Exploring Timescale in Language Comprehension with EEG

by

Sijie Ling

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Psychology
University of Alberta

Abstract

As we listen to spoken language, the brain performs multiple levels of computation, from understanding individual words to comprehending the arc of a story. Recently, computational models have been developed that also process text on multiple levels. These models, called multi-timescale long short-term memory (MTLSTM) models, use information from different timescales to predict the next word in a sequence. However, the link between these MTLSTMs and the brain has not been explored. Here, we use electroencephalogram (EEG) recorded [1] when subjects (n=19) passively listen to the first chapter of *Alice's Adventures in Wonderland* by Lewis Carroll. We train ridge regression models that use patterns in the EEG to predict the different timescales of an MTLSTM model [2, 3] processing the same text. We find that segments of EEG signals can reliably predict the MTLSTM semantic representations of different timescales. For long timescales, the prediction accuracy is significant for most of the -2s to 2s window surrounding the onset of a word. For short timescales, prediction is significant in a short period around the onset of a word. We also observe reliable predictions for the short timescale at time points distant from word onset (-1s, 2.2s). This indicates that the timescales of the MTLSTM model have a connection to language understanding in the brain, while the brain has a complicated strategy, including anticipating and recalling short timescale information. The findings of this work give insight into the brain's timeline of efficiently managing different types of information. Additionally, they indicate the similarities and differences between the computational models and the brain in language processing.

Keywords: EEG, LSTM, Machine Learning, Psycholinguistics

Preface

This thesis is an original work by Sijie Ling. No part of this thesis has been previously published.

Acknowledgements

I would like to thank my supervisor, Dr. Alona Fyshe, for her excellent support, encouragement, and mentorship, especially when all the research activities were online in my difficult first year.

I would also like to thank my lab mates. They gave me a warm welcome when I came to Edmonton and helped me to get accustomed to living in another country. They also shared their insightful thoughts and useful resources in academic discussions.

I am grateful to my friends who always supported me. When I was depressed about continuing issues, they encouraged me to become confident and overcome the difficulties.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Language comprehension is one of the most prominent functions of the human brain. However, the function is intricate because it requires multiple levels and various kinds of computations working together as a whole. When listening to speeches, stories, and dialogues in daily life, the human brain interprets the meanings of single words, constructs them into sentences with grammar, and extracts the substance of the paragraph. Considering the timeline of processing, the brain continuously recalls the previous text and predicts the incoming text. But how do all these computations perform in the brain? How do they cooperate to make the brain efficient in processing the countless words we encounter daily? For nearly a century, studies in psycholinguistics have proposed models to explain how the human brain processes language. Most of these studies, however, focused on narrow aspects of language function and explained one aspect at a time. Due to the diversity in experiment design and modeling, it is difficult to synthesize these conclusions and compare the models for different levels.

At the same time, with the development of artificial neural networks (ANNs), computer scientists have begun to use a different method to study language. Unlike in psycholinguistic experiments, in ANN language models, few assumptions about language are integrated into the model. Instead, computer scientists have designed many tasks that are based on language usage (e.g. summarization, translation, or question answering [4]). To achieve better performance in these tasks, the model

follows a strategy to learn from a corpus and adjust its parameters. After training, the model becomes a general model that has remarkable human-like performance in many language tasks. Studies in recent years [5, 6] have found that the intermediate representations of these models have similarities with brain activities when a person processes the same text. These studies ushered in a new method to study the language function of the brain.

In this thesis, we explore and compare the timeline of the processing of information at different timescales in the brain. This work is based on an ANN called a multi-timescale long short-term memory (MTLSTM) model [2]. In previous studies [3], researchers predicted fMRI responses with the information of different timescales separated by the model. From the prediction performance, they displayed the brain areas that are sensitive to short and long timescales of language processing. We extend this work by using small windows of EEG signals to predict the extracted timescale information and construct brain processing timelines for different timescales. These timelines, together with brain areas, give us insight into how the brain executes the language function as a whole.

1.1 Contributions

The core findings from the research provide evidence that:

- When subjects are hearing natural text, the EEG signals recorded around the onset of a word can predict word semantics.
- The EEG signals can predict information of different timescales. The time range around the onset for processing information is wider for longer timescales and narrower for shorter timescales, but short timescales can be predicted as early as before word onset.
- The brain predicts the incoming word before the onset and does further semantic processing after the onset.

- Surrounding the onset, the whole brain is involved in semantic processing while the right temporal lobe and prefrontal cortex mainly participated at distant time points.

1.2 Thesis organization

In the next chapter, Chapter 2, we summarize previous studies in cognitive neuroscience and language models that set the foundations for the research. In Chapter 3, we illustrate the details of the experiment design, including the construction of data sets and the training and evaluation paradigm for prediction models. In Chapter 4, we report the results and discussions of multiple analysis in prediction models. Finally, in Chapter 5, we conclude the thesis by summarizing the study and listing the future work.

Chapter 2

Background

In this chapter, we first review the achievements and limitations of assumption-based linguistic research with brain imaging techniques in Section 2.1. Then in Section 2.2 we discuss the machine learning methods important for the analysis. Finally, in Section 2.3 we introduce the encoding and decoding methods that can efficiently find patterns related to language phenomena in brain imaging data.

2.1 Cognitive neuroscience of language

2.1.1 Brain imaging techniques

The cognitive neuroscience of language focuses on the brain's neural activity of language processing. In order to explore how the brain accomplishes such a complex task of language comprehension, researchers use many brain imaging techniques to collect various types of signals.

The most commonly used brain recording technologies are electroencephalography (EEG), magnetoencephalography (MEG), and functional magnetic resonance imaging (fMRI). They are all non-invasive and collect data from different aspects that reflect brain activity. Here, we are most interested in EEG because we want to explore the timescales in the human brain.

EEG and MEG are more suitable for time-related analysis because they have a high temporal resolution of millisecond precision. The sensors are placed on the scalp

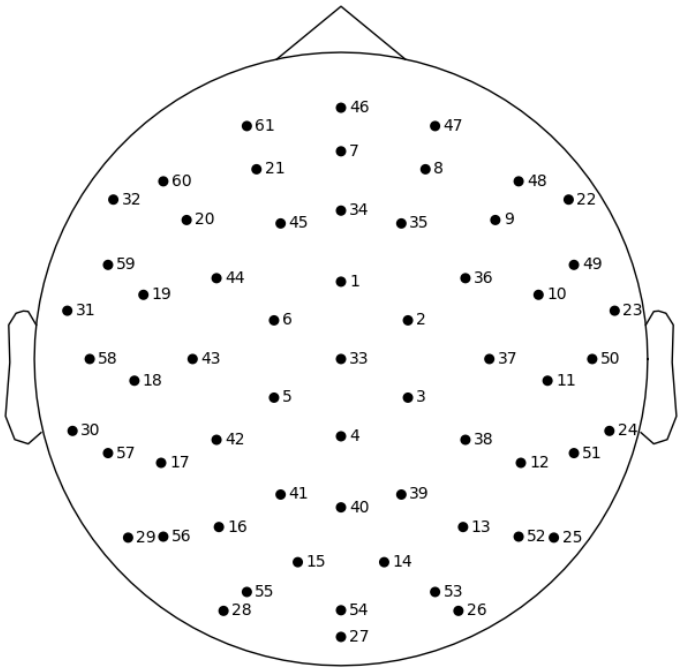


Figure 2.1: The sensor position of the Easycap-M10 EEG system.

to measure the small electric and magnetic fields induced by mass neuron activities. They can be used to construct a processing timeline of certain language functions at the cost of low spatial resolution - only a limited number of sensors can be arranged on the scalp.

The fMRI, on the other hand, uses the blood-oxygen-level-dependent (BOLD) signal to reflect 3-D voxel-wise brain activation. Because the BOLD signals need seconds to respond, the sampling interval of an fMRI is as long as 1 or 2 seconds. However, the millimeter-wise spatial resolution enables us to accurately locate the brain areas responsible for particular language functions.

In order to provide different aspects of neural activity, multiple brain imaging techniques can be applied to the same research topic. For example, if we collect both EEG and fMRI data on participants performing the same task, we can acquire both temporal and spatial information of the brain's activity.

2.1.2 Research paradigm

Because brain imaging data have complex patterns and a low signal-to-noise ratio, we need specific research paradigms to reliably extract useful information from the data.

In the history of cognitive neuroscience in language, most studies followed an assumption-driven approach. Firstly, they define a language function of interest and make an assumption about the processing model for this function. Then they choose conditions relevant to the assumption and create elaborate stimuli that only differ in these conditions. Finally, they analyze the differences between conditions in brain imaging data to verify their assumption. For example, Kutas and Hillyard [7] defined semantic incongruity as a sentence with a surprising ending: "I take coffee with cream and DOG." They assumed that the brain has a response to this phenomenon and designed sentence pairs that only differ in a congruent or an incongruent ending word. They compared the average EEG signals in these two conditions and found the N400, a negative EEG signal appeared 400 milliseconds after the onset of the ending word, correlated with semantic incongruity.

The assumption-driven approach explained many language phenomena. However, they usually focused on small aspects because complicated assumptions increased the difficulty of creating stimuli. The stimuli were created only for particular assumptions, so it was difficult to combine these individual results to answer how the brain regions for different levels work orderly as a whole.

Despite this, some studies did try to compare the brain activities when processing normal text and scrambled text at different levels to explore the hierarchical processing structure of the brain. Lerner et al. [8] recorded subjects' fMRI data when they were hearing stories disorganized at the sound, word, sentence, and paragraph levels. They defined the temporal receptive window (TRWs) in language to describe the time range of stimuli that may influence the brain response. After layered compar-

isons between these conditions, they found a hierarchical structure in the brain: Early auditory areas have short TRWs and adjacent areas along the superior temporal gyrus have intermediate TRWs. The long TRWs mainly belong to the temporal-parietal junction and frontal cortex. Brennan et al. [9] created similar scrambled text but used MEG to focus more on the temporal distribution of brain activity. They chose eight regions of interest and found that most of them have a different activation pattern between sentences and scrambled words at 250-300 milliseconds after the word onset.

These studies focused on whole-brain analysis, but they still followed the assumption-driven approach and compared between specially created stimuli. One issue is that when subjects are exposed to these elaborate stimuli, the brain may not work in the same way as when it processes natural text. The new approach to solve this issue is to use typical text as stimuli and try to find some explicable patterns from brain imaging data without comparisons. This is the start of the data-driven approach, in which subjects usually passively read or hear some language materials just like what they experience in daily life. If one language property is of interest, researchers will analyze the brain imaging data collected when text related to this property is played.

The examples that follow the approach are the two papers by Ding et al. [10, 11]. In their design, their stimuli were sentences containing 4 monosyllabic words. The first 2 words form a noun phrase and the second 2 words form a verb phrase. All the words have the same length (1/4 second), so there are 3 constant frequencies 1Hz, 2Hz, and 4Hz that correspond with sentence, phrase, and word. They ensured the existence of signals with the special frequencies in MEG and EEG data and found the brain areas that produce these frequencies. With this method, the advantage is that we do not need to use "fake language" as control and focus only on brain activity with the natural language.

The studies still have limitations: monosyllabic words and fixed frequencies are strict restrictions in natural language. Besides, since brain imaging data have a large

amount of data, we have to make sure that the frequency patterns we find are exactly related to the language stimuli. We also would like to benefit from the abundance of data and use patterns other than frequency. The methods for efficiently and reliably finding essential patterns from complex data are developed with machine learning techniques discussed in the next section.

2.2 Machine learning techniques

2.2.1 Framework of machine learning

Machine learning is a technique for improving a computer's performance on a task by learning from experiences (data) related to the task. In a typical framework of machine learning, the task is to use a function $h(x)$ that predicts a target y with a sample x. The experiences are two sample sets $X = \{x_1, x_2...x_n\}$ and $Y = \{y_1, y_2, ...y_n\}$ that contain the instances of x and y. With a loss function $L(h, X, Y)$ that evaluates the performance of $h(x)$, an optimization algorithm is applied to use pairs of $[x_i, y_i]$ to improve $h(x)$.

Machine learning has advantages suited for studies in the cognitive neuroscience of psycholinguistics, which involve various types of brain imaging and text data. We can use multiple machine learning algorithms to efficiently explore linear or non-linear relationships between these different types of data (e.g. relationship between grammar and EEG data). If we find the relationship and make a prediction model $h(x)$, we can explain why the model performs well by analyzing the parameters of the optimized $h(x)$. The parameters provide insight into the meaningful patterns corresponding to language characteristics or brain activities in the complex data.

2.2.2 Regularized regression

Regularized regression is an algorithm that explores the linear relationship between two variables. With regularization, it produces a model with better stability and generalizability. We introduce Ridge regression (L_2 regularized regression) [12] used

in our study.

To define the linear relationship, if we have two vectors x and y with dimensions a and b, we want to find a $a * b$ matrix W so that $xW = y$.

