings (Arora et al., 2015). The authors try an alternative feature space for the words using word2vec ([33]), and find that they get slightly worse performance than with their 985-dimensional vectors. They conclude that perhaps 300-dimensions is not enough to capture semantic meanings present in the text and the 985-dimensional space has a richer representation of the stimuli, but it has also been shown by Arora et al. that lower dimensionality actually improves embeddings and that lower dimensionality acts as a denoising operation [4]. Moreover, it is very clear that the word vectors are very semantically rich, given their ability to decompose into fine-grained atoms of meaning [5]. Therefore, it remains a bit mysterious as to why the 985-dimensional vectors are able to perform so well at this task. Their results suggests that simple cooccurrence is what is being measured in the brain, since the 985-dimensional vectors of Huth et al. (2016) only rely upon this simplistic construction.

Chapter 3

Building Context Vectors from Natural Language

In this chapter, we explain how to construct semantic embeddings of context for several descriptive sentences worth of text. For now, let us imagine that a context is a few sentences. The idea is simple: For a given sentence, we would like to get low-dimensional vectors for which it is possible to recover as many as possible word descriptors of what is going on in the sentence.

3.1 Skipthought Vectors

One direct approach to modeling context vectors is to use the code directly from Kiros et al. (2015) [28]. Recall that this model learns vectors for sentences by applying the Skip-Gram approach to the sequence-to-sequence learning framework. In order to generate sentence vectors for a small corpus paired with fMRI images, we simply need to take the pre-trained model and continue training the model: The hidden state in the RNN after inputting each sentence is the sentence vector. The result outputs 4800-dimensional vectors which are supposed to be broadly applicable as features for generic NLP tasks.

We tried this approach on the Chapter 9 of the first Harry Potter book. However,

the heuristic quality of the vectors was not good, and the dimension was too high, so we abandoned this approach hereafter.

3.2 Corpus Size and Transfer Learning

In order to take advantage of word cooccurrence properties, it is necessary to examine a large enough body of text such that words of interest cooccur with enough other words to discern their relative meaning. One easily accessible such corpus is the data dump of the English Wikipedia. Arora et al. use this corpus to construct word vectors [4] as well as atoms of meaning [5]. They only construct word vectors for words which appear at least 1000 times so as to ensure the assumptions of their model hold.

This setup brings us to an issue when we consider constructing word vectors for a small corpus which has out-of-vocabulary words. In this thesis, two such corpuses are the textual descriptions of scenes in the Sherlock dataset and the text of *Harry Potter and the Sorceror's Stone*, Chapter 9. Both corpuses have very small vocabulary sizes (on the order of 2000 distinct words), and not many words overall. Moreover, if a word in this smaller corpus takes on additional meanings that are rarely or not at all present in the large corpus, then using the large corpus vector directly will cause issues in the smaller corpus.

3.2.1 Transfer Learning

This conundrum leads us to the notion of transfer learning. In order to augment the large-corpus semantics with small corpus semantics, we propose the following algorithm. Let the large corpus vocabulary be V_{big} and the small corpus vocabulary be V_{small}. Suppose the dimension of the original word vectors is d.

1. Initialize the word vectors as follows: If $w \in V_{big} \cap V_{small}$, then initialize the new word vector of w to be $\tilde{v}_w = [v_w \ \eta]$, where v_w is a d-dimensional vector learned from the large corpus and η is a κ-dimensional random initialization. Then the new dimensionality of our word vectors will be $d + \kappa$. If $w \notin V_{big} \cap V_{small}$ (i.e. w is out-of-

vocabulary), then initialize $\tilde{v}_w$ to be a random $(d+\kappa)$-dimensional vector with norm equal to the average of the norm of the large corpus word vectors. In our application, $d = 300, \kappa = 20$.

2. Run the Squared-Norm matrix-factorization algorithm on the small corpus using $\{\tilde{v}_w\}_{w \in V_{small}}$ as initialization. One can choose to either update the first d dimensions or leave them alone. Additionally, one can choose a weight η for the first d dimensions to prioritize them over the newly learned dimensions. These are parameters that must be empirically tuned.

For the case of the Harry Potter corpus, it is possible to take an intermediate step between the Wikipedia corpus and Chapter 9 of Book One. Transfer learning is particularly critical in the Harry Potter dataset since several words are made-up specifically for the world of the Harry Potter books. Moreover, there are distinct characters in the books who are referred to by common names (i.e., "Harry"). Therefore, we propose using all seven Harry Potter books as an intermediate transfer learning corpus to learn the senses of words like "Quidditch" and "Hogwarts" [40]. Then it is possible to use these word vectors for the Chapter 9 Book One vocabulary.

For the Sherlock annotation corpus, there is no intermediate corpus to transfer learn on. Because the annotation corpus only consists of 1000 sentences, this dataset is much smaller than the entire Harry Potter series. However, it turns out that the vocabulary overlap between the Sherlock corpus and Wikipedia is very small, excluding the characters introduced in the show. Therefore, it is possible to replace all out-of-Wikipedia-vocabulary terms with synonyms which are in the vocabulary of the Wikipedia corpus without changing the meaning of the annotations too much. We are careful to avoid using name-vectors from the Wikipedia corpus in place of character names. This problem arises since characters like Dr. Watson have generic first names: He is referred to as "John" throughout the Sherlock episode. In the Wikipedia corpus, John shows up in many places and has a whole host of word senses associated with it, most of which are not at all relevant to the character of John Watson. Therefore, we also attempt the transfer learning algorithm in this setting to

solve this problem, but since the annotation corpus is too small, this modification does not help matters.

3.3 Sparse Coding as Word Sense Filtration

We just saw that for small, specific corpuses like the Sherlock annotation corpus, large corpuses like Wikipedia often insert multiple, irrelevant senses into the word vectors. It would be very useful if we could somehow clean up these irrelevant senses automatically, leaving behind pristine word vectors for use in the smaller corpus. Our approach will be centrally based on Arora et al. (2016) [5]: We apply sparse coding to decompose the word vectors of a corpus into coherent, fine-grained atoms of meaning, which can then be sparsely linearly combined to create the original word vectors. This process involves setting a parameter for the desired number of atoms (for large corpuses, we use 2000 as the benchmark and for small corpuses, we use between 50 and 100). Then we learn five dictionaries for the same set of word vectors and prune bad atoms (those atoms which are not close to any word vectors). In the end, we combine these atom sets together. Note that we have the option of performing sparse coding on the word vectors of a large corpus, or on the transfer-learned word vectors of a small corpus. On the Wikipedia word vector set, 3-sparse coding with 2000 selected atoms results in 2607 atoms in total.

3.3.1 Subtracting the First Principal Component

If a corpus is big enough, performing sparse coding directly on the word vectors of a corpus results in several atoms which do not appear to mean anything. The nearest words to these atoms consist of vague, common words like “but” and “and”. We would like to remove these atoms *a priori*.

There are two approaches we can attempt to do this. For the first, consider the matrix of all word vectors of the vocabulary. Then, expressing this matrix via singular value decomposition as $U\Sigma V^T$, simply set the first singular value to 0 and perform the reconstruction. Then sparse-coding can proceed. Typically, we use a sparsity of 3 in our experiments. The

idea is that if the first principal component is small enough, you remove an "average" trend towards a generic atom. In practice, this operation works decently well on large corpuses.

However, if the corpus is small, then the top singular value is very large, and performing this operation results in the removal of almost all semantic content, and sparse coding gives nonsense results (this situation is precisely what occurs when we try to create atoms for the transfer-learned word vectors on the Sherlock annotation corpus). This situation motivates a slightly different idea for removing generic atoms. Instead of setting the top singular value to 0, we instead subtract the scaled top principal component of all the word vectors in the vocabulary from every word vector in the vocabulary. We perform the scaling by multiplying the normalized top principal component by the size of the average word vector. We can think of this vector as directly representing the "generic word vector". Moreover, subtracting by a single vector does not change the rank by very much. Using the linear algebraic properties of meaning, we can view this subtraction as removing the direction towards "genericness": We have found a translation which moves the word vectors away from the region of semantic space that is close to generic words. Empirically, this approach works better on both large and small corpuses, and after sparse coding produces more reasonable atoms for the Sherlock annotation corpus.

3.3.2 Vocabulary Subsetting and Manual Deletion

Despite our efforts in the previous section, the resulting atoms due to 3-sparse coding on the small corpus are not fine-grained and consistent enough for use. Therefore, we pursue a different approach: Manual removal of irrelevant or bad atoms. The strategy is as follows:

1. Prune the number of out-of-vocabulary words in the Sherlock vocabulary by finding synonyms. We also remove various numbers and so on which are irrelevant. Exclude character names; we will handle character appearances in future work. For now we wish to focus on distributional semantic meaning, and the Sherlock annotation corpus is too small to learn individual word vectors for each character.

2. Subset Wikipedia's vocabulary by the Sherlock vocabulary, and only consider atoms

which help explain words in the Sherlock vocabulary. This procedure reduces the number of atoms to consider from 2607 to 1550.

3. Examine the 20 closest words to each of the 1550 atoms. If they seem relatively coherent (to a human judge), and the atom seems either relevant to the Sherlock story or is general enough (i.e., "pulling, moving, dragging"), keep the atom. Otherwise, remove it. This pruning step reduces the number of atoms considered to 477.

4. Express each word in the Sherlock vocabulary as a weighted combination of the remaining atoms. This step means that the resulting word vectors have now lost their irrelevant senses for this corpus, and are finer-grained as a result. However, the previous step may have caused some words to lose all their atoms. We either replace these words with more synonyms to accomodate our new vocabulary (the words spanned by the 477 remaining atoms), or we ignore these words alltogether since the original word vector from the Wikipedia corpus had a nonsensical interpretation in the context of Sherlock.

It may be possible in the future to help automate the third step of this process by using a language ontology like WordNet [11]. Nevertheless, it seems a difficult problem to decide which atoms should be kept and which should be thrown away.

3.3.3 Application to Harry Potter Dataset

We would like to apply this approach to all seven Harry Potter books, as it would likely help create finer grained word vectors. It would be particularly helpful to figure out an automated method for performing this pruning, since it may take a longer time to evaluate the resulting word-vectors and atoms. For now, we do not use the approach outlined above, and as a result, have medicore word vectors for the Harry Potter dataset. We can visualize the correlation in a time-time correlation matrix, where we see that there is practially zero correlation involved.