In linear regression, if we have N samples of x-y pairs, the sample sets X and Y are matrices with dimensions $N * a$ and $N * b$. The loss function is defined as the total squared error between real and predicted targets:

$$L = \|Y - XW\|^2$$

It has been proven [13] that L is minimized if:

$$W = (X^T X)^{-1} X^T Y$$

Linear regression is efficient, but it also has a problem: if the data in X have high multicollinearity, $X^T X$ is nearly singular and produces a biased inverse. Hoerl and Kennard [12] invented Ridge regression to solve this problem by adding a regularization term with a positive parameter λ, making the new loss function:

$$L = \|Y - XW\|^2 + \lambda \|W\|^2$$

The W that minimizes L can be computed with:

$$W = (X^T X + \lambda I)^{-1} X^T Y$$

in which the I is the identity matrix.

Each of the a dimensions in X is a factor that is possibly related to Y. In Ridge regression, in order to reduce $\lambda \|W\|^2$, the elements in W corresponding to unimportant factors in X are close to 0. This prevents overfitting and produces a more generalizable model than linear regression, especially when the sample size N is smaller than the data dimension a [14].

In our analysis, we use a time window of EEG signals with a dimension of 2950 to predict word vectors with only 720 pairs of samples, i.e. $N = 720$ and $a = 2950$ from above. This makes Ridge regression important for producing a better model.

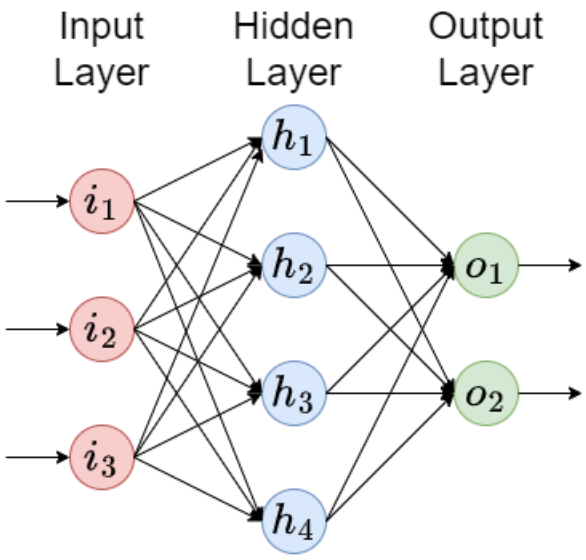


Figure 2.2: An example ANN that has 3 layers. There are 3, 4, and 2 neurons in the input layer, the hidden layer, and the output layer. Each neuron has its own weight matrix, bias vector, and non-linear activation function. Each neuron independently does the computation with its received input.

2.2.3 Artificial neural network

The artificial neural network (ANN) is an important component of machine learning. Different from regularized regression which constructs a linear transformation, an ANN is powerful enough to fit non-linear functions between X and Y . Because of this, large-scale ANNs have achieved remarkable performance in the fields of computer vision, natural language processing, and game strategy [15].

An ANN has a layered structure and each layer contains computing units called neurons. These ideas are derived from the biological neural network in the brain [16]. A typical ANN, as shown in Figure 2.2, has an input layer and an output layer. There can be a variable number of hidden layers between the two layers. Each layer performs computation with the input from the previous layer and outputs to the next layer. Namely, the i-th neuron in the j-th layer accepts the input p_{j-1} from the $(j-1)$-th layer and output p_j^i with the function:

$$p_j^i = \sigma(W_j^i p_{j-1} + b_j^i)$$

in which W_j^i and b_j^i are the neuron's weight matrix and bias vector. σ is called a non-linear activation function that can be a Sigmoid or tanh function. This is why an ANN can be used to explore non-linear relationships. Despite the complex structure, many algorithms, including back-propogation [17], have been implemented by accessible software packages to efficiently train the ANN.

The development of ANN has promoted research on language with a core concept different from psycholinguistics. In psycholinguistics, research on language is based on principles discovered in the field of linguistics. The researchers need to define a property of language (e.g. semantic incongruity) and check how the brain handles this property (e.g. the event-related potential N400). With the ANN models, however, there are few explicit assumptions or definitions about language. Instead, words are converted to vectors, and ANNs are trained by back-propagation to perform relevant tasks, like sentiment analysis, summarization, and question answering. If the ANN models master these tasks, we can infer that the models learn the essential characteristic of language. Moreover, based on the models, we also acquire these secondary discoveries:

- The vectors, converted by ANN from words, represent the characteristic of words [18, 19].
- The outputs of hidden layers (hidden states, or artifacts) in ANNs reflect different levels of integration of language [20].

In the next sections, we discuss specific ANN language models that embody these discoveries.

2.2.4 Word vector models

In order to explore the language comprehension mechanism with computer programs, every word needs to be converted to a vector. Intuitive methods, like using one-hot vectors or encoding by letters, lack an essential element for language comprehension:

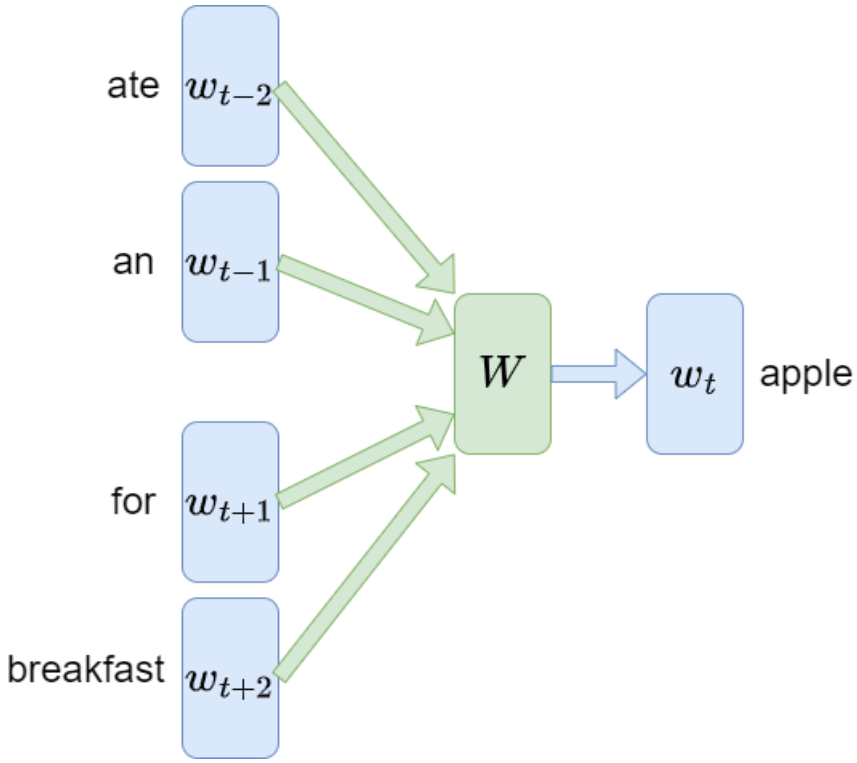


Figure 2.3: The diagram of the CBOW algorithm in Word2Vec. We use the 4 surrounding words to predict the center word w_t. The W in the middle is a 2-layer neural network that converts words (one-hot vectors) to real number vectors.

semantics. A better method considers that words in contexts have related meanings and words with similar meanings appear in similar contexts. Researchers construct a word co-occurrence matrix and decompose it to represent the semantic relationships between words. The Non-negative Sparse Embedding method by Murphy et al. [21] is a typical example of this method.

A milestone in efficiently producing word vectors was achieved by Google with the Word2Vec [18, 19], a set of algorithms that trains two-layer ANNs on a large corpus. Both the skip-gram and continuous bag-of-words (CBOW) algorithms select a window of several words. The CBOW predicts the central word with its surroundings while skip-gram predicts in the opposite direction. From Figure 2.3, the n-dimensional word vectors produced by ANN W are proven to have a good ability for semantic reasoning [18, 19].

One limitation of these word vectors is that they do not consider the context. This is because the window of Word2vec is too small to capture long-distance text dependency. This may cause some issues:

- Multiple meanings of a polyseme will be combined in one vector, but the brain analyzes the context and may activate one explanation.
- If we want to convert a phrase or a sentence to a semantic vector, a vector produced by connecting or averaging word vectors cannot represent the integrated semantics.

2.2.5 Long short-term memory model

A new generation of ANN, the recurrent neural network (RNN), solves these issues of word vector models by accepting much longer texts while extracting and memorizing essential information. Different from ANN neurons that process one-shot input and do not memorize, the neurons in RNN are designed for accepting a series of inputs while saving some information. This makes it possible to process text series of any length. Namely, the i-th neuron in the j-th layer has a function f_j^i. At time step t, It accepts both the input from the previous layer $p_{j-1,t}$ and its memory from the last time step $h_{j,t-1}^i$. The function f_j^i outputs $p_{j-1,t}^i$ to the next layer and saves the new memory $h_{j,t}^i$. The diagram is shown in Figure 2.4.

In an RNN model, a text for processing will be partitioned into a series of minimum meaningful elements called "tokens" (words like "apple", and subwords like "want-" and "-ed" in "wanted"). During the training process, some tokens are intentionally masked and the models need to predict these unseen tokens. The parameters in the models are adjusted to maximize the probability of the correct word for these masked tokens.

The long short-term memory model (LSTM) [22] is a special RNN effective for natural language processing. It has two kinds of memory, short-term and long-term,

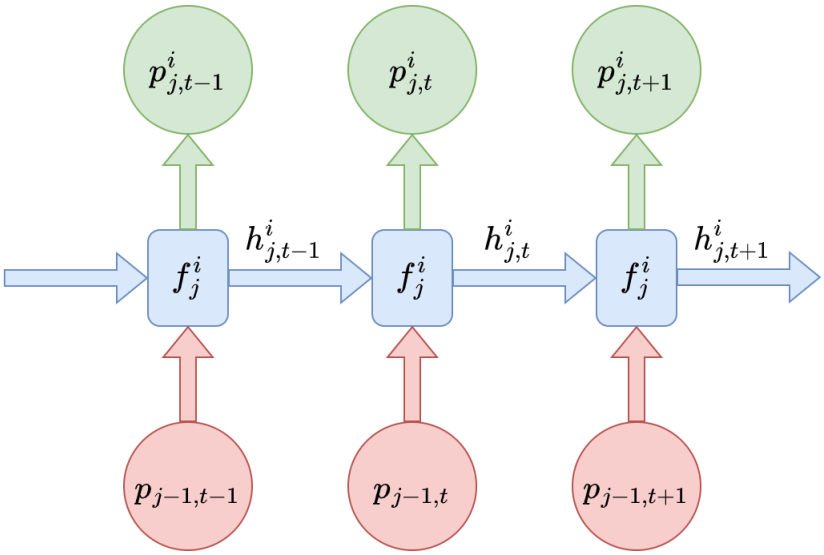


Figure 2.4: The diagram of the i-th RNN neuron on j-th layer. In each time step t the neuron function f^i_j outputs $p^i_{j,t}$ to the next layer based on the input $p^i_{j-1,t}$ from the previous layer and its memory $h^i_{j,t-1}$ from the last time step. It renews its memory to $h^i_{j,t}$ and saves the value for the next time step.

to store both its current output and its inner state. With the stored information, we can trace the relationship between the current word and the words from the preceding text. An LSTM unit uses 3 "gates" (forget gate, input gate, output gate, denoted by f, i, o) to adjust the information flow, and the principle can be shown by Figure 2.5, Table 2.1, and the following formula:

$$f_t = \sigma(W_{fh} \odot h_{t-1} + W_{fx} \odot x_t + b_f)$$

$$i_t = \sigma(W_{ih} \odot h_{t-1} + W_{ix} \odot x_t + b_i)$$

$$o_t = \sigma(W_{oh} \odot h_{t-1} + W_{ox} \odot x_t + b_o)$$

$$\widetilde{c_t} = \tanh(W_{ch} \odot h_{t-1} + W_{cx} \odot x_t + b_c)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t}$$

$$h_t = o_t \odot \tanh(c_t)$$

The $\odot$ means the element-wise multiplication and the σ means the Sigmoid function. For every time step, the unit accepts an input x_t and outputs the hidden state h_t

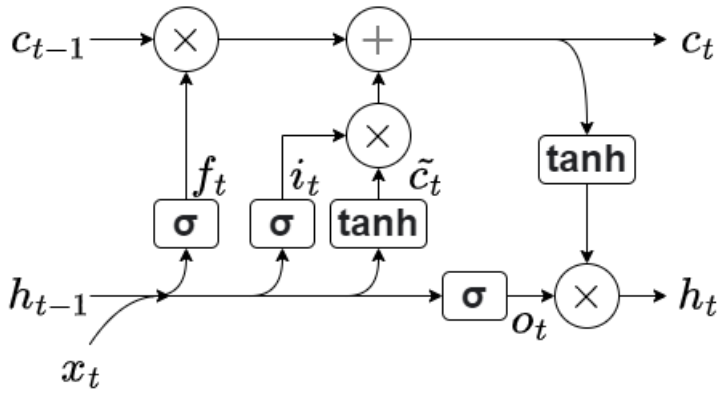


Figure 2.5: The diagram of an LSTM layer.

Table 2.1: The meanings and ranges for LSTM variables.