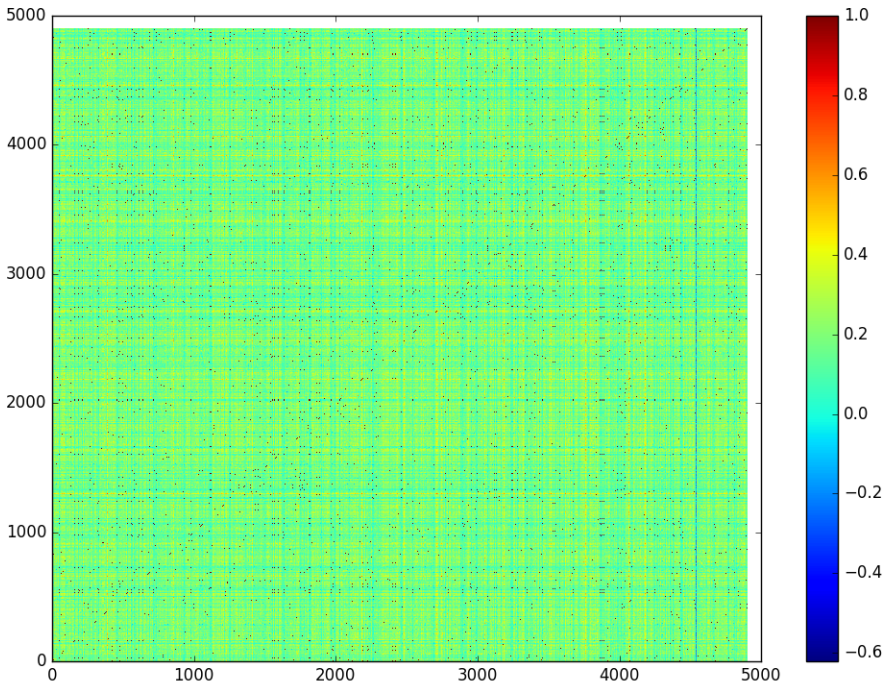

Figure 3.1: The Time-Time Correlation Matrix of the Averaged Harry Potter Context Vectors

3.4 Creating Contexts

Now that we have a bunch of coherent, relevant atoms of meaning, we have to organize the sub-second level descriptions of what is going on in the Sherlock movie into a single context per TR. Recall that for the Sherlock dataset, a TR is a 1.5 second fMRI snapshot of the brain. Some of the descriptions overlap across TRs, i.e., the TRs do not cleanly break up the annotations into chunks. When this issue occurs, we simply add the overlapping portion of the description to both segments. Finally, we have a set of words per TR, each of which is associated with ≤ 3 atoms with weights. These tuples of weighted atoms are now ready to be converted into context vectors via a variety of methods which we now outline.

3.4.1 Evaluation Methods

How can we evaluate the quality of context vectors? Throughout this section we will heuristically mention that certain approaches are good or bad. Here are the metrics we use to assess vector quality:

- Inspecting the time-time correlation matrix. That is, we take the context vectors at time points t_i, t_j and take the dot product $\langle c_{t_i}, c_{t_j} \rangle$ for all possible t_i, t_j. This method is handy for quickly checking whether something is clearly wrong with the word vectors (everything correlated, nothing correlated). See Figures 3.2 and **??**

- Inspecting nearby atoms and the nearby words of atoms to get an idea for how coherent the atoms are and how well the context vectors recover appropriate atoms.

- Hand-inspecting the atom weights for a given word vector: If a weight is negative, the meaning should be to some extent the opposite of the true meaning, if a weight is large, then the sense of the word should match the meaning of the atom very closely.

3.4.2 Averaging

For the Harry Potter words, recall that we have four words per TR. Following [45], we simply average the word vectors for each of the important words in the set of TRs. We can also double TR size to use more words for each context. In both cases, the performance is not so good: The context vectors have very little correlation with other context vectors, and are therefore rather meaningless.

In the Sherlock setup, we also simply average the word vectors which we refined in previous steps. This method results in too much correlation across word vectors of all times: The word vectors end up being close to everything.

3.4.3 k-Means and Principal Component Approaches

Another approach outlined in [5] is to represent context as a low-dimensional linear space. We can use PCA to identify the best approximation rank 3 subspace of the word vectors

present in the context. We can then take linear combinations of these principal components to describe what the hyperplane looks like. If we need a single vector to represent the context, we can always concatenate the vectors together in order of singular values.

We can also use a non-linear approach to get representative vectors. Consider performing k-means on the word vectors for a given context, and finding the closest atoms to each of the means. Setting $k = 3$ gives a similar situation to the rank-3 approximation of the word vector space. Again, we can concatenate the vectors to form a single context vector.

However, there is an issue with using concatenations of various vectors as a context: The ordering is somewhat arbitrary, and we would like to consider all cross-term dot products. A reasonable way to implement this condition for concatenated vectors is to look at $v^T A w$, where A is an interaction matrix that encodes some permutation of the vector pairs of $v = (v_1, v_2, v_3)$ and $w = (w_1, w_2, w_3)$ (i.e, $(v_1, w_1), (v_2, w_1), \cdots$). We would then need an approach to consider all possible combinations. In general, we would like to express a geometry between sets of vectors, but this essentially requires moving to tensor algebra. We adopt a simpler approach instead.

3.4.4 Truncated Weighted Sums

For each context, we have a list of tuples of atoms and weights, according to the words in the context. The absolute value of the weight determines how relevant that atom is to a specific word. Therefore, we propose setting a cutoff for the minimum weight an atom must have to be considered in the context. Then, we sort the atoms by weight and choose the top four atoms in terms of weight. Because we set a cutoff, there may be situations where no atoms meet the required criteria. In these cases, it is necessary to go back and add further description to the context using the 477-atom vocabulary to do so.

Having selected at most four top atoms, we take their weighted average and declare that the context. Empirically, this method works relatively well.

Some structure shows up when we look at correlation between word vectors. We also examine the individual contexts and see that the atoms which are near them are generally

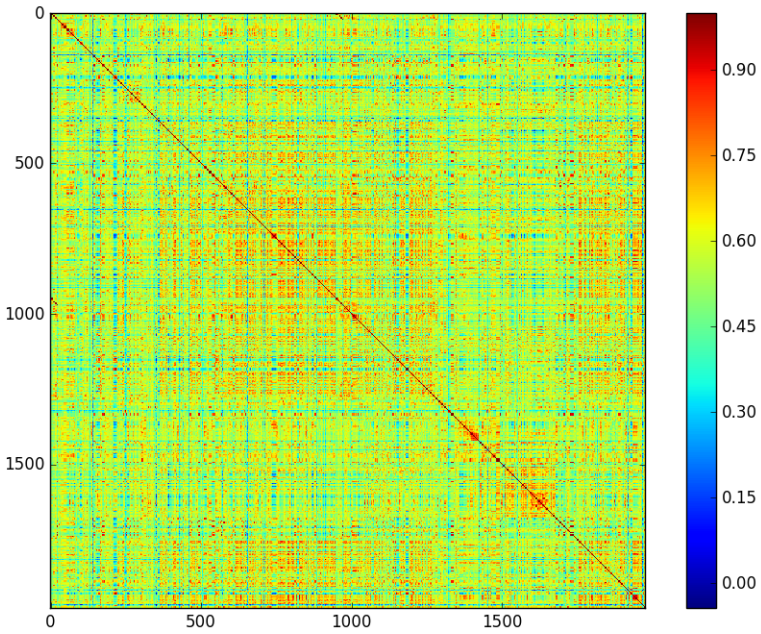

Figure 3.2: The Time-Time Correlation Matrix of the Top 4-Truncated-Weights Context Vectors. There are 1976 TRs and the vectors are 300-dimensional.

somewhat relevant.

TR #230: ['donovan', 'reporters', 'preliminary', 'investigations'], ['lestrade', 'distressed', 'donovan', 'suggest', 'suicide', 'can', 'confirm']							
Atom #1343 (corr. 0.78)		Atom # 1218 (corr 0.74)		Atom #111 (corr 0.71)		Atom # 1125 (corr. 0.70)	
investigation		murder		cnn		investigation	
investigations		murders		reuters		police	
investigators		suicide		jazeera		fbi	
investigating		perpetrator		msnbc		investigating	

Figure 3.3: The context at TR #230, and the top 4 atoms associated with the vector.

However, these context vectors are by no means perfect. The biggest issue is missing some crucial word in the sentences. The atoms are also not as fine-grained as would be desired. Future work in this area should investigate structural computational linguistics approaches for pruning and selecting the atoms to use in the context, perhaps by using some information about grammar and word order.

3.4.5 Sparse Atom Weight Vectors

We come up with another representation of contexts mostly for display. For each context, we create a 1550-dim vector where each index represents an atom. In the slots for the atoms which are activated by the current context, we put the associated atom weights. Thus, we have a 1550-dimensional sparse vector with which to view the evolution of context over time. The main idea this representation espouses is the notion that the atoms of meaning are an interpretable basis of context. The downside to this approach is that first of all, these sparse vectors no longer live in the word space and thus no longer have the desirable geometric properties of the word vectors and semantic atoms.

3.4.6 Aside: On Randomized Dimension Reduction

The word vector, atom, and context space we work in is typically 300-dimensional. However, it is easy to get a low-dimensional version of these vectors which preserves the distance geometry of the atoms.

Lemma 3.4.1. *Johnson-Lindenstrauss (*1984*) [1].*
The Johnson-Lindenstrauss lemma gives that for a set of points $x_1, \cdots, x_m \in \mathbb{R}^n$*, there exists a linear map* $f : \mathbb{R}^n \to \mathbb{R}^p$ *with* $p = \Omega\left(\frac{\log m}{\epsilon^2 \log(1/\epsilon)}\right)$ *such that for* $\epsilon \in (0, 1)$*, there exists a set of points* $f(x_1) = y_1, \cdots, f(x_m) = y_m \in \mathbb{R}^p$ *such that for all* i, j

$$(1 - \epsilon)\|x_i - x_j\|_2^2 \leq \|y_i - y_j\|_2^2 \leq (1 + \epsilon)\|x_i - x_j\|_2^2 \tag{3.1}$$

Therefore, supposing we want lower dimensional vectors to work with, we can always apply Johnson-Lindenstrauss and take a random projection to lower dimensional space until we find a set of points in the lower dimensional space satisfy the property of our desired ϵ. For instance, supposing $m = 500$ (as is roughly the case with the atoms) and $\epsilon = 0.5$, we have that $p \geq 9 * 4 = 36$ suffices for finding a projection which maintains pairwise distances with minimal decay in polynomial time. We can view this approach to dimension reduction as a bound on **worst-case** performance compared to the average case performance that performing low-rank SVD guarantees.

Chapter 4

fMRI Preprocessing and Quality Control

4.1 Region of Interest Masks

In our fMRI analyses, we first specify regions of the brain related to abstract meaning and context understanding. As per the Background Work section, the default mode network (DMN) has been shown to be very related to notions of story context understanding.