Variable	Meaning	Range
x_t	Input vector at time step t	$\mathbb{R}^d$
h_{t-1}, h_t	Stored and renewed hidden state vectors	$(-1,1)^h$
c_{t-1}, c_t	Stored and renewed cell state vectors	$(-1,1)^h$
f_t, i_t, o_t	Output vectors of 3 gates	$(0,1)^d$
W_{fh}, W_{ih}, W_{oh}	Weight matrices of 3 gates for hidden state vector	$\mathbb{R}^{h*h}$
W_{fx}, W_{ix}, W_{ox}	Weight matrices of 3 gates for input vector	$\mathbb{R}^{h*d}$
b_f, b_i, b_o	Bias vectors of 3 gates	$\mathbb{R}^h$
$\widetilde{c}_t$	Input activation vector	$(-1,1)^h$
W_{ch}	Weight matrix of $\widetilde{c}_t$ for hidden state vector	$\mathbb{R}^{h*h}$
W_{cx}	Weight matrix of $\widetilde{c}_t$ for input vector	$\mathbb{R}^{h*d}$
b_c	Bias vector of $\widetilde{c}_t$	$\mathbb{R}^h$

(short-term memory), while the cell state c_t represents the unit's long-term memory.

The LSTM model can be modified to accommodate other tasks besides missing token prediction. These improved LSTM models have achieved good performance in many tasks including natural language inference [23] and sentence topic prediction [24]. This indicates that it learns the characteristic of language. The hidden states of LSTM can also integrate word semantics to represent a larger scale of text: Aina

et al. [25] have found that the hidden states in LSTM contain both the lexical and contextual information for words. Zhang et al. [26] have shown that modified LSTM can generate document-level representations.

2.2.6 Multi-timescale LSTM model

Based on the good performance of LSTM in language processing, computing scientists want to further explore its trained parameters. They want to decompose the model and find some units responsible for specified language functions.

One method to explore the interpretability of LSTM is to select a language property and analyze the parameters responsible for it by visualization and ablation. Kementchedjhieva & Lopez [27] found some meaningful hidden states that could be activated by closing punctuations, words, and latinate suffixes. Lakretz et al. [28] found some dimensions that manage long-distance number information and subject-verb dependency. These results show that subunits for various kinds of information can be extracted from LSTM parameters. However, these methods are inefficient and cannot cover all aspects of language.

The other method to study LSTM is to control the LSTM's parameters explicitly. Tallec and Ollivier [29] discovered the timescale property in the LSTM model and found it dependent on the forget gate bias.

From the diagram of LSTM, we know that the cell state c_t is updated by two parts: the old memory c_{t-1} and the new information $\widetilde{c_t}$. If we assume that $f_t + i_t$ is approximately 1, the forget gate output f_t decides the proportion of c_{t-1} that contributes to the new c_t.

Therefore, the forget gate bias b_f plays an important role in memory maintenance [2]. This is because, if we assume $x_t = 0$ after t_0 and no leakage through the hidden state ($W_{ch} = 0$, $b_c = 0$, and $W_{fh} = 0$): $c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t}$ can be simplified to $c_t = f_t \odot c_{t-1}$. This results in:

$$c_t = f_0^{t-t_0} \odot c_0 = e^{(\log f_0)(t-t_0)} \odot c_0$$

If we let $t_0 = 0$ and denote

$$T = -\frac{1}{\log f_0} = \frac{1}{\log(1 + e^{-b_f})}$$

We get $c_t = e^{-\frac{t}{T}} \odot c_0$. The T is the timescale of LSTM decided by the forget gate bias b_f.

This means, following the assumption, if b_f of one unit has a large value, f_t of the corresponding unit will be close to 1 after activation of the Sigmoid function, which means that c_t will retain most of the memory of c_{t-1}. On the other hand, if b_f of one unit has a small value, which results in a close to 0 activation in f_t, c_t will forget most of c_{t-1} and be mainly decided by the current input activation vector $\widetilde{c_t}$. The large b_f corresponds with the long timescale information because the memory tends to be stable. The small b_f corresponds with the short timescale information because the memory decays fast.

The different forget gate biases correspond to multiple timescales, so these LSTMs with modified forget gate biases are called multi-timescale LSTM (MTLSTM). Many studies [30–32] have made changes to LSTM to construct a monotonic forget gate bias series. These models achieved good performance in many tasks, which influenced the models by Mahto et al. [2] that manually set forget gate biases. They proposed the first MTLSTM model that had a specific distribution of forget gate biases. It considered the properties of natural language:

The mutual information between words follows a power law decay t^{-d} [33].

However, the memory in an LSTM unit follows an exponential decay $e^{-\frac{t}{T}}$.

The two decay rates are not equal. To let the LSTM corresponds with the characteristic of language, the timescales T of multiple units should follow a distribution $P(T)$ so that the combined effect of exponential decay becomes a power law decay:

$$t^{-d} \propto \mathbb{E}_T[e^{-\frac{t}{T}}] = \int_0^\infty P(T) e^{-\frac{t}{T}} dT$$

After solving this, the timescale T in the model should follow an inverse gamma distribution:

$$P(T;\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}(\frac{1}{T})^{\alpha+1}e^{-\frac{\beta}{T}}$$

And the forget gate biases can be calculated with T.

This MTLSTM model achieves smaller prediction perplexity than baseline LSTM in both Penn Treebank (PTB) [34] and WikiText-2 [35] datasets. The authors used information routing and found a possible reason: The MTLSTM has more long timescale units. They carry the information of low-frequency words and improve the prediction of these words. In our analysis, we use the hidden states of the MTLSTM model to represent information of different timescales in language comprehension.

2.3 Encoding and decoding methods

2.3.1 Framework of encoding and decoding methods

The encoding and decoding methods are used to explore the relationship between stimuli and brain imaging data. In a psycholinguistic experiment, we first create high-dimensional vectors to represent the stimuli (text). Then we pair the stimuli vector to the brain imaging data produced by a person processing the corresponding stimuli. Finally, we train a machine learning model to learn a mapping between the two elements in the pair. In the encoding method, we use stimuli to predict brain activity while in the decoding method the direction of prediction is the opposite. The assumption and logic behind the methods are simple: the brain creates a mapping of stimuli to its activity; similarly, if the activity in some brain regions can predict the stimuli well during a period of time, it indicates that the stimuli are processed in those brain regions in the period.

The encoding and decoding methods are typical examples of the data-driven approach. With natural next as stimuli, the converted text vectors contain many language characteristics (e.g. word semantics, grammar) that are not restricted by as-

sumptions. Scientists can use both brain imaging data and text vectors to freely explore possible connections between the brain and language characteristics. The following two studies show the process of this approach.

Wehbe et al. [36] collected fMRI data when subjects were reading the ninth chapter of Harry Potter and the Sorcerer's Stone word by word. Semantic representations for words are constructed by decomposing a word co-occurrence matrix using the Non-negative Sparse Embedding method [21]. Together with syntactic (e.g. word length, verb tense) and discourse (e.g. character, emotion) features, they constructed 195 dimension vectors of story segments for every 4 words that accord with the 2-second sampling rate. Based on the encoding results of fMRI data, they mapped each feature to brain areas with high sensitivity to the feature. There are many interesting conclusions based on the whole-brain analysis. First, the brain areas for syntactic and semantic information have a big overlap. Second, the posterior temporal cortex/angular gyrus that perceives emotion also helps understand characters' moods.

In the work of de Heer et al. [37], subjects listened to hours of natural narrative speech. An encoding model was trained to predict fMRI data from 3 feature spaces: spectral, articulatory, and semantics. The spectral and articulatory features were represented by the cochelogram and the phoneme vectors. For the semantics feature of a word, they used a vector that records its co-occurrence frequency with 985 most common words from abundant language materials. The variance explained was used to evaluate the contribution of 3 feature spaces to the prediction. Based on their analysis, they drew atlases of brain areas that corresponded to every feature of language. Important results included: First, both hemispheres were equally activated in speech comprehension. Second, the auditory region processed semantics but with a smaller proportion compared to other language-related regions. These quantitative analyses provided more details than previous experiments that used an assumption-driven approach.

In these two examples, we know that the encoding and decoding methods are effi-

cient for doing whole-brain studies. We also notice that the strategies for constructing word vectors are different between them. We can infer that the quality of these results depends on what extent these vectors can represent semantics [38]. In the next sections, we discuss the language models with ANNs that produce artifacts representing text stimuli.

2.3.2 Encoding and decoding methods with ANNs

The ANN word vector models, including Word2vec [18, 19], GLoVe [39], Fasttext [40], etc., provide us with high-quality word vectors suitable for downstream encoding and decoding tasks, especially for the studies on semantic integration.

Pereira et al. [41] analyzed the correspondence of brain areas in comprehending words and sentences from a large number of semantic categories: Word stimuli are represented by GLoVe vectors and sentence vectors are the average of word vectors in a sentence. The model trained to decode word vectors from fMRI data also performs well in predicting sentence vectors from corresponding fMRI responses. This shows the sentence comprehension process involves analyzing word semantics.

Fyshe et al. [42] recorded the MEG signals when subjects were reading adjective-noun phrases. They trained models to use every time window of 0.1 seconds on the timeline to predict the Word2Vec vectors of both the adjective and the noun. The above-chance prediction for adjectives does not only appear after the onset of adjectives but also reappears and lasts for a period at the onset of the following nouns, indicating a process of integrating word meanings into phrases in the brain.

The encoding and decoding methods also take advantage of RNNs and other ANNs that involve contextual information. These ANNs have a layered structure. The outputs of different hidden layers may represent different levels of abstractions of the text. The encoding and decoding model can be used to compare these hidden representations with the activities in different brain areas. This explores whether the brain and ANNs are using the same algorithm in language processing.

Jat et al. [5] trained decoding models with MEG data and compared both non-contextual and contextual sentence semantics produced by GLoVe [39], ELMo [43], and BERT [44]. They found that the contextual BERT hidden states corresponded the best with brain data. The decoding model not only had a good prediction for the whole sentence but was also sensitive to word differences in similar sentences. This shows the similarity between the brain and language models in semantic processing.

Toneva and Wehbe [6] used the contextual word vectors produced by different contextual language models like ELMo [43] and BERT [44] to predict fMRI recordings and found the hierarchies in these language models all have some similarities with the human brain structures. These results indicate that the shared features in both language models and the brain may contain the core of language comprehension.

2.3.3 Exploring timescales in the brain with MTLSTM

The LSTM model, as introduced in Section 2.2.6, has timescales in its hidden state representations that can help us study how the brain represents these timescales. Before the invention of MTLSTM [2], Jain et al. [45] controlled the timescales by changing the length of the context input into a normal 3-layer LSTM. They recorded subjects' fMRI data when listening to some narrative stories and let the LSTM model process the same text and produce hidden states. Then they used LSTM hidden states with different context lengths to predict fMRI data. With the variation in prediction result, they found that each brain areas have different preferences for context length. Most voxels processed long-context information, while some low-level areas like the auditory cortex (AC), Broca's area, and left temporo-parietal junction (TPJ) focused on short-context word meanings.

The studies have some limitations: The manually controlled context length is correlated with, but not the same as, timescales. This is because context length must be an integer while timescale only needs to be positive. The study also shows that an LSTM fails to integrate contextual information beyond 10-15 words [45]. The

MTLSTM solves these issues. It uses its well-defined timescales to select by itself what to memorize and forget. It also has very long timescale units that exceed the 10-15 word limit.

The work of Jain et al. [3] was the first study that took advantage of separated timescales to interpret the functions of brain regions. The study was an improvement on their previous study [45] by changing the normal LSTM to MTLSTM. After they trained an encoding model that used hidden states of timescales to predict brain responses, they computed the average preferred timescales for each voxel with the contribution of different timescales in the prediction model. They found that the auditory cortex (AC) handles short timescales while the inferior parietal region manages long timescales. They also observed that the timescales in the prefrontal (PFC) cortex and the precuneus (Pr) change smoothly in space. These results corroborated the previous findings and provided more details.

However, there is one limitation: the fMRI has a low temporal resolution. A few words passed in its 2-second sampling interval. When they downsampled the hidden states to fit this frequency, they unavoidably lost information in the word vectors for both short and long timescales.

In our study, we use a decoding method and take advantage of the high temporal resolution of EEG data. Each word corresponds to a unique period of EEG data, so we do not need to downsample the hidden states. Besides, if we align the EEG data with the onsets of words and predict the hidden states using time windows surrounding the onsets, the prediction results will indicate the processing timeline of the brain for different timescales. If the brain and the MTLSTM model have the same definition of timescales, we can make a basic assumption: the short timescales are processed only around the onset and for adjacent words, but the processing for long timescales has a wide span on the timeline.

Chapter 3

Methods

In this chapter, we discuss the method used in the analysis. Sections 3.1 and 3.2 describe how the sample set (EEG data) and target set (contextual word vectors) were built. Section 3.3 discusses the training and evaluation process of the linear regression model (decoding model) that found the connection between these two sets. Sections 3.4 and 3.5 introduce two methods that gave more in-depth explanations: The temporal generalization method (TGM) and single brain area decoding. Section 3.6 discusses the statistical method that tested the reliability of the results.

3.1 EEG data

We used EEG data from Alice [1], a public dataset containing EEG signals collected while subjects (n=49) passively listened to *Alice's Adventure in Wonderland* Chapter 1. In this section we discuss the construction process of the sample set.

3.1.1 Participants

49 native English speakers (14 male, age range 18-29) participated in the study. The dataset excluded the data from 16 subjects because of excessive noise for preprocessing (n=8) and low comprehension questionnaire performance (n=8). After preprocessing as described in 3.1.4 and 3.1.5, we excluded another 14 subjects because of incorrect token numbers (n=4), inadequate good segments (n=6), and abnormal baseline

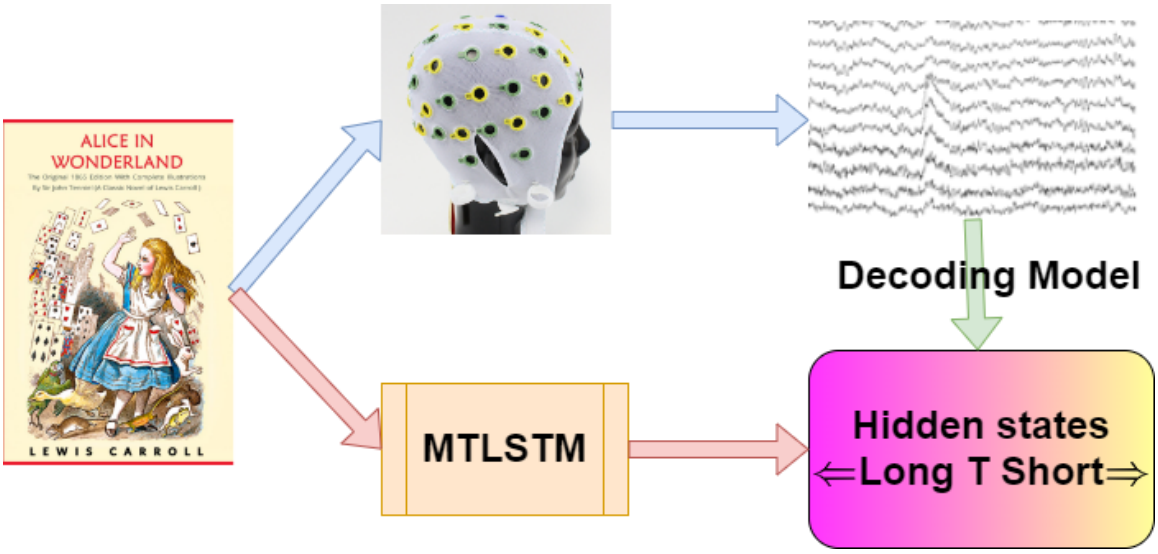


Figure 3.1: The diagram of the analysis. Subjects heard *Alice's Adventures in Wonderland* Chapter 1 while their EEG signal was recorded. The MTLSTM model processed the same text and produced hidden states with different timescales. A decoding model was used to find the correlation between the EEG and hidden states.

(n=4).

Data from 19 participants (3 male, age range 18-25) were included in the final analysis.

3.1.2 Stimulus

The stimulus was Kristen McQuillan's reading version of the first chapter of *Alice's Adventures in Wonderland* by Lewis Carroll on LibriVox. Unlike other chapters that contain obscure metaphoric words or special structures like poems, the first chapter describes a comprehensible narrative story, which is easy to fully understand by a native English speaker. Therefore, the chapter is representative of natural language.

The recording was slowed by 20% with a pitch-preserving PSOLA algorithm, implemented in Praat [46], and normalized to 70dB SPL to sound more natural.

The stimulus contained 2129 tokens (914 lexical) in 84 sentences and lasts for 12.4 minutes.

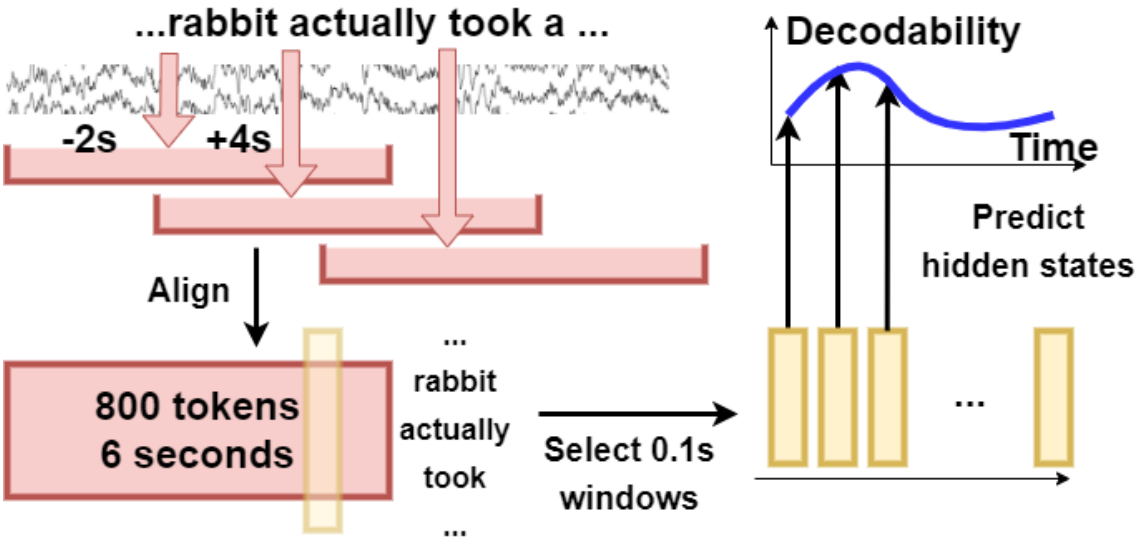


Figure 3.2: The diagram of processing EEG signals. A -2s to 4s time window of EEG around the onset of each word was selected. The time windows were aligned based on the onsets. Time windows of 0.1s were selected to predict the word vectors. The prediction performance (decodability) of one time window corresponds with one point on the timeline.

3.1.3 Procedure

Participants were placed in an isolated booth and passively listened to the stimulus with inserted earphones. EEG signals were collected at 500Hz using a 61-electrode elastic cap (actiCap, Brain Products GmbH) with the Easycap M10 layout. Channel 25 (right mastoid) acted as the reference channel. For each participant, the onset of each word was recorded as a timestamp along with the EEG data.

In order to ensure the subjects were focusing on the material, after listening, participants needed to complete an 8-question multiple choice questionnaire about story events. The participants' attentiveness when listening to the stimulus was evaluated by the accuracy of the questionnaire. The EEG data were excluded from the dataset if a participant failed to answer over 4 questions.

3.1.4 Removing global artifacts

The original EEG signals have low signal-to-noise ratios. Continuously recording for 12 minutes causes instability in the EEG signals. Subjects' movement and physiological signals introduce noise into the data. Therefore, in order to improve the data quality, we took advantage of the arguments provided in the dataset to do basic preprocessing. For each subject, we used the following preprocessing methods to remove noise and artifacts with the toolbox provided by MNE [47].

1. Removed the data of 2 auxiliary channels, VEOG and Aux5.
2. Re-referenced to both mastoids (Channel 25 and 29) and removed the data of these reference channels.
3. Applied a 0.1Hz to 100Hz band filter and a 60Hz notch filter to the rest of the channels.
4. Marked bad channels (with high impedance or apparent noise) recorded in the dataset documents and identified by observation.
5. Used FastICA to separate the rest channels into independent components.
6. Removed identifiable electrooculogram, electrocardiogram, and electromyogram artifacts.
7. Used the Spherical Spline Interpolation method to repair the marked bad channels.

3.1.5 Creating sample set

We chose 800 tokens from 914 lexical words based on the criteria we will mention in Section 3.2.4. For each subject and each token, a -2s to 4s time window around the onset was selected. A time window was treated as a bad segment and rejected if the

maximum amplitude in this period was larger than 80 µV. For one token, the EEG was the average of all good segments of this token.

During this process, we found that the number of good segments for some subjects was smaller than 300 (inadequate good segments, n=6). For some participants (abnormal baseline, n=4), after averaging all the good segments, the average EEG signals deviated from the baseline or showed obvious alpha waves. These phenomena indicated the inattention of the participants or the dissatisfactory quality of the data. Therefore, we also rejected these participants.

After preprocessing, our EEG had a size of 800*59*3000, which corresponded with the number of tokens, channels, and time steps (500Hz * 6s).

3.2 Word vectors

We used two kinds of word vectors in the analysis, pretrained word vectors from Word2vec and contextual word vectors from MTLSTM. We chose the 300-dimension pretrained Word2vec vectors trained on the Googlenews dataset with the Skipgram algorithm [19]. This section discusses the MTLSTM word vectors.

3.2.1 Materials

Wikitext-2 is a collection of high-quality articles from Wikipedia. It has a 2M-token training set, a 217K-token validation set, and a 245K-token test set with a vocabulary of 33,278 unique tokens. We used it as the pretrain set to accelerate the training process and avoid overfitting.

For the fine-tuning materials, three Lewis Carroll Books, *Alice's Adventures in Wonderland*, *Through the Looking-Glass and What Alice Found There*, and *Sylvie and Bruno* from Project Gutenberg were used as materials for training and testing MTLSTM. The data was parsed with the Spacy toolbox [48]. All the punctuations were removed, except for those in meaningful abbreviations like "'s" or "n't".

Because Chapter 1 of *Alice in Wonderland* was the test set (2.1K tokens), consid-

ering the similarity of adjacency, we used Chapters 2-3 as the validation set (3.9K tokens). The rest of the materials were used as the training set (115K tokens) to build the token vocabulary. The test set was modified to contain the same 2129 tokens as the EEG materials. Tokens that existed in the validation and test sets but not in the training set were replaced with the "Unknown" token (<unk>). The vocabulary contained 7006 unique tokens and covered 2052 of the 2129 tokens (867 of the 914 lexical tokens) in the test set. For the sake of batching in training, the "End of sentence" token (<eos>) was added at the end of each paragraph. We call this modified dataset LC (Lewis Carroll) set. An example of tokenized text is in Appendix A.

3.2.2 MTLSTM setting

The settings of MTLSTM followed the work of Jain et al. [3]. It is a stateful LSTM with three layers that accepts a 400-dimension word embedding and outputs a vector of the same dimension. The first two layers have 1150 units, and the third has 400 units. Before the LSTM layers, the model has an embedding layer that encodes each word in the vocabulary into a 400-dimension word embedding. The same parameters in the embedding layer are used to convert the output of LSTM into a probability distribution of possible words in the vocabulary, therefore calculating prediction loss and perplexity.

In the first layer, half of the units are assigned with the forget gate bias T=3 and half with T=4 to ensure they only process short timescale information. In the second layer, the forget gate biases follow an inverse gamma distribution of alpha=0.56 and scale=1. In both layers, the input gate biases are set to the negative value of the forget gate biases in the corresponding units. These biases are fixed and not trained. All other parameters are randomly initialized within a range [-0.1, 0.1] and optimized during training.

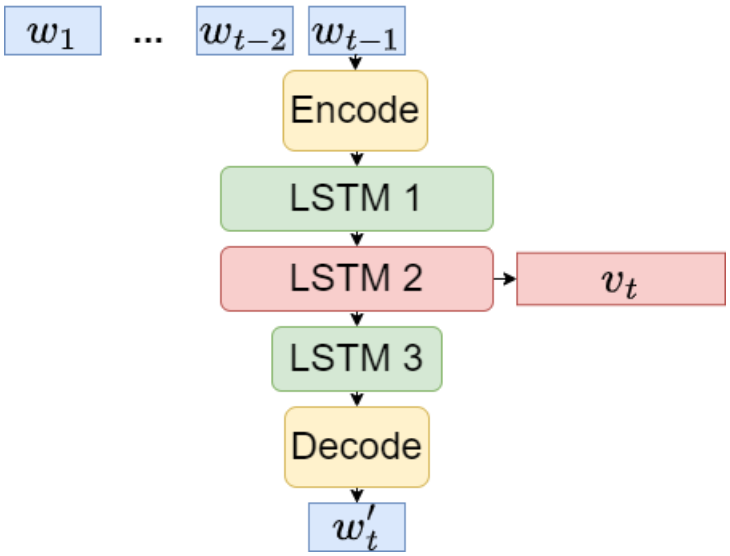


Figure 3.3: The diagram for the MTLSTM model. The model has 3 LSTM layers. The encoder and the decoder share the weights. In order to get the word vectors for word w_t, we sequentially input the word w_1 through w_{t-1} and the model produce a prediction w'_t. After we input w_{t-1}, the output of the second LSTM layer, v_t is the word vecor of word w_t.

3.2.3 MTLSTM training

In order to obtain a more generalizable model and accelerate training, we trained the MTLSTM with the Stochastic Gradient Descent (SGD) [49] algorithm in two stages.

In the pretrain stage, we used the Wikitext-2 dataset to train 10 MTLSTM models with different initialization for 1600 epochs.