4.1.1 Sherlock Masks

We use several of the masks due to Simony et al. (2016) [41], visualized in Figure 4.1.

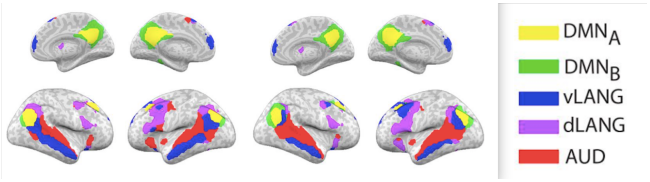

Figure 4.1: The DMN A, DMN B, Ventral Language, Dorsal Language, and Auditory Networks [41]

Two masks not in represented in Figure 4.1 are the occipital lobe and early visual cortex, both of which house several visual processing centers of the brain. We list all ROIs here:

- DMN A Network (2329 voxels)
- DMN B Network (1233 voxels)
- Ventral Language Network (2232 voxels)
- Dorsal Language Network (1412 voxels)
- Auditory Network (1189 voxels)
- Occipital Lobe (6474 voxels)
- Early Visual Cortex (307 voxels)

4.1.2 Harry Potter Masks

The Harry Potter dataset differs from the Sherlock dataset in that each subject has a different number of voxels. However, the data comes annotated with the locatoin of each voxel in a brain atlas, allowing us to subset voxels based on which region of interests they come from.

For comparison purposes and the reasons we mentioned in the previous section, we use the posterior cingulate cortex (PCC), the precuneus, the medial prefrontal cortex (PMC), the ventral and dorsal language areas DMN, which typically has between $12,000$ and $14,000$ voxels in the Harry Potter dataset.

4.2 Correcting for Noise Bias and Normalizing

fMRI data is inherently very noisy [30]. Therefore, it is necessary to identify sources of noise due to machinery, head movement, and so on, and remove them early in the analysis. These requirements mean that it takes a large amount of time to produce a good fMRI dataset. Often, fMRI studies are conducted over several sessions so that the human subjects may take breaks. It is necessary to correct for the beginning and ending of each of these sessions with appropriate experiment design (i.e., leaving a blank stimulus at the beginning and end of each session, and then removing these data points in the analysis).

4.2.1 How to Recognize Artifacts in the Data

A simple technique for recognizing overriding bias signals in fMRI data is to simply plot the ℓ_2 norm of the voxel activity at a given timepoint. If any strange periodic signals show up, it is an indication that there is some noise correction to be performed. We see in Figure 4.2 an example of periodic quadratic artificats which appear over the course of each fMRI session. These artifacts are likely due to bias from the machine itself, and should be removed.

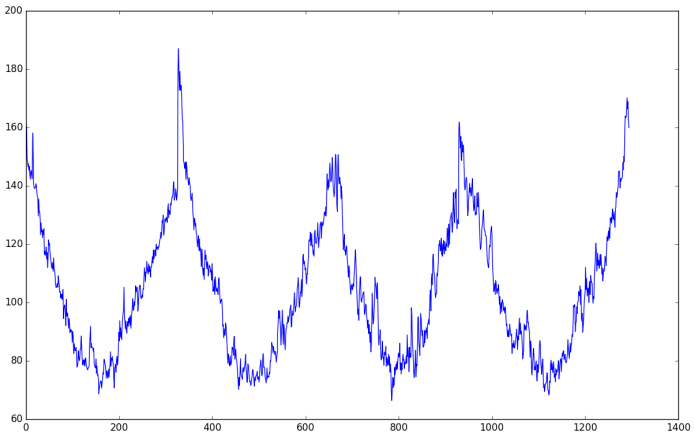

Figure 4.2: Norm of fMRI Voxel Activation Plotted over Time Pre-Noise Correction

4.2.2 Low-High Pass Filters and Polynomial Subtraction

A common strategy for removing such noise is filtering out frequencies of the data which one does not care about for a given experiment. For instance, it is known that the default mode network (DMN) has a low-frequency long temporal response. Therefore, it is reasonable to filter out high-frequency components. Another approach for removing such noise is polynomial fitting and subtraction. Since we suspect there is an additional signal present in the data unrelated to the fMRI BOLD response which appears over a very long time scale, we just fit this function and remove it. In the case of the Harry Potter data, it fits to a quadratic equation very closely, and we subtract this quadratic to get the noise-removed data.

4.3 Methods for Time-Aligning Stimuli and Responses

It is critical to line up the fMRI BOLD response with the stimulus which produced it. This task is not as simple as simply matching TR to TR, since the BOLD response is rather slow and there is a time delay between stimulus and response.

4.3.1 One-Time Shift

The simplest possible approach is to estimate a time-lag based on the average delay of the BOLD response to a stimulus (this is roughly on the order of $4-5$ seconds [30]). Therefore, just count out the number of TRs that comes out closest to this number of seconds and shift the stimulus input and response output accordingly by that number of TRs. We take this approach in featurizing the Sherlock data, and choose the number of TRs to throw out to be 3. We throw out the first 3 TRs of the response data (since it doesn't correspond to the stimulus), and the last 3 TRs of the stimulus data (since we never see the fMRI response). The advantage of this method is that it avoids modifying the original data and is simple to implement, introducing a small assumption about the relative positioning of the data.

4.3.2 The Hemodynamic Response Function (HRF)

The hemodynamic response to a single brief and intense neural event (imagine the Kronecker-δ function impulse) is referred to as the hemodynamic response function (HRF) [30]. The canonical HRF model suggests that the BOLD signal begins to increase about 2 seconds after the onset of the external stimulus, and peaks around $5-8$ seconds after the neural activity has passed (see Figure 4.3). After peaking, the BOLD signal goes below its baseline level for around 10 seconds [30].

Notably, this canonical hemodynamic response function is known to vary across individuals and even across different portions of the brain for the same subject. Therefore, in some models, instead of using the hemodynamic response *a priori*, researchers learn a subject-specific hemodynamic response [20].

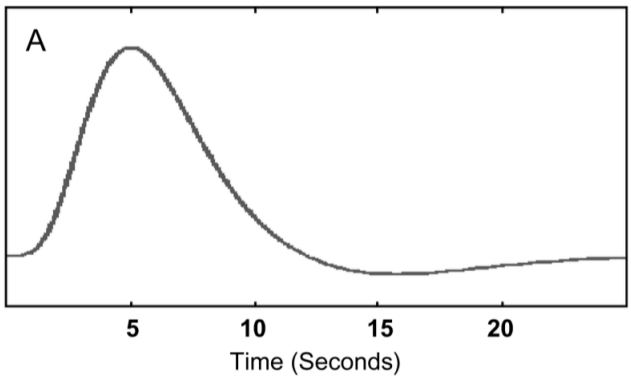

Figure 4.3: Shape of the Hemodynamic Response Function (HRF); picture due to Lindquist et al. (2008)[30]

Convolution by HRF

The hemodynamic response function provides a way to estimate the fMRI BOLD signal given a stimulus occurring over time. Given a neuronal spiking stimulus and a hemodynamic response function, convolving the HRF with the neuronal spikes produces a predicted fMRI BOLD response dependent on the specific HRF chosen. We can apply this idea to vector valued stimuli as well. If we convolve a known HRF treating the time-vector for each feature of the semantic vector as a neuronal stimulus, we are replacing the feature's time series with a predicted fMRI BOLD response based on that single feature as a stimulus. This convolution must be carried out in the discrete sense, where we sample a discretization of the HRF to get its values at the times of our TRs. We can then fit some linear model to the transformed stimulus [20].

Glancing at Figure 4.3, we see that the convolution will have the effect of pushing the main effect of the stimulus at the current time into the future by around 5 seconds, with some additional decay starting at around the 6 second mark. Thus, we recover the effect of ignoring the first few seconds of the response, and also gain the potential benefit of weighted smoothing transformation operating on the stimulus.

4.3.3 Learning a Convolution Operation

The approach taken by Wehbe et al. (2014) [45] as well as by Huth et al. (2016) [24] is to add additional regressors to their ridge regression models to learn parameter weights for stimuli for a window of length 4 TRs. That is, they are including interactions from the previous four stimuli in order to predict fMRI BOLD response: Effectively, they are learning their own hemodynamic response function for a small time window, which may be tuned across subjects and across voxel location in the brain. As a sacrifice, these models end up using more parameters.

We use this approach to perform time alignment between fMRI responses and semantic context vectors for the Harry Potter dataset. This model modification is relatively easy to accomplish, as all we need to do is modify the featurization of the stimuli: We simply concatenate the context vector at time t with the context vectors at times $t-1, t-2, t-3$, in order from most recent to least recent. If $t < 3$, we pad with zeros.

4.4 Quality of Individual Subjects' fMRI Responses

In this section, we consider approaches to assess the quality of an fMRI dataset. If basic properties do not hold, then it is entirely possible that the dataset had errors in its collection methodology or experimental design.

4.4.1 Approximate Rank of the Temporal Correlation Matrix

We are first interested in investigating the right dimensional space with which to view the fMRI data. How much low-dimensional structure is there, and can we take advantage of it via dimension reduction approaches?

An easy way to evaluate the linear dependence structure present within the voxels is to perform Principal Components Analysis on the data matrix X, which amounts to analyzing the energy distribution of the singular values of the voxel-voxel correlation matrix (which calculates correlations over time). If 80% of the variance is explained by the top 10 singular

vectors (or principal components), then the voxel space is low-dimensional, which means we should expect dimension-reduced versions of the fMRI data to work just as well if not better in fitting predictive models. This approach has the advantage of being easy to visualize as well. In Figure 4.4, we visualize the voxel-voxel correlation matrices for the DMN region of the first subject for the Harry Potter dataset and the DMN-A region of the first subject in the Sherlock dataset. It is obvious at first glance that the Sherlock dataset has a lot more structure than the Harry Potter dataset.

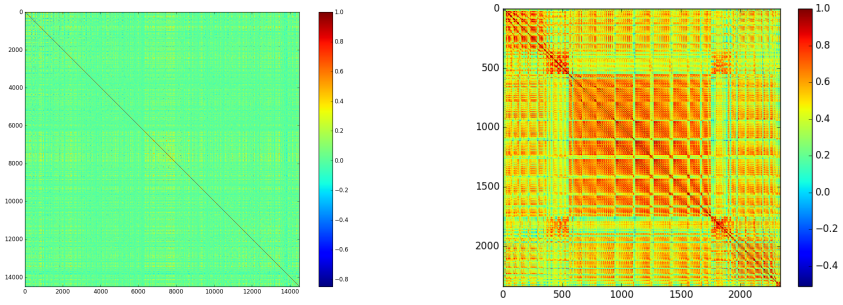

Figure 4.4: Voxel-Voxel Correlation Matrix for a Representative Subject in Harry Potter (left) and Sherlock (right)

The approximate rank of the Sherlock data is between 10 and 20 out of 1976 possible singular values for all masks: These singular values explains $75-90\%$ of the Frobenius norm of the voxel-voxel correlation matrix. On the other hand, the Harry Potter dataset requires 550 out of 1295 possible singular values to explain 75% of the norm, suggesting there is potentially a lot of noise in the Harry Potter dataset.