In the fine-tuning stage, we first replaced the embedding layer with the vocabulary of the LC set. Then we fine-tuned each model with the LC set for another 30000 epochs. The average perplexity was 83.57±0.82 for the test set. We chose the best model with test perplexity 82.59 to produce word vectors for the downstream task.

3.2.4 Token selection

In order to produce the multi-timescale contextual word vectors for every word, we continuously input the test set text into the selected model. After inputing token w_{t-1}, we recorded the hidden states of the second layer v_t. Every v_t had a dimension

of 1150.

Because we would like to explore how semantic representations are carried in the model to predict the next lexical word, we were more interested in the vectors of lexical words rather than functional words. We chose 800 from the 914 lexical words with the following criteria:

- Not in the beginning of the passage, because the context is short and not compatible with other tokens.
- Not at the end of the passage, because some EEG signals do not have a time window as long as 4 seconds after the onset for analysis.
- Not a name like "Alice" or "Dinah" in the story.

For these 800 tokens, we got an $800 * 1150$ target set which corresponds with the sample set of size $800 * 59 * 3000$.

3.3 Prediction model

In this section, we discuss the prediction model that explores the correlation between EEG and word vectors in Section 3.3.1. We also discuss the training and evaluating paradigm in Section 3.3.2.

3.3.1 Ridge regression model

The EEG data has a high temporal resolution, so we can train decoding models at different time points and produce a timeline for the decoding performance. Instead of using the EEG signals on the whole timeline for training, we continuously selected 0.1s time windows with a shift of 0.01s. Neighboring windows had an overlap of 0.09s. Each time we trained data from a single time window for all the samples (800*59*50) to predict the same word vectors (800*1150). The prediction performance reflects the variation of the brain's correspondence with the representation along the timeline.

The prediction model we used was a Ridge regression model (linear regression model with L2 regularization) to avoid overfitting. We suppose X is the prediction sample and Y is the target. With regularization factor λ, We need to find a weight matrix W to minimize the loss L:

$$L = \|Y - XW\|^2 + \lambda \|W\|^2$$

This results in the weight matrix:

$$W = (X^T X + \lambda I)^{-1} X^T Y$$

in which the I is the identity matrix.

In our analysis, if we have N pairs of EEG and word vectors, the X in the formula is flattened samples of EEG signals ($N*(59*50)$). The Y is one dimension of the word vectors ($N * 1$). Because the brain's correspondence with different timescales varies even in the same time window, for each dimension of the word vectors, a separate model was independently regularized, trained, and evaluated.

3.3.2 Training and evaluating paradigm

We used a paradigm of cross-validation to evaluate the model. The core motivation of the paradigm is that the model should be generalizable to similar, unseen data.

For training and testing, we did 10-fold cross-validation. The data set ($N = 800$) was randomly divided into 10 parts. Each time the model was trained on 9 parts ($N_{train} = 720$) and tested on the other 1 part (($N_{test} = 80$)). For the training part, we calculated the mean and standard deviation of every column (factor) of X. Then we computed the standard scores of every factor to normalize the data. The means and standard deviations acquired in the normalization were also used to adjust the test part.

During training, we used Leave-One-Out cross-validation (LOOCV) with mean squared error to select the best regularization parameter λ. In one round of LOOCV,

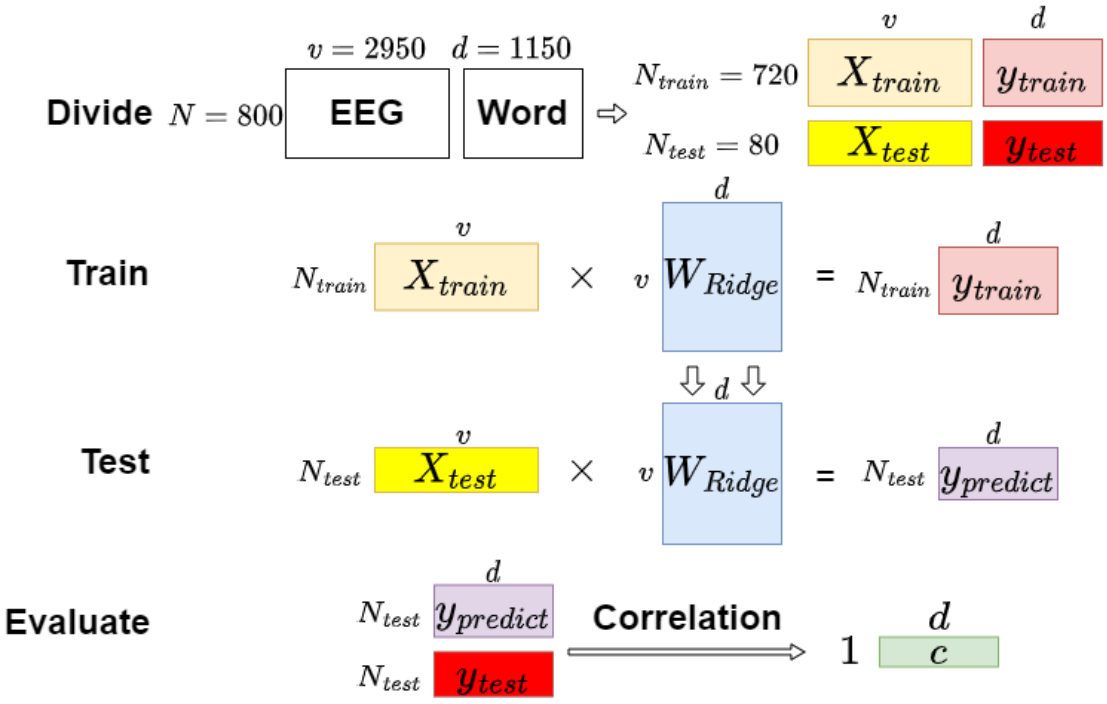


Figure 3.4: The diagram of training, testing, and evaluating a decoding model.

one of the samples is left out. Then a model is trained on all the rest of the samples and validated on the left-out sample to get the error. This process is repeated for every sample in the set. For every λ we can compute a total error, which reflects the generalizability of the model. We use the λ of the best-performing model to train on all the samples in the training set.

Each best-performing model is evaluated on the test set by the correlation between the real word vectors and predicted ones. For each model, we produce a 600*1150 matrix of the correlation value. 600 is the number of time windows on the timeline and 1150 is the number of word vector dimensions. Therefore, for every timescale in the MTLSTM, we can plot a timeline measuring the change in decoding performance as measured by correlation.

To achieve a more stable result, the above process was repeated 10 times with 10 distinct random partitions of data. The final result is the average of these repetitions.

3.4 Temporal generalization method

On the timeline, for each time window of EEG signals, we train a separate ridge regression model to predict the word vectors. However, the performance curve cannot reflect whether the regression models at different time points are similar. If the models on two time windows both have good performance, do the brain's representations of the stimuli change? We use the temporal generalization method (TGM) [50] to answer this question.

Instead of testing with samples from the same time window, the TGM evaluates the model by testing samples from all time windows. For a group of specified timescales, with T time windows on the timeline, the TGM produces a $T * T$ dimension matrix that indicates the generalizability of the models. On one hand, if models on two time windows can be generalized to each other, the brain's representations are stable. On the other hand, if they cannot be generalized, it suggests a difference between the brain's representations.

3.5 Single brain areas decoding

This section describes our framework for determining which brain areas contribute most to the performance of the decoding model. The core motivation is to use the EEG data from one sensor to train the decoding model. The prediction performance reflects the importance of the brain areas detected by the sensor.

Considering the low signal-to-noise ratio of a single sensor, a sensor group is created for each sensor. The sensor group consists of a central sensor and all its adjacent sensors according to the Easycap-M10 montage. The sensor groups provide more stable results for comparing the contribution of different brain areas.

The training process was the same as in Section 3.3. The difference was that each time we only use one sensor group, rather than the whole brain EEG data to train the model. For each period of interest with specified timescales, we averaged the test

correlation for each sensor and plot the results on the scalp figure with the MNE toolbox [47].

3.6 The permutation test

The purpose of training and testing the decoding model is to explore whether there is a connection between the brain and LSTM models in language processing. To show statistical significance, we need to reject the null hypothesis that there is no connection, implemented by the permutation test. In the permutation test, the data in the target set are permuted so that each EEG sample is assigned another word vector instead of the corresponding one. This simulates no connection between EEG data and word vectors.

For each of the analyses, we ran permutation tests with different random seeds and then followed the same procedure as the original test. We used the kernel density estimation with a Gaussian kernel to simulate a null hypothesis and calculated the p-value. For multiple comparisons performed on the timeline, we used the Benjamini-Hochberg-Yekutieli False Discovery Rate (FDR) [51] with no dependency assumption to correct the p-value.

Chapter 4

Results

In this chapter, we used the methodology mentioned in Chapter 3 to do multiple analyses on decoding models. In Section 4.1 we describe an initial experiment that ensures the decodability of semantic information from EEG signals. In Sections 4.2-4.5 we analyze the decodability for MTLSTM word vectors and evaluate different timescales separately. The next two sections introduce experiments that provide more temporal and spatial details: Section 4.6 uses the TGM method to compare the models between different time points. Section 4.7 explores the brain areas that contribute more to decoding. Section 4.8 concludes the chapter.

4.1 Decoding EEG to pretrained Word2vec vectors

The objective of this analysis is to examine whether semantics can be decoded from EEG signals. The reason for this initial experiment is: The EEG signals were collected continuously during continuous speech listening, inducing complex activity in the brain. We have to ensure the EEG signals contain semantic information before we compare them with representations of different timescales.

To implement this test, we train our models using 0.1-second EEG time windows to predict each dimension of the pretrained Word2Vec embedding vectors. Adjacent windows on the timeline have a shift of 0.01s. If the decoding models show signifi-

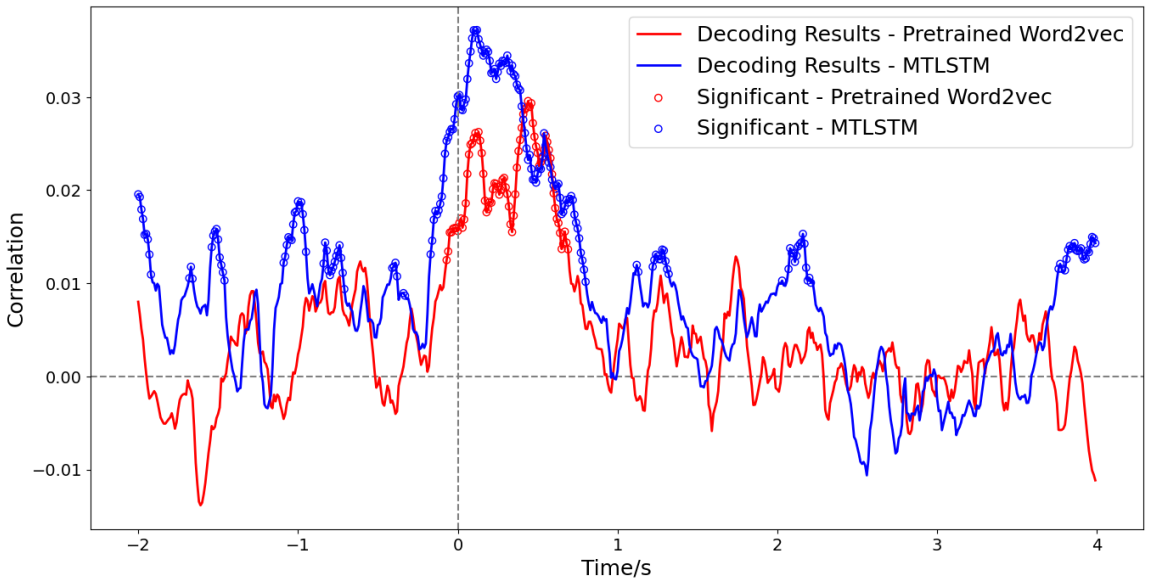


Figure 4.1: Average correlation between real word vectors and predicted ones for pretrained Word2Vec and MTLSTM vectors. Each data point on the line represents the decoding performance of a 50ms time window (the point marks the end of the time window). The circles show significantly better than chance predictions ($p < 0.05$, FDR corrected).

cantly better than chance prediction performance around word onset, we can conclude that the preprocessed EEG signals carry semantic representations.

From the red line in Figure 4.1, we found that reliable predictions appear at 0.07s before the onset and last until 0.69s after the onset. The curve has multiple peaks at 0.2s and 0.5s. This period is wider than the 0s-0.4s period in the study of Wehbe et al. [52] that used word embedding vectors to predict MEG signals.

The result shows that context-free semantic information can be decoded from EEG signals. Considering the average duration of a lexical word (about 0.3s), the period with a length of 0.76s indicates the complicated activity of the brain: because there exists context, the brain can predict the word semantics before the onset and keep processing after the word's appearance.

4.2 Decoding EEG to MTLSTM word vectors

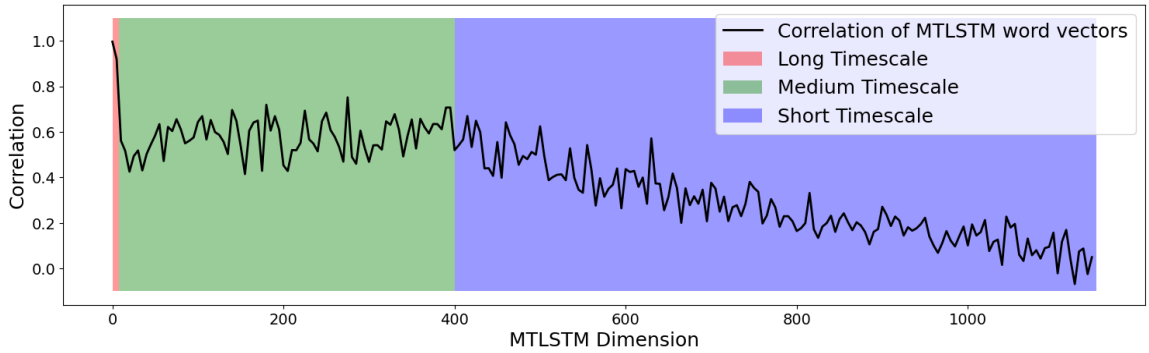
Once we had confirmed the decodability of context-free semantic information, we need to explore whether EEG can also predict MTLSTM hidden states.

According to Figure 3.3, for each word, the MTLSTM hidden state v_t is the intermediate output of predicting the next word w_t. Therefore v_t contains a semantic representation of the word w_t. We would expect that decoding with MTLSTM vectors will show similar patterns of performance. From another aspect, the MTLSTM model has processed word w_1 through w_{t-1}, so it has stored the essence of context, reflected in v_t. This context information is processed around w_t by the person, so we would also expect good decoding performance around the word onset. The experiment design is the same as in Section 4.1 except that we change the 300-dimension Word2vec vectors to the 1150-dimension MTLSTM hidden states.

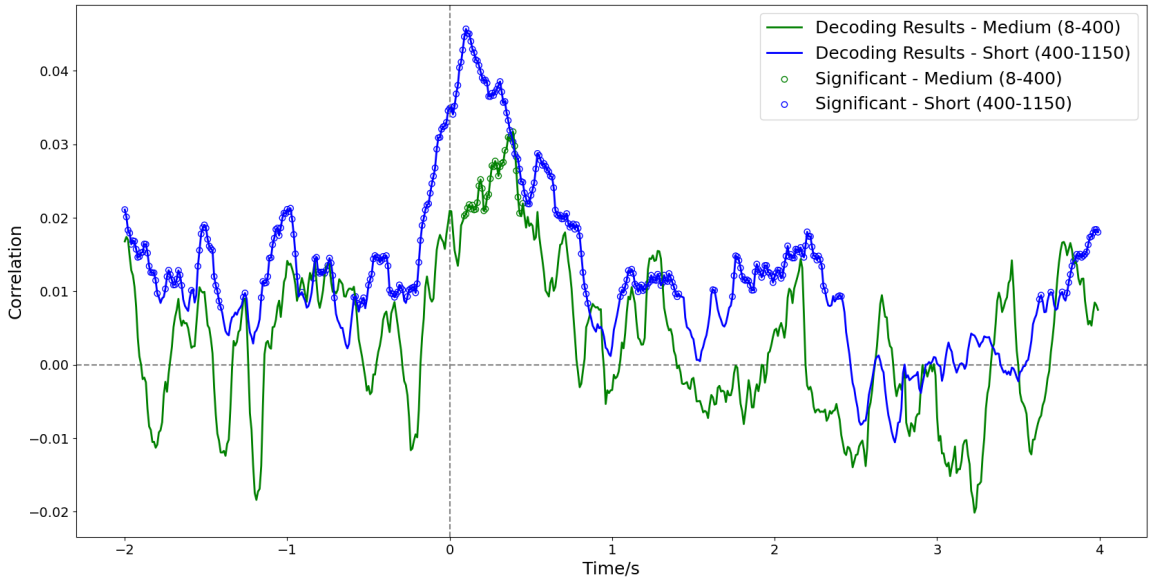
The result is shown with the blue line in Figure 4.1. As expected, the peak decoding performance appears around the word onset. The range is from 0.17s before the onset to 0.70s after the onset, which is wider than the results of pretrained vectors. This indicates that the brain creates contextual semantic representations that are similar to the MTLSTM model. We also found better than chance decoding performance at time points distant from word onsets (-2s, -1s, 1.2s, 2s, 4s). This suggests the more abundant contextual information in the MTLSTM hidden states that correlates with EEG signals in the brain during continuous text comprehension.

4.3 Selecting subgroups for different timescales

In the previous 2 sections, the prediction performance is derived from the accumulative effect of all vector dimensions. We cannot distinguish which dimension contributes to decodability. In this next analysis, we separately explore the prediction performance for different timescales. If the prediction is better than chance at one time point for a particular timescale, it may indicate that the information of this



(a) Self-correlation of MTLSTM hidden states and partition of MTLSTM dimensions into short, medium, and long timescales.



(b) Decoding results for MTLSTM word vectors based on the partition. Each data point on the line represents the decoding performance of a 50ms time window (the point marks the end of the time window). The circles show significantly better than chance predictions ($p < 0.05$, FDR corrected).

Figure 4.2: The primary partition (short, medium, and long) for MTLSTM hidden states dimensions based on correlation.

timescale is similar to what is being processed in the brain.

The primary task for this analysis is to determine the timescales of interest. The timescales are determined by the forget gate biases, the unique properties of LSTM units. We cannot independently analyze each dimension because one LSTM unit carries limited information and a lot of noise, resulting in a large variance in decoding

results. Therefore, we need to select a subset of units and average the results. Because the 1150 LSTM unit dimensions have ordered forget gate biases from large to small, units with adjacent indexes have similar timescales, and therefore may encode similar language properties. Based on this, we can create contiguous groups of dimensions and the average prediction performance of the dimensions shows the properties of the group.

To partition all 1150 dimensions, we plot the self-correlation for each dimension of the hidden state. Before a lexical token w_t is input into the MTLSTM, the hidden state is v_t. After input, the hidden state becomes v_{t+1}. We calculate the correlation for each dimension between v_t and v_{t+1}. A plot of correlation values appears in Figures 4.2a and 4.3a, which reflects the stability of each dimension. Because the correlations have a large variance, we group and average every 5 dimensions.

Using the original self-correlation data and the figure, we can partition these dimensions into 3 groups: The first group (1-8) has a very high (>0.8) correlation. The second group (8-400) has a relatively stable correlation that is around 0.6. The third group (400-1150) has a correlation that is steadily decreasing from around 0.6 to 0. We regard them as the long, medium, and short timescale groups as shown in Figure 4.2a.

Moreover, for the short timescale group, because of the enormous size (750) and linear decreasing of correlation, we further separate it into 3 equal subsets as shown in Figure 4.3a. In this partition, the short-long (S-Long) timescales are dimensions 400 to 650. the short-medium (S-Medium) 650 to 900 and the short-short (S-Short) 900 to 1150.

4.4 Timescales in decoding MTLSTM vectors

Based on the grouping of MTLSTM dimensions defined in Section 4.3, we can investigate which timescales support good decodability.

Figure 4.2b displays the average prediction performance with the unbalanced first

partition shown in Figure 4.2a. The long timescale group (Figure B.1) has obvious oscillations. The maximum correlation is larger than 0.05 and the minimum is -0.15. However, no results are significant anywhere on the timeline. For the medium and short timescales (Figure 4.2b), there is no rapid change along the timeline. The medium timescale is only significant in a small period after the onset of the word with a correlation peak of 0.03. The short timescale has the highest peak (0.05) and widest range (1 second) of significant correlation around the onset of word.

The failure in decoding long timescale information does not imply there is no stable information in the brain. One possible reason is the correlation for the word vector dimensions are close to 1. This means the values output by these MTLSTM units are very stable. A short paragraph will cause meaningful changes in neither the values of these dimensions nor the brain's representation of stable information. Therefore, the variances for stable information in EEG and long timescale dimensions are not correlated, and the regression model may fail to study their relationship.

The medium and short timescale groups are also worth discussion because the ranges of reliable decoding disagree with the correlation analysis in Section 4.3: For a series of words, if the MTLSTM's representations for some timescales are similar (with high correlation), we assume that the brain's representations for these timescales may also be similar. Therefore, for the medium timescale, the brain's representation of one word may better predict the MTLSTM's representation of neighboring words, compared with the short timescale. This is the opposite of the decoding result: At time points distant from the center word onset, the short timescale information for neighboring words is less similar to the center word than the medium timescale, but the brain activity while processing neighboring words predicts the center word vector more accurately than medium timescales.

One explanation can be derived from the work of Mahto et al. [2], which mentions that the longer timescale units of the MTLSTM model improve the MTLSTM's prediction of infrequent words. Following this assumption, the information on the

infrequent words can be stored in longer timescale units with more stable memory. The memory is partly reflected by the hidden states of these units. However, the brain may use another strategy: it stores the infrequent word information in the brain's memory and only activates it near the onset of its relevant words. Due to the infrequency, there may be seconds of intervals between these relevant words, larger than the time range of our analysis. During the intervals, the EEG signals may not reflect the memory of the relevant words. The long interval and mechanism difference may cause the failure to decode at distant time points for the medium timescale.

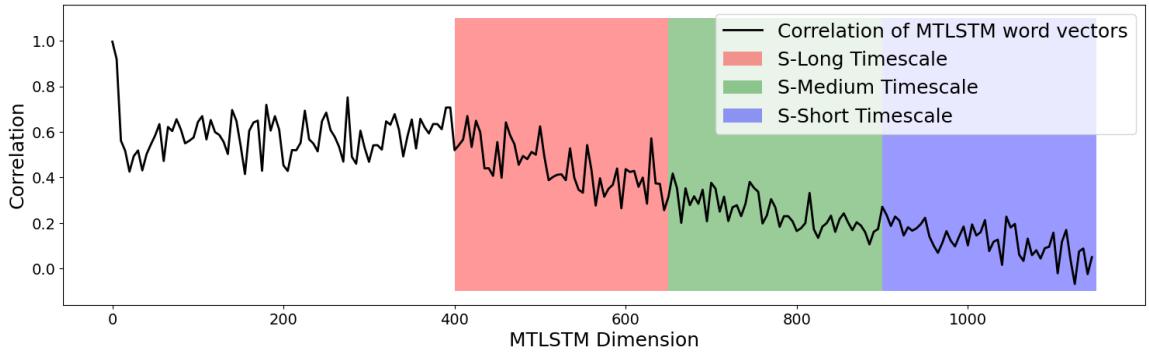
From this aspect, in an MTLSTM model, a longer timescale unit may hold stabler memory, but it may not need continuous processing in the brain reflected in the EEG signals. On the other hand, a shorter timescale unit produces less stable representations, but these representations are repeatedly processed by the brain near the onset.

4.5 Timescales in decoding MTLSTM vectors with a further partition

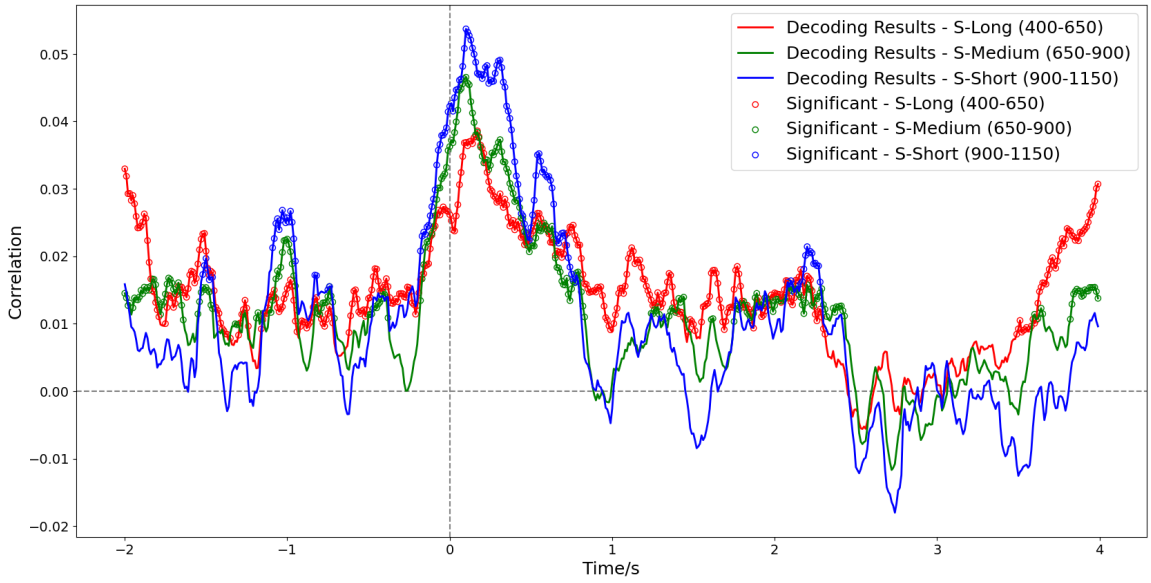
Because the curve of short timescales in Figure 4.2b is very similar to the overall decoding result in Figure 4.1, we still need to trace the source of the above chance prediction distant from the onset. To deeply explore the short timescale group, we divide the 750 dimensions into 3 subsets of the same size as mentioned in Section 4.3.