Nonlinear Dimension Reduction via Voxel Clustering

This visualization method suggests a nonlinear dimensionality reduction approach via performing voxel selection. Simply perform k-means clustering on the temporal correlation matrix. Then map each voxel to a supervoxel, i.e., the cluster which contains it.

4.4.2 Visualizing Timepoints which Correlate

We can also calculate the time-time correlation matrix (where correlations are spatial, and over voxels). This visualization is another approach to get an idea about the quality of an fMRI dataset. If the stimulus is a narrative, we expect to see blocks of correlation in "scenes" which are similar to each other. We can choose to analyze correlation along blocks, a task we refer to as "automatic scene detection", or we can choose to analyze correlation in the off-blocks, which is essentially "time-segment matching". In this thesis, we focus on the latter problem.

In Figure 4.5, we compare the time-time correlation matrices of a representative subject in both the Harry Potter and Sherlock datasets. Again, we find considerably more structure in the Sherlock dataset.

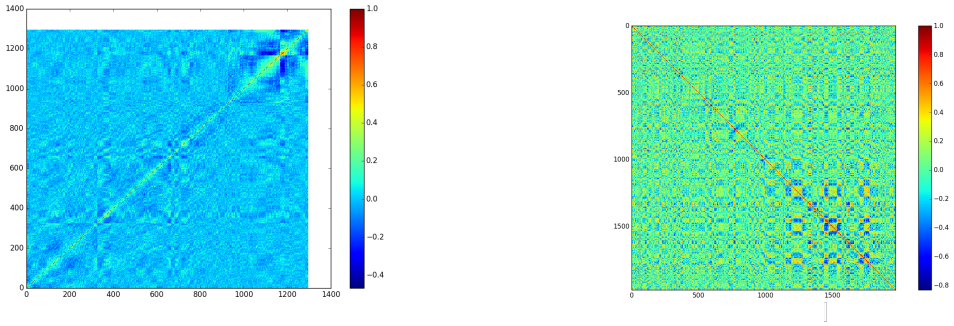

Figure 4.5: Time-Time Correlation Matrix for a Representative Subject in Harry Potter (left) and Sherlock (right)

Chapter 5

Shared Embeddings and Maps Between Language and fMRI

In this section, we study the central goal of this thesis: To learn relationships between fMRI spaces and semantic word embedding spaces.

5.1 Models

5.1.1 Ridge Regression

Ridge regression is one of the simplest modifications to linear regression. We use a similar approach to Mitchell et al. (2008), Pereira et al. (2011), and Wehbe et al. (2014) [34, 38, 46]. See the description of ridge regression in Chapter 2 for more details. We make the modification in that we do not learn a hemodynamic response function for the shift, but we rather just throw out the first 3 TRs of the fMRI data and the last 3 TRs of the semantic context vectors. Note that on the Sherlock dataset, this deletion corresponds to a 4.5 second time shift. This shift is incredibly important: If it is not carried out, then our experiments fail. In this work, we do not cross-validate over λ, and simply set it to 1 for comparison purposes.

5.1.2 Shared Response Model (SRM)

The Shared Response Model (SRM) (Chen et al., 2015)[9] is a probabilistic latent variable model for multisubject fMRI data under a time synchronized stimulus. From each subjects's fMRI view of the movie, SRM learns projections to a shared space that captures semantic aspects of the fMRI response. Specifically, SRM learns orthogonal-column maps W_i such that $\|X_i - W_i S\|_F$ is minimized over $\{W_i\}, S$, where $X_i \in \mathbb{R}^{v \times t}$ is the i^{th} subject's fMRI response (v voxels by T repetition times) and $S \in \mathbb{R}^{k \times T}$ is a feature time-series in a k-dimensional shared space.

The orthogonal maps W_i let us travel from the shared space S to a specific subject's response X_i. Since $W_i^T W_i = I$, we also have that we can go back from X_i to S via W_i^T.

In the probabilistic setting, we consider a shared latent variable at time t $s_t \sim \mathcal{N}(0, \Sigma_s)$, where $s_t \in \mathbb{R}^k$. Then, our probabilistic model for the response for subject i at time t $x_{it} \in \mathbb{R}^v$ conditioned on the shared feature vector s_t is given by

$$x_{it}|s_t \sim \mathcal{N}(W_i s_t + \mu_i, \rho_i^2 I) \tag{5.1}$$

where we require that $W_i^T W_i = I_k$ and we choose parameters μ_i, ρ_i, Σ_s. We furthermore assume isotropic noise for each subject (that is, we do not bias the voxels in any particular way). By conditioning x_{it} on s_t, and share s_t across all subjects i, we recover the notion that the shared vectors are in fact shared. Note that $X_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{it} \end{bmatrix}$, $S = \begin{bmatrix} s_1 & s_2 & \cdots & s_{it} \end{bmatrix}$. This probabilistic model for SRM can be optimized by a constrained EM-algorithm (Chen et al., 2015)[9].

Semantic Shared Response Model

The semantic shared response model is just two layers of SRM. The first layer learns an fMRI shared space S_{fMRI}, while the second layer learns a shared space between S_{fMRI} and the semantic context embeddings Y. We call the shared space S_{joint}. Note that for normal SRM, the inputs typically have similar rank since they are all fMRI data views of the same stimulus. For S_{joint}, however, this may not be the case. The extent to which

inputs having a similar rank is a necessary condition for success of the SRM model remains to be seen. Future work should check whether SRM performance suffers in situations where one subject's fMRI data has a drastically different variance structure. We hypothesize the answer is yes.

5.2 Experiments

5.2.1 Mystery Segment Ranking

In order to evaluate the quality of our shared spaces, we investigate whether mapping multiple views of the same stimulus to the shared space results in increased correlation between different views of related stimuli, and decreased correlation between different views of unrelated stimuli.

One experiment which captures the aim of our question is the mystery segment ranking task. Chen et al. (2015) introduces the mystery segment task as follows: First, train a shared response model $(\{W_i\}_i, S)$ on the first half of the Sherlock dataset (temporally). Consider the second half of TRs of the raw fMRI responses X_i. Map the second half of each of these responses using the pre-learned W_i^T. From now on, we will work in the shared space induced by projecting $W_i^T X_i$. Now, choose a holdout subject j. We come up with an average template for the other subjects by finding $Z = \text{avg}_{i \neq j}(W_i^T X_i)$ on the second half of the dataset. Similarly, we have separately $W_j^T X_j$. Now for every window $[w : w+r]$ of length r in the held-out part of the stimulus, calculate the correlation between $W_j^T X_j[w : w+r]$ and $Z[u : u+r]$ for every possible u, where $[u : u+r]$ denotes the values in the window of length r for some time point u. If $u = w$ maximizes the correlation, then the mystery segment task has correctly identified the matching stimulus in the average view Z for the heldout subject j. We can relax the notion of top 1 correlation to top-k, to allow for situations where there may be several reasonably similar stimuli in the dataset (Chen et al., 2015)[9]. Note that if our windows are allowed to overlap, then it is harder to identify chance level if we use top$-k$, since there are intercorrelations in the time series itself.

Scene Matching

The scene classification task maps a scene from a held-out view of the stimulus into some shared space, and evaluates the top-1 correlation rank over all other scenes. This task is essentially identical to mystery segment, except there are only 50 windows in the Sherlock dataset and they are pre-defined based on semantic content in the Sherlock movie stimulus and do not overlap. Training occurs on 25 out of the 50 windows, and we cross-validate over 10 random splits of scenes. We test on the remaining 25 scenes. Therefore, chance level is $1/25 = 4\%$ for this task. Instead of holding out a single subject as we did in mystery segment, we instead split the 17 subjects into two groups of size 8 and 9 and average the projected fMRI response in the shared space. Then we do the same task as before. Note that this scene matching test is very similar to the one used in Chen et al. (2016) [8].

Note that we can use this experiment for the ridge regression model as well. We consider the case where we fit a linear map from semantic context vectors $Y \to S_{fMRI}$ on a training portion of the data. Then, we can take the raw fMRI views from the testing part of the data from different subjects and map them into S_{fMRI} as an average. We also take the testing part of Y and map it into S_{fMRI} as the heldout "subject". Then, we perform the scene matching task in exactly the same manner.

5.2.2 Voxel Reconstruction

Voxel reconstruction is the task of measuring the predictive generalization performance of a linear map $f : Y \to X$, where X is some fMRI space and Y is word-embedding space, by looking at the magnitude of some distance measure $d(X, f(X) = \hat{X})$. The smaller the distance the better. This experiment is stricter than the Mystery Segment experiment for it measures overall correlation directly, while the Mystery Segment task can succeed as long as the correlation between correct pairs of scenes is larger than the correlation between incorrect pairs of scenes (the absolute magnitude may be low). A possible exception to this reasoning may be in cases where there are many similar scenes in the stimulus, in which the top-1 rank may overly harshly penalize the score since truly, any of the scenes would have

been a reasonable answer. This problem can be mitigated by considering the top$-k$ rank instead, where a scene pair (α, β) is deemed correct if α is in the k closest (by correlation) scenes to β and β is in the k closest scenes to α.

Evaluation Metrics for Voxel Reconstruction

Definition 5.2.1. Pearson correlation r.

For two vectors $x, y \in R^n$, we have

$$r = \frac{x^T y}{\sigma_x \sigma_y} \tag{5.2}$$

where σ_x, σ_y are the standard deviations of x, y. This can be thought of as a scaled measure of cosine similarity, which is identical when x, y have zero-mean.

Definition 5.2.2. Residual fraction f.

For two matrices X, Y, we have that the residual fraction with respect to X is

$$f_X = \frac{\|X - Y\|_F}{\|X\|_F} \tag{5.3}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix (i.e., $\|X\|_F^2 = \sum_{i,j} x_{i,j}^2$). In all cases, we choose $Y = \hat{X}$ for some prediction $\hat{X}$ of X. Then we define $f = f_X$.

In cases where we want to calculate Pearson correlation for matrix $X, Y \in \mathbb{R}^{m \times n}$, we unpack the matrices into a $(1 \times m \cdot n)$-length vector and apply the definition as before. Typically, y will be some prediction of $\hat{x}$. The correlation can be thought of as capturing the angle between the matrix-vectors in high-dimensional space. For high-dimensional vectors, the probability that two randomly chosen vectors are orthogonal is well-known to be closer to 1 the higher the dimension. Therefore, medium sized (say, 0.4) correlation can actually suggest a lot of reconstruction potential. The residual fraction is a more direct measure of how close a predicted matrix is to the true matrix. We normalize by the true matrix so that values of the residual fraction for reasonable predictions are ≈ 1.

ROIs [41]	Reconstruction Correlation	Residual Fraction
Ventral Language Network		
Auditory Network	0.90	0.45
DMN (A) Network	0.87	0.48
DMN (B) Network	0.89	0.45
Dorsal Language Network	0.84	0.53
Occipital Lobe	0.83	0.54
Early Visual Cortex	0.97	0.22

Table 5.1: Sherlock: Reconstruction of pure fMRI Heldout Data using SRM

5.3 Results

5.3.1 Performance of the Shared Response Model on pure fMRI

We would now like to evaluate the extent to which we can find a shared space for the fMRI responses of all subjects synchronized to the same temporal stimulus, like a movie.

Reconstruction

First, we report the Pearson correlation reconstruction results for each mask in the Sherlock dataset in Table 5.1. We first learn the maps W_i on the training half of the data. Then, on the testing part of the data, we project $W_i^T X_i$ into the shared space. In order to measure reconstruction, we evaluate $\text{corr}(\langle X_i, W_i W_i^T X_i \rangle)$. For the pure fMRI SRM, the correlations values are very high and the residual fraction is correspondingly low on the testing data.

The Harry Potter reconstruction performance is considerably poorer. Average testing residual fraction over subjects in the Harry Potter dataset is 0.83, which suggests that the difference vector between X_i and $\hat{X}_i$ is 0.8× the size of the vector we approximate.

Mystery-Segment and Scene Matching

For the Sherlock dataset, the results (Figure 5.1) show the averaged prediction accuracy over 10 random splits on movie scenes and 40 random left out subjects for each split. We note that peak accuracy (65%), which is fairly significant over chance of 2%, is achieved for only 20-dimensional shared space S.

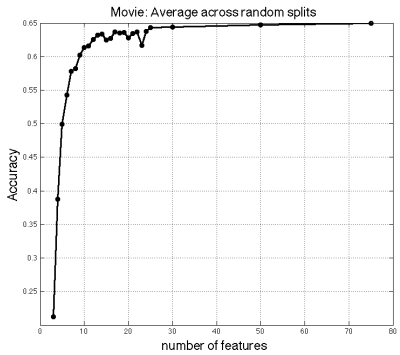

Figure 5.1: Sherlock Scene Matching: Scene Prediction Experiment with SRM Using a DMN ROI (image due to Janice Chen)

For the Harry Potter dataset, we split training and testing into the first and second half of the TRs. Each section lasted around 21 minutes. The top-1 variant of Mystery Segment performed relatively poorly when we choose a window size of 9 (this corresponds to 18 seconds): We got around 7% accuracy. However, as we increased the window size to $15, 30, 45, 60$ TRs, mystery segment performance increased to $14.5\%, 18.8\%, 22.9\%$. Note that 60 TRs corresponds to 2 minutes, which is already a rather long time. It may be more reasonable to use a longer time scale since we are using the DMN region of interest (which is known to have long time scales, see Section 2.2). However, Chen et al. (2015) are able to get 33% accuracy using only a window size of 9 inside the postcingulate cortex (PCC), a subset of the DMN which also should have a long timescale [9]. This fact suggests that our Harry Potter data has low signal.

5.3.2 Performance of Ridge Regression between fMRI Spaces and Semantic Context Vectors

Here we first note that the Harry Potter semantic context vectors had essentially zero correlation with the fMRI data on the testing part of the data. Since all results were equally bad, we do not mention them again. The average value of the residual fraction was around 1.13, and the average correlation was 0.

ROIs [41]	20-dim SRM	raw fMRI
Ventral Language Network	0.15	0.06
Auditory Network	0.11	0.05
DMN (A) Network	0.11	0.04
DMN (B) Network	0.08	0.03
Dorsal Language Network	0.10	0.03
Occipital Lobe	0.08	0.04
Early Visual Cortex	0.08	0.04

Table 5.2: Sherlock Ridge Regression Shared Space Reconstruction: Comparing $\text{corr}(\hat{S}, S)$ and avg. $\text{corr}(\hat{X}_i, X_i)$

Reconstruction

Here for the Sherlock dataset, we look at linear maps $f : Y \to S_{fMRI}$ and $f_i : Y \to X_i$ learned through ridge regression. The results are summarized in Table 5.2 for 20-dimensional SRM. We saw a considerable improvement in performance after using low-dimensional SRM, suggesting that SRM results in a semantically relevant low-dimensional space. Interesting, the ventral language network had the best performance across the ROIs. We ran the same experiment for 50-dimensional and 100-dimensional SRM, and noted that while we still saw a great boost in performance, the numbers were not quite as high as they were for the 20-dimensional SRM (perhaps 0.01 lower on average). Therefore, it seems that for these particular masks, the voxel-space of the ROIs is truly low-dimensional.

However, if we try to reverse the map with a linear map $g : S_{fMRI} \to Y$ to perform word decoding, the performance is considerably worse. Reconstructing the word vectors results in only 0.03 Pearson correlation. However, the fact that S_{fMRI} is 20-dimensional while Y is 300-dimensional implicitly imposes a constraint on the rank of the image of g: Therefore, we are implicitly learning a rank 20 approximation of the context vector space. Since this map does not fit very well, the approximate rank of the word vector space is higher than 20. Interestingly enough, when we calculate the low-rank approximations for several values, for all the different ROIs, we find that the rank 60 consistently explains 75% of the norm, despite the fact that there are different numbers of voxels involved. Therefore, it would be an interesting experiment to see if whether we learned 60 dimensional word vectors directly instead of 300-dimensional had any influence on the performance of $g : S_{fMRI} \to Y$.

ROIs [41]	Mystery Segment Test Accuracy
Ventral Language Network	0.16
Auditory Network	0.12
DMN (A) Network	0.20
DMN (B) Network	0.16
Dorsal Language Network	0.36
Occipital Lobe	0.08
Early Visual Cortex	0.0

Table 5.3: Sherlock Ridge Regression Top-1 Mystery Segment Accuracy

Mystery Segment

Here we only give results for maps from $Y \rightarrow S_{fMRI}$ in the Sherlock dataset, since the other ridge regression variants had rather poor performance. The results are summarized in Table 5.3. We have above-chance accuracy ($> 4\%$) for all masks except for the visual cortex, which had zero accuracy. Notably, the Dorsal Language Network performs the best with a 36% accuracy for top 1 scene classification. This means that using the map we learned on training data, both from SRM and ridge regression, we can map context vectors as well as original subject fMRI response into the same space and succeed at matching the context vector to the correct fMRI segment 36% of the time when we focus on the Dorsal Language Network. The DMN (A) Network has the second highest accuracy at 20%. Notably, these results are comparable to the Scene Matching task performed in Chen et al. (2016) [8], where the authors got 38.4% accuracy on a similar task which only dealt with multiple subject views instead of views from a different modality, i.e. semantic context vectors.

5.3.3 Performance of 2-Layer Semantic Shared Response Model

In this section, we again only discuss the Sherlock dataset results since the Harry Potter dataset results were insignificant. Since the performance was so poor, it was not worth checking the scene matching task at this point.

Reconstruction

For reconstruction of either S_{fMRI} or the semantic context vectors Y, the Semantic Shared Response Model performed a bit worse than the ridge maps between the raw fMRI activations and Y. That is, the reconstructions $S_{joint} \rightarrow S_{fMRI}$ and $S_{joint} \rightarrow Y$ both had Pearson correlation only around 0.02; i.e., essentially nothing. We speculate that it may be possible to improve performance if we are able to find a low-dimensional space for the word vectors.

5.4 Discussion

In this chapter, our primary positive result on the Sherlock dataset was increased correlation between the context vectors and the pure fMRI shared space, compared to the correlation between semantic context vectors and individual fMRI responses X_i. We were able to get 36% accuracy with the Dorsal Language Network, 20% accuracy with the DMN (A) ROI, and 16% with the Ventral Language Network and DMN (B) ROI over 4% chance at the Mystery Segment experiment by mapping semantic context vectors and raw individual fMRI responses into the same space, suggesting that SRM is able to act as a shared space for data views beyond the fMRI modality. The map from semantic context vectors into S_{fMRI} also performed alright at raw reconstruction capability, with a 0.15 correlation between prediction and truth for the Ventral Language Network and 0.11 correlation for the DMN (A) and Auditory networks. Interestingly, both Language Network ROIs performed the best at one of the tasks, and the Auditory Network and DMN (B) were also in the top 4 regions of interest. Notably, Early Visual Cortex got 0% accuracy on the Mystery Segment task, and the Occipital Lobe also poorly performed, suggesting that our semantic context vectors correlate better with the language, "meaning", and auditory portions of the brain than the visual, a not entirely-surprising result. Nevertheless, there is still some above-chance correlation with the Occipital Lobe, an area focused on vision.

On the other hand, we were not able to learn a map going the other direction from the fMRI shared space to the context vectors. Both ridge regression and SRM failed as

approaches here, suggesting there is something beyond the particular method of fit which is causing the difficulty. One candidate issue which comes to mind is the differing dimensionality between S_{fMRI} and the semantic context vectors. Since the semantic context vector matrix is roughly rank 60, it is possible to use the Johnson-Lindenstrauss randomized dimension redution method. We will attempt this setup in future work.

Despite the SRM improving our voxel reconstruction capabilities, when we compare our results to Huth et al. ((2016)) [24], we fall considerably short in terms of correlation: Our best voxel reconstruction correlation is 0.15, while theirs is roughly 0.6. They also have several voxels reconstructed with Pearson correlation in the range $0.3 - 0.5$ [24]. This shockingly good performance leads us to ask what aspect of our analysis prevents us from getting as high numbers: The dataset itself? The preprocessing of the data? The semantic vectors? The linear maps we learn? Some of this gap may be explainable by their task being slightly easier in a certain sense: Huth et al. (2016) have fMRI data which records brain response to hearing specific words; they then match the featurizations of exact words that they heard in story sequence with the resulting fMRI data. Our task is slightly more challenging, since we match *descriptions* of scene information and story content to a holistic fMRI response to visual and auditory elements.

In the future it would be desirable to apply their approach to our dataset to see what the critical factor in Huth et al.'s performance is. We suspect that regardless of their performance, their results may improve if they also used SRM instead of post-processing the individual maps they learned. Our elusive goal of true thought-decoding remains elusive, though we have taken steps towards results which suggest it is possible to do. Notably, we can pseudo-decode fMRI in the brain by way of the mystery segment task, though this hack is considerably less satisfying than finding an embedding from fMRI data into semantic space which generalizes well. Thus the question of Huth et al. (2016) that asks on whether "the contents of thought, or internal speech, might be decoded using these voxel-wise models" remains open [24].