The three lines in Figure 4.3a show similar patterns. They all have a peak around the onset. However, the ranges for the above chance predictions are different. The S-Short timescales show significant results only around the onset, while the S-Long timescales are widely spread from -2s to 2s. This indicates that S-Short timescale information is mainly activated locally, while S-Long timescale information is continuously processed around the onset. From this aspect, the brain shows similarities with MTLSTM in language comprehension.

Although the long timescale group produces the most reliable predictions at distant



(a) Self-correlation of MTLSTM hidden states and further partition of short timescales into S-Long, S-Medium, and S-Short timescales.



(b) Decoding results for MTLSTM word vectors based on the partition of short timescales. Each data point on the line represents the decoding performance of a 50ms time window (the point marks the end of the time window). The circles show significantly better than chance predictions ($p < 0.05$, FDR corrected).

Figure 4.3: The further partition (S-Long, S-Medium, and S-Short) of short timescales for MTLSTM hidden states dimensions based on correlation.

time points, other timescale groups also produce them. For the shorter timescale, better than chance predictions also appear discontinuously at -1.5s, -1s, and 2s. The medium timescale is similar to this pattern, but with lower peaks, and wider time ranges. The longer timescale groups also show better than chance prediction around 4s after the word onset.

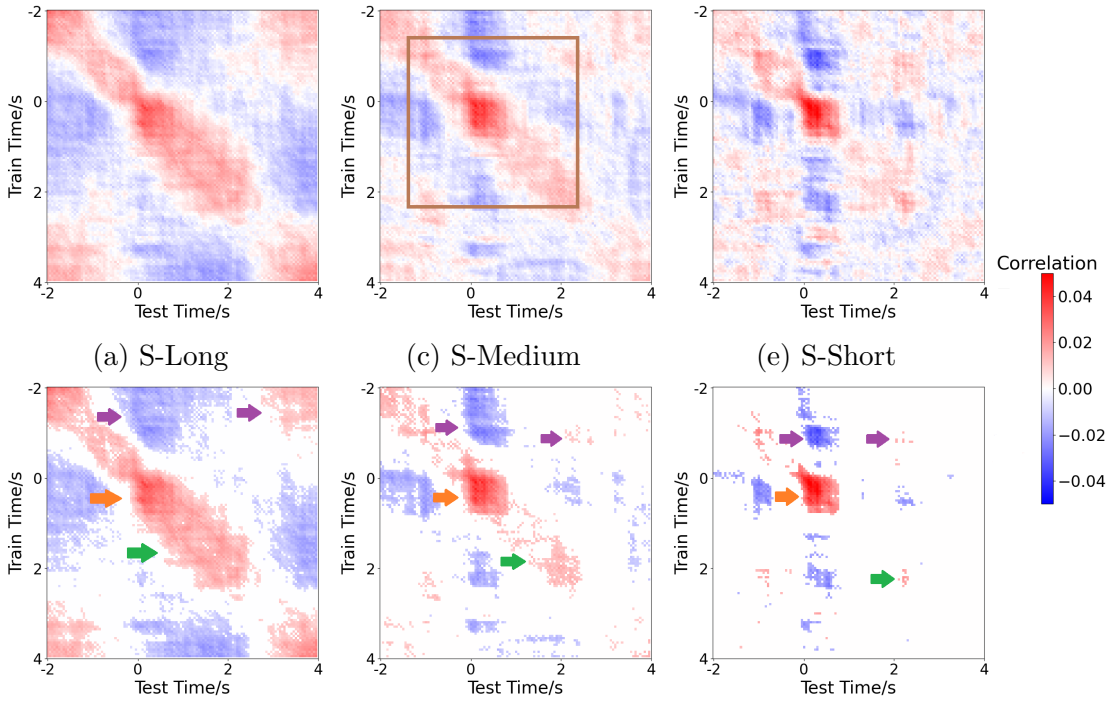
These results provide evidence that the brain handles long and short timescales differently from an MTLSTM model. One possible speculation is: some neural circuits produce oscillations with different frequencies in the brain to deal with the information flow of different timescales. For example, there may be an oscillation of 1Hz to process the short timescale information. It activates every second and explains the good decoding performance at -1s, 0s, 1s, and 2s. The frequency of 1Hz may correspond with phrases. The long timescale may correspond with an oscillation with a period of seconds to process sentences. The studies of Ding et al [10] that discover the oscillations of different frequencies in the brain for processing words, phrases, and sentences may support this speculation.

Another explanation is the complex behavior of the brain. Because related words appear both before and after the center words, the brain is repeatedly processing similar semantic information: It uses the previous words to predict the next few words before their onset and recalls the words during a period to integrate semantics after the onset.

Though these explanations may be sound, we need further experiments with different settings in future work to confirm them. One issue is: because of the varying word lengths, we only align the EEG signals with the onset of the center words rather than adjacent words. The significant peaks are the cumulative effect of multiple related nearby words with random onsets on the timeline.

4.6 Time generalization of decoding models

From the timescale analysis in Sections 4.3-4.5, we know that the prediction performance of S-Long, S-Medium, and S-Short timescales have different patterns on the timeline, which indicates that the corresponding information is processed in the brain with different durations and frequencies. In this section, we use the TGM method (Section 3.4) to investigate the generalizability of a model over time. In a TGM analysis, a model trained on one time point is also tested and evaluated at other time



(a) S-Long (c) S-Medium (e) S-Short

(b) S-Long, FDR (d) S-Medium, FDR (f) S-Short, FDR

Figure 4.4: TGM figures for S-Long, S-Medium, and S-Short timescales. The top 3 subfigures show all correlation values and the bottom 3 subfigures show only better than chance correlation values ($p < 0.05$, FDR corrected).

points. To be specific, if a model trained with data from one time window also has good decodability when tested on data from another time window, it indicates that the brain is representing similar information. This helps us explore the similarities of each significant period in Figure 4.3b and test whether they are repeated or unique processes of the brain.

We used the same partition of timescales described in Section 4.5 and changed the steps between time windows from 0.01s to 0.05s to decrease the amount of computation required. The results and the FDR corrected results for S-Long, S-Medium, and S-Short groups are shown in Figure 4.4.

Figures 4.4b, 4.4d, 4.4f show a similar square-like pattern in the middle (orange arrows). This means that the model trained around word onset generalizes well to other time windows in the range of 1s surrounding word onset. This implies some subset of brain activity is similar for these timescales in the range.

Despite the similarities, the three figures show obvious differences in the range of generalization windows: In the S-Long timescale (Figure 4.4b), the small square is a part of a band-like pattern (green arrow) that lasts from before 0s to after 2s. This pattern, symmetric along the diagonal, shows that each model on the timeline in this 0s-2s range can be generalized to a 1s time window around it. This indicates both the persistence of S-Long timescale representations in the brain. For the S-Short timescale (Figure 4.4f), the pattern is a separate cluster (patterns pointed by orange arrow and green arrow are not connected), showing that the brain processes this information only around the onset. The S-Medium timescale (Figure 4.4d) has a transitional form: there exists not only a square resembling the short timescale but also an obscure band-like pattern (green arrow) with a shorter width than the long timescale.

The other interesting property is the reliable prediction at distant time points. For the S-Long timescale (Figure 4.4b), The models around -2s and 4s share some similarities, but they all show a negative correlation with the middle band (purple arrows). The S-Short timescale (Figure 4.4f), with another 2 above chance predictions at -1s and 2s, follows the same pattern (purple arrows). However, at these time points, they are only momentarily appearances rather than continuous ones in long timescales. The S-Medium timescale (Figure 4.4d) is a transitional form between them: On one hand, it is similar to the short timescales but enlarges the generalization window for distant time points (purple arrows). On the other hand, the time range -1s to 2s in S-Medium timescales (brown box in Figure 4.4c) has a similar structure to -2s to 4s in S-Long timescales (Figure 4.4a).

These results provide us with further evidence that the brain treats long and short timescale information differently. The timescale length is reflected both in the processing duration and intervals between processing. The result indicates that the brain's representations around word onset are opposite to those distant from the onset. It also indicates that for the distant time windows, the brain's representations before

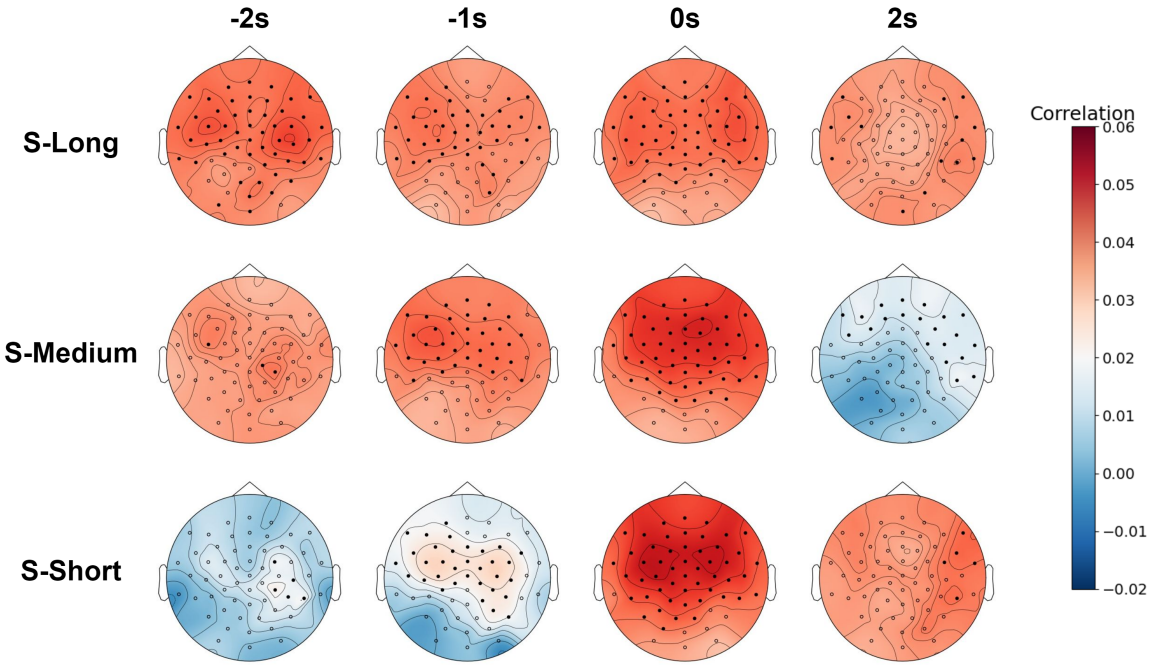


Figure 4.5: The topographic map at a few key time points for S-Long, S-Medium, and S-Short timescales. Solid dots show significantly better than chance predictions ($p < 0.05$, FDR corrected).

and after word onset are similar. This can be because of the oscillation as explained in Section 4.5.

4.7 Brain areas contributing to decoding models

In this section, we analyze the decoding model from a spatial perspective: we tend to see which brain areas are responsible for language comprehension. In this analysis, we train models with a sensor and its adjacent neighbors. The accuracy of the prediction at a few key time points with reliable prediction is shown with the topographic map in Figure 4.5.

Due to the low spatial resolution and noise in EEG, it is difficult to tell from the figure the specific brain areas for language processing. We can only compare the figures by row or by column to produce some preliminary conclusion about the differences between long and short timescales.

At time points closer to the onset (-1s and 0s), for all 3 timescales, we can decode

word vectors from nearly every sensor. The highest accuracy appears at the frontal lobe from both hemispheres. The figure does not show an obvious preference for one hemisphere.

At distant time points (-2s and 2s), the S-Long timescales can still be decoded from nearly the whole brain, but the S-Medium and S-Short timescales concentrate on limited brain areas. We can see that the area for the S-Short timescale is mainly the right temporal lobe, which contains the auditory cortex. The S-Medium timescale also has activation in the prefrontal cortex. This result corresponds with the study of Jain et al. [3] that the auditory cortex prefers short timescales while the prefrontal cortex processes long timescales.

These results show that the brain areas that handle long and short timescale information have differences that correspond with previous findings, but we need to improve the method in order to draw more accurate results.

4.8 Summary of analyses

In this Chapter, we discussed the multiple analyses that decode EEG signals to the information of different timescales from MTLSTM hidden states. We first confirmed the decodability around word onset of both Word2vec and MTLSTM vectors from EEG signals. Then we partitioned the MTLSTM timescale by self-correlation and found the short timescale group was decoded with the best performance. Next, we did a further partition to the short timescale group and found S-Long timescales can be decoded for most of the time windows while S-Short timescales can be decoded at some distant time points from word onset. Next, we used the TGM analysis and found S-Long timescales have a wider generalization window than S-Short timescales. Finally, we predicted MTLSTM vectors with sensor groups and found brain areas specific for S-Medium and S-Short timescales at critical time points. From these results, we conclude that the brain's representations of timescales in language comprehension are correlated with timescales in MTLSTM, but the brain may have a complicated

strategy in processing these timescales.