Chapter 6

Future Work and Conclusions

The work presented in this thesis is ongoing; therefore, here we have enumerated several directions in which to consider proceeding.

6.1 Improving Word Context Vectors

One of the first tasks at hand is to further improve the semantic context vectors used in learned shared models. We desire that they be even more low-dimensional and fine-grained than they currently are. Furthermore, there is no negative correlation among word vectors, while there is negative correlation in parts of the brain. This tendency should be fixed.

Secondly, it would be very interesting to continue developing methods for automatically parsing story-specific information using the sparse coding approach. After creating the dataset consisting of all seven Harry Potter books, it seems worthwhile to experiment further with the NLP techniques to see if any results come out of it. Learning better semantic vectors would greatly help the performance of the various approaches to shared model learning.

6.2 Nonlinear Shared Models

Throughout this thesis we have solely considered *linear* models. It seems fruitful to consider whether nonlinear models can improve the generalization fit of the semantic word embeddings to the shared fMRI space. We would also like to see whether it is possible to construct a nonlinear shared space where different dimensionality features may be embedded. Here we propose two starting points, though the literature on nonlinear dimensionality reduction is quite vast.

6.2.1 Kernel SRM

A kernelized version of SRM exists: That is, consider fMRI vectors x, y. Suppose we would like to treat these vectors as living in some strange higher dimensional space; perhaps even a Hilbert space. Denote the mapping of x, y into this higher dimensional space by $\phi(x), \phi(y)$. Then, we have the property that $\langle \phi(x), \phi(y) \rangle = f(\langle x, y \rangle)$: That is, we can replace all inner products with a function of inner products and evaluate inner products in the higher dimensional space by only consider inner products in the current space. Since the objective of SRM can be expressed in terms of inner products, we are able to run SRM for nonlinear featurizations of the fMRI space. An interesting starting point is the quadratic kernel, which takes into account second order correlation information. Currently, no datasets have been found which drastically benefit from using a kernel, so it would be interesting to see if any ROIs in the Sherlock dataset could take advantage of this approach.

6.2.2 Convolutional Autoencoders

A yet more general nonlinear approach to finding a shared response is to use neural networks. An autoencoder is a nonlinear map from a higher dimensional space to a lower dimensional space back to a high dimensional space. In their simplest form, autoencoders act as a kind of low-dimensional approximation to input information, since you are forced to throw some information out when you pass through the lower-dimensional space. To create a shared space, simply force multiple subjects X_i to be reconstructed after passing

through a shared hidden layer. That way, the hidden layer averages out over subjects and sacrifices idiosyncratic terms in the individual X_i to learn an overall shared representation. Convolutional layers add constraints on the model, and act as small filters over the fMRI response. Convolutional neural networks have seen incredible successes in recent years especially in image processing and computer vision, so it is not too farfetched to think they would perform well at brain response featurization too. Learning a shared embedding space between fMRI response and language seems like a good idea to try, and is reminiscent of recent work performed by Johnson et al. (2015) [26].

6.3 Bootstrapping fMRI Decoders with End-to-End Image Captioning

We previously demonstrated that the multi-view SRM model produces a semantically relevant 20-dimensional space using views of multiple subjects watching *Sherlock*. However, our analysis omits the original visual and audio views of the movie to focus on the fMRI response and semantic word embeddings of scene annotations, and notably, this fact shows up in the performance of the Mystery Segment task where it appears the visual processing portions of the brain are not highly correlated with our semantic context representations. The next step is to mimic the image captioning system [26]. Their Dense-Cap model includes a CNN for images which feeds into an RNN language model for producing textual descriptions. Training the model end-to-end results in a shared space between images and semantic vectors. However, the current SRM cannot be simply incorporated into this end-to-end architecture. For this reason, the autoencoder variant of SRM becomes more attractive to try out, since it can more easily be included as a component of such an end-to-end architecture. Preliminary tests suggest that this new model performs well, but further testing remains to be completed.

6.4 Conclusion

Our primary positive result in this thesis is that the multi-view SRM model produces a semantically relevant 20-dimensional space using views X_i of multiple subjects watching *Sherlock*. This low-dimensional shared space S_{fMRI} is able to match fMRI responses to scenes with performance considerably above chance. We were also able to construct a 300-dimensional embedding of the semantic context induced by scene annotations Y. Finally, we showed that we can learn a linear map $f : Y \to S_{fMRI}$ such that $\text{corr}(S_{fMRI}, f(Y))$ is significantly larger than $\text{corr}(X_i, f_i(Y))$, implying the shared fMRI space better organizes the semantic meaning from individual fMRI views.

Of course, we also have many negative results: The Harry Potter dataset failed at most of our experiments, the 2-layer SRM did not work well at creating an embedding space for both language and fMRI voxel activations, and all current experiments have failed at constructing a map from the shared fMRI space S_{fMRI} to the word embedding space Y to facillitate thought decoding. However, there is reason to believe these failures may be due to the differing dimensionalities of these two spaces, and future work will focus on producing lower-dimensional semantic context vectors which have finer-grained lists of associated words to help mitigate this problem.

Appendix A

Embedding Words and Semantic Context in $\mathbb{R}^n$

A.1 Global Matrix Factorization Methods

Distributed embeddings for the semantic senses of words have been popular starting in the 1990s. Deerwester et al. introduced Latent Semantic Analysis (LSA) as a vector space model for language. The essential premise this work follows from is the **Distributional Hypothesis of Meaning**, which suggests that the meaning of a word is related to its co-ocurrence statistics with other words [43]. LSA considers words and "contexts" in which the words occur. For instance, a context could be defined as "within four words to the left or right of the word 'giraffe' ", or "whatever fills in the following blank: 'She was the best at __ in school.' ". Typically, context is taken to mean a document (as in topic modeling) or some text window of some radius. The radius can be one-sided (i.e., only consider co-occurrences to the *left* of a word) [43]. Thus, we see this natural language understanding approach to context relates to our understanding of a story context. Having defined a vocabulary for some body of text, and having defined our contexts, we can then proceed to ingest a corpus (for instance, the English language Wikipedia) and record the co-occurrence counts of each word in each context. After normalizing by row, we may then apply some

dynamic range transformation (i.e., take the logarithm) and proceed to take a low-rank approximation of the transformed matrix using singular value decomposition. Then, taking the resulting row vector for each word produces an embedding into the geometry of $\mathbb{R}^n$, where various distance metrics may be evaluated. As a result, these "word vectors" may now be used for various information retrieval and machine learning algorithms. This simple set of steps comprises latent semantic analysis [43]. Over the years, work in the vector space community continued, introducing probabilistic LSA and several other variants of the original algorithm. Most often, changes consider different transformations on the original values of the word-context matrix, like term frequency-inverse document frequency (tf-idf) and positive pointwise mutual information (PPMI). Distance metrics used to compare the resulting vectors also vary. Other variants of the basic LSA approach include the way these values are smoothed and sparsity constraints. Of course, the choice of context also remains variable, and some approaches constucted multiple word-context matrices, which are known as dual-space models. Other approaches model more complex interactions of context through tensors [43].

A.2 Local Context Window Approach

In the meantime, another line of work which continued throughout the 2000s was a neural network approach to featurizing language for machine learning algorithms. The original approach is due to Bengio in 2003, with the original Neural Network Language Model [7]. The goal of any language model is given a set of previous words, to output a distribution on the likelihood of the upcoming word. To perform this task, a single-hidden layer neural net is constructed where the inputs are a previous local window of text with the loss function defined by the softmax distribution $\frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_j}}$ over the possible vocabulary outputs y_j. In order to run this efficiently for large vocabulary sizes, it is intractable to simply encode each word in the vocabulary using the standard basis in $\mathbb{R}^{|V|}$, where $|V|$ is the size of the vocabulary. Instead, Bengio proposed to learn a representation of each word as a low-dimensional word embedding (both at input and output) and then to parametrize the

probability distribution in terms of the softmax of the image of the neural network function on these low-dimensional word embeddings. In the neural net architecture proposed, the single hidden layer can be though of as a combination of the input word vectors, and is thus a sort of semantic context embedded in real vector space [7].

Several variations of this approach occurred over the following decade, introducing Recurrent Neural Network (RNN) and Restricted Boltzmann Machine (RBM)-based models, both of which are considered "deep" in the sense that each used multiple layers of network to parametrize the functions which defined the language model. Ironically enough, the next big breakthrough which came in 2013 had *no* hidden layers. The word2vec model of Mikolov [33] is essentially a log bilinear model, where hidden layers are thrown away to speed up computation. Both an input vector v_{in} and an output vector v_{out} are learned for each word in the vocabulary. In the CBOW (continuous bag-of-words) model, the model attempts to predict the next word given the previous k words. Here, the probability of word w being the $k+1^{st}$ word is $\mathbb{P}\{w_{k+1}|\{w_i\}_{i=1}^k\} \propto \exp\left(\langle v_{w+1}, \frac{1}{k}\sum_{i=1}^k v_{w_i}\right)$. Comparing to Bengio's NNLM, we can think of the average as an alternative, non-hidden context vector. The fact that we use the moving average of word representations as a context vector also relates back to the psychological descriptions of context, as in [23]. In the more popular skip-gram variant of Mikolov's work, the training task is as follows: Given a word w, predict the words in a surrounding local context window of radius r. Given the input w, the probability of an output being word x $\mathbb{P}\{x|w\} \propto \exp\left(\langle v_w^{in}, v_x^{out}\rangle\right)$ (thus, log-linear). This objective is then trained using gradient descent. The final word vector for w is then typically the average $\frac{v_w^{in}+v_w^{out}}{2}$.

A.3 Explaining Analogy Properties of Word Vectors

Word2Vec became quite popular in 2013 due to the seemingly magical analogy property: For tuples of words like ("king", "queen", "man", "woman"), it turns out that $v_{king} - v_{man} + v_{woman} = v_{queen}$ [33]. In the following year, Pennington et al. came out with the GloVe model [37], which performed even better on various benchmark tasks than Word2Vec and

maintained the analogy property. GloVe looked back to the low-rank matrix factorization point of view of LSA, and saw that Word2Vec local context approach could be interpreted in the matrix factorization framework and thereby improved. However, theoretical justification for these methods was absent until 2015, when Arora et al. proved a theorem implying the analogy property follows from variants of a squared norm objective for weighted low-rank matrix factorization [4].

First, we represent our notion of an analogy mathematically. For the "man:king :: woman:queen" example, we have

$$\frac{\mathbb{P}\{\chi|\text{king}\}}{\mathbb{P}\{\chi|\text{man}\}} \approx \frac{\mathbb{P}\{\chi|\text{queen}\}}{\mathbb{P}\{\chi|\text{woman}\}} \tag{A.1}$$

This formulation is reasonably expressed as an objective that should be small:

$$\sum_{\chi} \left(\log\left(\frac{\mathbb{P}\{\chi|\text{king}\}}{\mathbb{P}\{\chi|\text{man}\}}\right) - \log\left(\frac{\mathbb{P}\{\chi|\text{queen}\}}{\mathbb{P}\{\chi|\text{woman}\}}\right)\right)^2 \tag{A.2}$$

where taking logarithms does not affect the relationship encoded - it is merely convenient to relate our measure to that of a standard procedure in building word-context matrices, applying pointwise mutual information (PMI) to every element in the matrix. Then we proceed to define a high-dimensional embedding of words into a vector space. Suppose we define the vector v_w for word w as being indexed by all contexts χ in which it appears, where $v_w(\chi) = \log\left(\frac{\mathbb{P}\{\chi|w\}}{\mathbb{P}\{\chi\}}\right)$ is $\text{PMI}(w, \chi)$. Therefore, in general for the $a : b :: c : d$ analogy:

$$\begin{aligned} \sum_{\chi} \left(\log\left(\frac{\mathbb{P}\{\chi|a\}}{\mathbb{P}\{\chi|b\}}\right) - \log\left(\frac{\mathbb{P}\{\chi|c\}}{\mathbb{P}\{\chi|d\}}\right)\right)^2 &= \sum_{\chi} \left(v_a(\chi) - v_b(\chi) - v_c(\chi) + v_d(\chi)\right)^2 \\ &= \|v_a - v_b - v_c + v_d\|_2^2 \end{aligned} \tag{A.3}$$

Note that taking logarithms of the probability quotients allows us to express our objective with simple vector addition and subtraction. In order to predict d optimally, we find

$$\hat{d} = \text{argmin}_j \|v_a - v_b - v_c + v_j\|_2^2$$

The equality in [A.3] only holds for the high-dimensional embedding we chose. However, the dimension of the vectors in word2vec is 300 - much smaller than the number of contexts χ. Arora et al. propose a model for low dimensional embeddings. For the simple case where the contexts are just single words, consider the PMI matrix $\mathcal{M}$ where $\mathcal{M}_{ij} = \text{PMI}(w_i, w_j) \approx v_{w_i} \cdot v_{w_j}$. The idea is that we want to express word vectors as a weighted low-rank factorization of $\mathcal{M}$ (low-rank to allow low-dimensional word vectors).

It turns out that this factorization is useful if the word vectors produced from the factorization are isotropic, and if $\mathcal{M}$ is close to positive semidefinite. The following theorem is proven in [4]:

Theorem A.3.1. *Log-Probability of Words.*
Assuming word vectors are drawn from scaled spherical Gaussian distribution $v \sim r \cdot \mathcal{N}(0, \sigma I)$ where r is a scalar random variable, and furthermore the partition function of the probability model is concentrated, then

$$\begin{aligned} \log \mathbb{P}\{w_1, w_2\} &= \frac{\|v_{w_1} + v_{w_2}\|_2^2}{2d} - 2\log Z + \log \binom{q}{2} \pm \epsilon \\ \log \mathbb{P}\{w\} &= \frac{\|v_w\|_2^2}{2d} - \log Z \pm \epsilon \end{aligned} \tag{A.4}$$

where d is the dimension of the space, q is the context window size, and Z is approximately the value of the partition function.

Note that the term $\log \binom{q}{2}$ directly follows from the fact that for two words to appear in a context of size q, there are $\binom{q}{2}$ possible locations, which is a multiplicative factor on the probability and thus an additive factor for the log probability. Let $\gamma = \log \binom{q}{2}$. These

together imply that

$$\begin{aligned} PMI(w_1, w_2) &= \log \frac{\mathbb{P}\{w_1, w_2\}}{\mathbb{P}\{w_1\}\mathbb{P}\{w_2\}} \\ &= \log \mathbb{P}\{w_1, w_2\} - \log \mathbb{P}\{w_1\} - \log \mathbb{P}\{w_2\} \\ &= \frac{\|v_{w_1} + v_{w_2}\|_2^2}{2d} - 2\log Z + \gamma \pm \epsilon - \left(\frac{\|v_{w_1}\|_2^2}{2d} - \log Z \pm \epsilon\right) - \left(\frac{\|v_{w_2}\|_2^2}{2d} - \log Z \pm \epsilon\right) \\ &= \frac{\|v_{w_1}\|_2^2 + \|v_{w_2}\|_2^2 + \gamma + 2\langle v_{w_1}, v_{w_2}\rangle}{2d} - \frac{\|v_{w_1}\|_2^2 + \|v_{w_2}\|_2^2}{2d} \pm \mathcal{O}(\epsilon) \\ &= \frac{\langle v_{w_1}, v_{w_2}\rangle}{d} + \gamma \pm \mathcal{O}(\epsilon) \end{aligned} \tag{A.5}$$

In order to prove Theorem [A.3.1], it is necessary to identify the probability model conditions under which we get the approximation $PMI(w_1, w_2) \approx \frac{\langle v_{w_1}, v_{w_2}\rangle}{d}$ so as to make the matrix factorization objective reasonable. The necessary conditions given in the theorem statement can be encapsulated by treating corpus generation as a dynamic process, where a new word is output at each time point t. We define the probability of word w being output at time t with respect to a unit ℓ_2-norm context vector $c_t \in \mathbb{R}^d$ called the **discourse vector**, which follows a random walk in the space. Note that this generalizes the notion of context from the CBOW word2vec model.

$$\mathbb{P}\{w \text{ emitted at time } t | c_t\} \propto e^{\langle c_t, v_w\rangle} \tag{A.6}$$

The idea of the discourse vector is that it represents the subject matter of the text at time t. Since the log probability is just a dot product over two vectors which we learn, this model is log bilinear, just like the word2vec model. By requiring $c_{t+1} = c_t + \eta_t$, where η_t is some random displacement vector, we recover the psychological notion of context as the slow drift of information introduced in a previous section. The context must drift slowly enough so that the partition function $\sum_w e^{\langle c_t, v_w\rangle}$ is nearly the constant Z (i.e., the random walk uniform over the unit sphere must mix quickly). We briefly note that the *maximum a posteriori* estimate of c_t is given by the c which maximizes $\mathbb{P}\{c|w_1, \cdots, w_k\} = \mathbb{P}\{c\}\mathbb{P}_{w_1,\cdots,w_k|c}\{=\}\mathbb{P}\{c\}\frac{e^{\sum_{i=1}^k \langle v_{w_i}, c\rangle}}{Z}$. Recalling that we draw c uniformly, we only need maximize the second term. By linearity of inner products, we have that

$c = \sum_{i=1}^{k} v_{w_i}$, normalized since we must have $\|c\|_2 = 1$ [4].

We can then make the following approximation using the dot products of the word vectors:

$$\begin{aligned}\text{argmin}_j \left\{\|v_a - v_b - v_c + v_j\|_2^2\right\} &\approx \text{argmin}_j \left\{\mathbb{E}[\chi] \cdot \|v_a \cdot v_\chi - v_b \cdot v_\chi - v_c \cdot v_\chi + v_j \cdot v_\chi\|_2^2\right\} \\ &\approx \text{argmin}_j \left\{\sum_\chi \left(\log\left(\frac{\mathbb{P}\{\chi|a\}}{\mathbb{P}\{\chi|b\}}\right) - \log\left(\frac{\mathbb{P}\{\chi|c\}}{\mathbb{P}\{\chi|j\}}\right)\right)^2\right\}\end{aligned} \tag{A.7}$$

by the definition of $\mathcal{M}$. Therefore, we may replace each dot product with the PMI. Thus Arora et al. give an explanation of why the vector addition approach approximates analogies for low-dimensional word vectors as well.

A.4 A Weighted Matrix Factorization Objective

We now define the **Squared-Norm Objective** for weighted matrix factorization, which can be derived by maximizing the log likelihood of the word cooccurrence distribution. Since we assume the random walk mixes quickly, the distribution is roughly multinomial [4]. We skip the algebra and approximations to give the definition:

Definition A.4.1. Squared-Norm Objective.

$$\min_{\{v_w\}, C} \sum_{w_1, w_2} X_{w_1,w_2} \left(\log(X_{w_1,w_2}) - \|v_{w_1} + v_{w_2}\|_2^2 - C\right)^2 \tag{A.8}$$

where C is a constant and X_{w_1,w_2} denotes the number of cooccurrences of w_1, w_2 in the same window. X_{w_1,w_2} can be replaced with the truncation $\min(X_{w_1,w_2}, \psi)$ for some constant ψ. This truncation is necessary for performance, and can be justified with the recognition that overly frequent words will obey the assumptions used in Theorem [A.3.1]. Note that this objective is essentially fitting the first equation of Theorem [A.3.1], while scaling the importance of the fit according to the number of cooccurrences involved. It naturally follows that an alternative objective would also be reasonable, replacing the log cooccurrence of

w_1, w_2 with $PMI(w_1, w_2)$ and replacing $\|v_{w_1} + v_{w_2}\|_2^2$ with $\langle v_{w_1}, v_{w_2} \rangle$, as per the conclusion following Theorem [A.3.1].

Both of the objectives in the preceding definition are versions of weighted SVD, which is an NP-hard problem. It is still possible to approximately solve these objectives using some form of gradient descent, and in practice this method works well.

A.5 Sparse Coding and Atoms of Meaning

Interestingly, the methods produced by the Squared-Norm objective have other useful properties. Consider a word with multiple meanings, for instance, the word "bank". A bank is a financial institution, it's possible to "bank" a shot off the board into the hoop in basketball, and the river "bank" is a place teeming with wildlife and plants, near the water. In a follow-up work, Arora et al. (2016) prove that word vectors with multiple senses learned by PMI factorization (see the previous section) are decomposable into weighted sums of the hypothetical word vectors for the various senses of the original word [5]. Furthermore, it is possible to recover these senses via sparse coding. That is, given n word vectors $v_1, \cdots, v_n \in \mathbb{R}^d$ and a sparsity parameter κ, we would like to find an overcomplete basis of vectors $a_1, \cdots, a_m$ such that we can write each v_w as

$$v_w = \sum_{j=1}^{m} \beta_{wj} a_j + \eta_w \tag{A.9}$$

where η_w is some noise term and $\#\{\beta_{wj} \text{ which are nonzero for } w\} \leq \kappa$. We learn $\{a_j\}_{j=1}^m$ as well as $\{\beta_{wj}\}_{w,j}$. The k-SVD algorithm can be used to optimize these variables and empirically achieves good results despite the original problem being non-convex [5]. We call the resulting $\{a_j\}_j$ semantic atoms. Investigating the nearby words using cosine distance results in sets of words which can be used to tag the atoms, which depending on the number and quality of the word vectors, can result in very fine-grained senses of meaning.