Chapter 5

Conclusion

In the final chapter, we conclude our experiment and analysis and list the core findings. We also raise some considerations for future work to improve our experiments and possibly produce additional insights.

5.1 Summary of contents

In this thesis, we first introduced the research topic: Understanding the timescales of language processing in the brain. Then we summarize the previous studies with different approaches, including cognitive neuroscience, artificial intelligence, and the encoding and decoding methods that build connections between them. After that, we discussed our methods to train decoding models that use EEG signals to predict the MTLSTM word vectors in continuous text comprehension. Details include the construction of sample and target set, and the paradigm of training and evaluating. Finally, we perform multiple analyses to explore the similarities and differences in the processing timeline of different timescales for the brain.

We conclude the main findings of the analyses:

- EEG signals for continuous text comprehension can predict the semantics from both pretrained Word2vec vectors and contextual hidden states from MTLSTM models (Section 4.1, 4.2).

- The brain momentarily processes short timescale information and continuously

processes long timescale information surrounding the onset (Section 4.3, 4.5). Some longer timescale information is only processed in a small period after the onset (Section 4.4).

- The TGM analyses indicate brain's representations for long timescales are more persistent than short timescales (Section 4.6).

- EEG signals from nearly the whole all sensors can predict the semantics surrounding the onset. For distant time points, the right temporal lobe prefers shorter timescales while the prefrontal cortex prefers longer timescales (Section 4.7).

5.2 Future work

In the experiment, the timescales are defined only based on the property of an LSTM unit. We do not know the exact information each timescale carries. We also mention that the onset of adjacent words is not controlled, so reliable predictions can be an accumulative effect of words with different lengths. If we classify the relationship between the central words and adjacent words by their part of speech and align the onset of adjacent words, we can investigate the timescales' preference for different information and how the brain processes these timescales.

The oscillation of information in the brain, if it exists, is also of interest. The speech we process every day has different speeds, complexity, and topics. One question is: Do our findings also apply to language materials with other topics or varying speeds? Answering the question requires us to collect various types of language materials to perform comparative experiments.

5.3 Summary of thesis

To summarize, in this thesis we extend the study of Jain et al. [3] and understand better the relationship between MTLSTM and brain activity in language compre-

hension. The representations of different timescales in MTLSTM and the brain have both similarities and differences. These preliminary results help us to design future experiments to further explore how multiple levels of computations are performed in the brain for language comprehension.

Bibliography

[1] S. Bhattasali, J. Brennan, W.-M. Luh, B. Franzluebbers, and J. Hale, "The alice datasets: Fmri & eeg observations of natural language comprehension," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 120–125.

[2] S. Mahto, V. A. Vo, J. S. Turek, and A. G. Huth, "Multi-timescale representation learning in lstm language models," *arXiv preprint arXiv:2009.12727*, 2020.

[3] S. Jain, V. Vo, S. Mahto, A. LeBel, J. S. Turek, and A. Huth, "Interpretable multi-timescale models for predicting fmri responses to continuous natural speech," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 738–13 749, 2020.

[4] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," *arXiv preprint arXiv:1806.08730*, 2018.

[5] S. Jat, H. Tang, P. Talukdar, and T. Mitchell, "Relating simple sentence representations in deep neural networks and the brain," *arXiv preprint arXiv:1906.11861*, 2019.

[6] M. Toneva and L. Wehbe, "Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain)," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[7] M. Kutas and S. A. Hillyard, "Reading senseless sentences: Brain potentials reflect semantic incongruity," *Science*, vol. 207, no. 4427, pp. 203–205, 1980.

[8] Y. Lerner, C. J. Honey, L. J. Silbert, and U. Hasson, "Topographic mapping of a hierarchy of temporal receptive windows using a narrated story," *Journal of Neuroscience*, vol. 31, no. 8, pp. 2906–2915, 2011.

[9] J. Brennan and L. Pylkkänen, "The time-course and spatial distribution of brain activity associated with sentence processing," *Neuroimage*, vol. 60, no. 2, pp. 1139–1148, 2012.

[10] N. Ding, L. Melloni, H. Zhang, X. Tian, and D. Poeppel, "Cortical tracking of hierarchical linguistic structures in connected speech," *Nature neuroscience*, vol. 19, no. 1, pp. 158–164, 2016.

[11] N. Ding, L. Melloni, A. Yang, Y. Wang, W. Zhang, and D. Poeppel, "Characterizing neural entrainment to hierarchical linguistic units using electroencephalography (eeg)," *Frontiers in human neuroscience*, vol. 11, p. 481, 2017.

[12] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[13] A. Ben-Israel and T. N. Greville, *Generalized inverses: theory and applications*. Springer Science & Business Media, 2003, vol. 15.

[14] A. Tsigler and P. L. Bartlett, "Benign overfitting in ridge regression," *arXiv preprint arXiv:2009.14286*, 2020.

[15] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, e00938, 2018.

[16] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[17] Y. LeCun, D Touresky, G Hinton, and T Sejnowski, "A theoretical framework for back-propagation," in *Proceedings of the 1988 connectionist models summer school*, vol. 1, 1988, pp. 21–28.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[20] C. Guan, X. Wang, Q. Zhang, R. Chen, D. He, and X. Xie, "Towards a deep and unified understanding of deep neural models in nlp," in *International conference on machine learning*, PMLR, 2019, pp. 2454–2463.

[21] B. Murphy, P. Talukdar, and T. Mitchell, "Learning effective and interpretable semantic models using non-negative sparse embedding," in *Proceedings of COLING 2012*, 2012, pp. 1933–1950.

[22] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[23] S. Wang and J. Jiang, "Learning natural language inference with lstm," *arXiv preprint arXiv:1512.08849*, 2015.

[24] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck, "Contextual lstm (clstm) models for large scale nlp tasks," *arXiv preprint arXiv:1602.06291*, 2016.

[25] L. Aina, K. Gulordava, and G. Boleda, "Putting words in context: Lstm language models and lexical ambiguity," *arXiv preprint arXiv:1906.05149*, 2019.

[26] W. Zhang, Y. Li, and S. Wang, "Learning document representation via topic-enhanced lstm model," *Knowledge-Based Systems*, vol. 174, pp. 194–204, 2019.

[27] Y. Kementchedjhieva and A. Lopez, "Indicatements that character language models learn english morpho-syntactic units and regularities," *arXiv preprint arXiv:1809.00066*, 2018.

[28] Y. Lakretz, G. Kruszewski, T. Desbordes, D. Hupkes, S. Dehaene, and M. Baroni, "The emergence of number and syntax units in lstm language models," *arXiv preprint arXiv:1903.07435*, 2019.

[29] C. Tallec and Y. Ollivier, "Can recurrent neural networks warp time?" *arXiv preprint arXiv:1804.11188*, 2018.

[30] P. Liu, X. Qiu, X. Chen, S. Wu, and X.-J. Huang, "Multi-timescale long short-term memory neural network for modelling sentences and documents," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2326–2335.

[31] J. Xu, D. Chen, X. Qiu, and X. Huang, "Cached long short-term memory neural networks for document-level sentiment classification," *arXiv preprint arXiv:1610.04989*, 2016.

[32] Y. Shen, S. Tan, A. Sordoni, and A. Courville, "Ordered neurons: Integrating tree structures into recurrent neural networks," *arXiv preprint arXiv:1810.09536*, 2018.

[33] H. W. Lin and M. Tegmark, "Critical behavior from deep dynamics: A hidden dimension in natural language," *arXiv preprint arXiv:1606.06737*, 2016.

[34] M. A. Marcinkiewicz, "Building a large annotated corpus of english: The penn treebank," *Using Large Corpora*, vol. 273, 1994.

[35] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[36] 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*, vol. 9, no. 11, e112575, 2014.

[37] W. A. de Heer, A. G. Huth, T. L. Griffiths, J. L. Gallant, and F. E. Theunissen, "The hierarchical cortical organization of human speech processing," *Journal of Neuroscience*, vol. 37, no. 27, pp. 6539–6557, 2017.

[38] S. Abnar, R. Ahmed, M. Mijnheer, and W. Zuidema, "Experiential, distributional and dependency-based word embeddings have complementary roles in decoding brain activity," *arXiv preprint arXiv:1711.09285*, 2017.

[39] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[40] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.

[41] F. Pereira *et al.*, "Toward a universal decoder of linguistic meaning from brain activation," *Nature communications*, vol. 9, no. 1, pp. 1–13, 2018.

[42] A. Fyshe, G. Sudre, L. Wehbe, N. Rafidi, and T. M. Mitchell, "The lexical semantics of adjective–noun phrases in the human brain," *Human brain mapping*, vol. 40, no. 15, pp. 4457–4469, 2019.

[43] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. [Online]. Available: https://aclanthology.org/N18-1202.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[45] S. Jain and A. Huth, "Incorporating context into language encoding models for fmri," *Advances in neural information processing systems*, vol. 31, 2018.

[46] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot. Int.*, vol. 5, no. 9, pp. 341–345, 2001.

[47] A. Gramfort *et al.*, "Meg and eeg data analysis with mne-python," *Frontiers in neuroscience*, p. 267, 2013.

[48] M Honnibal and I Montani, "Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *Unpublished software application. https://spacy. io*, 2017.

[49] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[50] A. Fyshe, "Studying language in context using the temporal generalization method," *Philosophical Transactions of the Royal Society B*, vol. 375, no. 1791, p. 20 180 531, 2020.

[51] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal statistical society: series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[52] L. Wehbe, A. Vaswani, K. Knight, and T. Mitchell, "Aligning context-based statistical models of language with brain activity during reading," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 233–243.

Appendix A: Example of tokenized material

This appendix show the original and tokenized version of one paragraph.

A.1 Original text

Alice was not a bit hurt, and she jumped up onto her feet in a moment: she looked up, but it was all dark overhead; before her was another long passage, and the White Rabbit was still in sight, hurrying down it. There was not a moment to be lost: away went Alice like the wind, and was just in time to hear it say, as it turned a corner, “Oh my ears and whiskers, how late it’s getting!” She was close behind it when she turned the corner, but the Rabbit was no longer to be seen: she found herself in a long, low hall, which was lit up by a row of lamps hanging from the roof.

A.2 Tokenized text

alice was not a bit hurt and she jumped up <unk>her feet in a moment she looked up but it was all dark <unk>before her was another long passage and the white rabbit was still in sight hurrying down it there was not a moment to be lost away went alice like the wind and was just in time to hear it say as it turned a corner oh my ears and whiskers how late it ’s getting she was close behind it when she turned the corner but the rabbit was no longer to be seen she found herself in a long low hall which was lit up by a row of lamps hanging from the roof <eos>

Appendix B: Supplemental figures

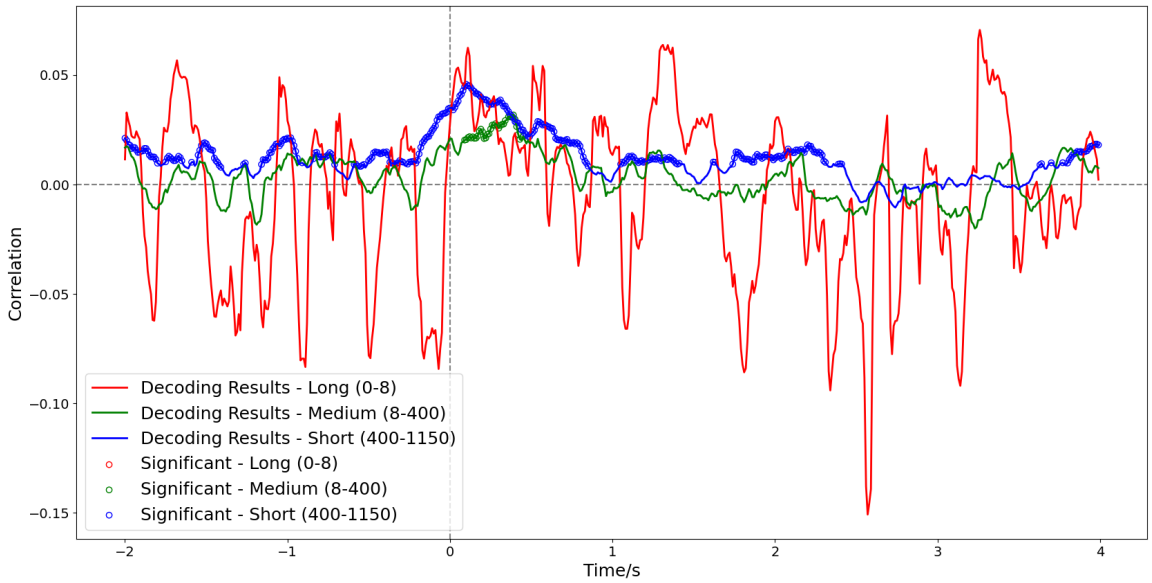


Figure B.1: Decoding results for MTLSTM word vectors based on the partition: Long (0-8), Medium (8-400), and Short (400-1150). Each data point on the line represents the decoding performance of a 50ms time window (the point marks the end of the time window). The circles show significantly better than chance predictions ($p < 0.05$, FDR corrected).