A.6 Paragraph and Sentence Vectors

We now backtrack a bit and recall where we left off with word2vec by Mikolov et al. [33]. While one can consider estimates of the context vector via averages of word vectors or the MAP estimate of the random walk context of Arora et al. (2015), other work has moved forward to attempt to directly estimate real space embeddings of sentences, paragraphs, and larger textual context groups. Le and Mikolov in 2014 came up with Paragraph Vector, an algorithm for learning feature representations for variable lengths of text [29]. The algorithm is very similar to the CBOW and Skip-Gram models of word2vec, and is also thus reminiscent of the work in psychology by Howard [23]. The first model, the Distributed Memory Model of Paragraph Vectors (PV-DM), is essentially the same as CBOW (learn low-dimensional inputs for the word vectors and use their average to predict the next word) with the addition of a "paragraph context" vector in addition to the words which gets shared across all words in the paragraph. The second model, Distributed Bag of Words (PV-DBOW), is more similar to the Skip-Gram model in word2vec, and uses a single paragraph vector to predict words in a small window for all words in the paragraph. Though the authors use the word "paragraph", this can really be applied to any size of text [29]. The best performance is obtained by combining the paragraph vectors learned by these two models via concatenation.

Another more recent approach to learning variable length distributed representations of text is the work on Skip-Thought vectors by Kiros et al. in 2015 [28]. This work is inspired by Skip-Gram as well as the sequence-to-sequence learning framework introduced by Sutskever et al. (2014) for machine translation [28]. The idea is to consider sentences as sequences of words, represented by unfolded LSTM networks (a specific model of Recurrent Neural Net). Then, much like in the Skip-Gram model of Mikolov et al. [33], the task is to predict the *sequence* before the current sequence and the sequence after the current sequence. The model is trained on a corpus of sentences from many kinds of books, and the gradients are computed via Backpropagation Through Time for the before-sequence and the after-sequence, in a manner identical to Sutskever's machine translation model. This

model has the advantage in that the sentence vectors produced as a result act as inputs to a decoder RNN, which can then spew out more sentences relevant to the input sentence of the vector. Considering this model from the cognitive science point of view is interesting because the RNN models sentence context directly as a form of semi-leaky memory due to the use of LSTM networks, which act as differentiable memory cells (one can learn when to keep or throw away information without experiencing technical issues like disappearing gradients or gradient explosion [28]).

Both of these models claim good performance on a wide variety of benchmark tasks, and suggest that they are good for use as featurizations in all sorts of NLP tasks.

A.7 Summary

In this section, we have overviewed a wide variety of approaches for encoding the meaning of words and semantic context in real vector space $\mathbb{R}^n$. The history of the approaches is relevant to our work because half of our task is based in natural language understanding: In order to construct maps between fMRI data and language, we need a method for featurizing the meaning of context in a space which has meaningful geometry.

References

[1] N. Ailon and B. Chazelle. Approximate Nearest Neighbors and Fast Johnson-Lindenstrauss Transform. pages 557–563, 2006.

[2] D. Ames, C. J. Honey, M. Chow, A. Todorov, and U. Hasson. Contextual Alignment of Cognitive and Neural Dynamics. *Journal of Cognitive Neuroscience*, 27:655–664, 2014.

[3] S. Arora, R. Ge, and A. Moitra. New Algorithms for Learning Incoherent and Overcomplete Dictionaries. 2014.

[4] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. RAND-WALK: A latent variable model approach to word embeddings. 2015.

[5] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. Linear Algebraic Structure of Word Senses, with Applications to Polysemy. 2016.

[6] C. Baldassano, D. M. Beck, and F.-F. Li. Parcellating connectivity in spatial maps. *PeerJ*, 2015.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[8] J. Chen, Y. C. Leong, K. A. Norman, and U. Hasson. Shared experience, shared memory: a common structure for brain activity during naturalistic recall. *(in review)*, 2016.

[9] P.-H. Chen, J. Chen, Y. Yeshurun, U. Hasson, J. V. Haxby, and P. J. Ramadge. A Reduced-Dimension fMRI Shared Response Model. *The 29th Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[10] H. Eavani, T. D. Satterthwaite, R. Filipovych, R. E. Gur, R. C. Gur, and C. Davatzikos. Identifying Sparse Connectivity Patterns in the brain using resting-state fMRI. *NeuroImage*, 105:286–299, 2015.

[11] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[12] M. D. Fox, A. Z. Snyder, J. L. Vincent, M. Corbetta, D. C. Van Essen, and M. E. Raichle. The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *PNAS*, 102:9673–9678, 2005.

[13] A. Fyshe, P. Talukdar, B. Murphy, and T. M. Mitchell. Documents and Dependencies: an Exploration of Vector Space Models for Semantic Composition. 2013.

[14] A. Fyshe, P. P. Talukdar, B. Murphy, and T. M. Mitchell. Interpretable Semantic Vectors from a Joint Model of Brain and Text-Based Meaning. pages 489–499, 2014.

[15] A. Fyshe, L. Wehbe, P. P. Talukdar, B. Murphy, and T. M. Mitchell. A Compositional and Interpretable Semantic Space. 2015.

[16] M. Hanke, F. J. Baumgartner, P. Ibe, F. R. Kaule, S. Pollmann, O. Speck, W. Zinke, and J. Stadler. A high-resolution 7-Tesla fMRI dataset from complex natural stimulation with an audio movie. *Nature: Scientific Data*, 1, 2014.

[17] T. B. Hashimoto, D. Alvarez-Melis, and T. S. Jaakkola. Word, graph, and manifold embedding from Markov process. 2015.

[18] U. Hasson, R. Malach, and D. Heeger. Reliability of cortical activity during natural stimulation. *Trends in Cognitive Science*, 14:40–48, 2010.

[19] U. Hasson, Y. Nir, I. Levy, G. Fuhrmann, and R. Malach. Intersubject Synchronization of Cortical Activity During Natural Vision. *Science*, 303:1634–1640, 2004.

[20] R. Henson and K. Friston. Convolutional Models for fMRI. In *Statistical Parametric Mapping*. Elsevier UK, 2007.

[21] C. J. Honey, C. R. Thompson, Y. Lerner, and U. Hasson. Not Lost in Translation: Neural Responses Shared Across Languages. *Journal of Neuroscience*, 32:15277–15283, 2012.

[22] T. Horikawa and Y. Kamitani. Generic Decoding of Seen and Imagined Objects using Hierarchical Visual Features. October 2015.

[23] M. W. Howard and M. J. Kahana. A Distributed Representation of Temporal Context. *Journal of Mathematical Psychology*, 46:269–299, 2002.

[24] A. G. Huth, W. A. deHeer, T. L. Griffiths, F. E. Theunissen, and J. L. Gallant. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532:453–458, April 2016.

[25] A. G. Huth, S. Nishimoto, A. T. Vu, and J. Gallant. A Continuous Semantic Space Describes the Representation of Thousands of Object and Action Categories across the Human Brain. *Neuron*, 76:1210–1224, 2012.

[26] J. Johnson, A. Karpathy, and F.-F. Li. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. 2015.

[27] A. Karpathy and F.-F. Li. Deep Visual-Semantic Alignment for Generating Image Descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[28] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-Thought Vectors. 2015.

[29] Q. Le and T. Mikolov. Distributed Representations of Sentences and Documents. volume 32, 2014.

[30] M. A. Lindquist. The Statistical Analysis of fMRI Data. *Statistical Science*, 23:439–464, 2008.

[31] J. R. Manning, M. J. Kahana, and K. A. Norman. The role of context in memory. In M. Gazzaniga, editor, *The Cognitive Neurosciences, Fifth Edition*. MIT Press, Cambridge, MA, 2015.

[32] J. R. Manning, R. Ranganath, K. A. Norman, and D. M. Blei. Topographic Factor Analysis: A Bayesian Model for Inferring Brain Networks from Neural Data. *PLOS ONE*, 9, May 2014.

[33] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. 2013.

[34] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just. Predicting Human Brain Activity Associated with the Meanings of Nouns. *Science*, 320:1191–1194, 2008.

[35] S. Nishimoto, A. T. Vu, T. Naselaris, Y. Benjamini, B. Yu, and J. L. Gallant. Reconstructing Visual Experiences from Brain Activity Evoked by Natural Movies. *Current Biology*, 21:1641–1646, October 2011.

[36] K. A. Norman, S. M. Polyn, G. J. Detre, and J. V. Haxby. Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *TRENDS in Cognitive Sciences*, 2006.

[37] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[38] F. Pereira, G. Detre, and M. Botvinnick. Generating text from functional brain images. *Frontier in Human Neuroscience*, 5, August 2011.

[39] M. Regev, C. J. Honey, E. Simony, and U. Hasson. Selective and Invariant Neural Responses to Spoken and Written Narratives. *J Neurosci*, 33:1597815988, 2013.

[40] J. K. Rowling. *Harry Potter Series; Books 1 - 7*. Scholastic, New York, 2007.

[41] E. Simony, C. J. Honey, J. Chen, O. Lositsky, Y. Yeshurun, and U. Hasson. History dependent dynamical reconfiguration of the default mode network during narrative comprehension. *(in review)*, 2016.

[42] D. E. Stansbury, T. Naselaris, and J. L. Gallant. Natural Scene Statistics Account for the Representation of Scene Categories in Human Visual Cortex. *Neuron*, 79:1025–1034, 2013.

[43] P. D. Turney and P. Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artifical Intelligence Research*, 37:141–188, 2010.

[44] Y. Wang, J. D. Cohen, K. Li, and N. B. Turk-Browne. Full correlation matrix analysis (FCMA): An unbiased method for task-related functional connectivity. *Journal of Neuroscience Methods*, 251:108–119, 2015.

[45] L. Wehbe, B. Murphy, P. Talukdar, A. Fyshe, A. Ramdas, and T. Mitchell. Simultaneously Uncovering the Patterns of Brain Regions Involved in Different Story Reading Subprocesses. *PLOS ONE*, 9, November 2014.

[46] L. Wehbe, A. Vaswani, K. Knight, and T. Mitchell. Aligning context-based statistical models of language with brain activity during reading. pages 233–243, 2014.

[47] Y. Yeshurun, S. Swanson, E. Simony, J. Chen, C. Lazaridi, C. J. Honey, and U. Hasson. Same story, different story: the neural representation of interpretive frameworks. *(in review)*, 2016.