Aligning context-based statistical models of language with brain activity during reading

Leila Wehbe[1,2], **Ashish Vaswani**[3], **Kevin Knight**[3] **and Tom Mitchell**[1,2]
[1] Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA
[2] Center for the Neural Basis of Computation, Carnegie Mellon University, Pittsburgh, PA
[3] Information Sciences Institute, University of Southern California, Los Angeles, CA
lwehbe@cs.cmu.edu, vaswani@usc.edu, knight@isi.edu, tom.mitchell@cs.cmu.edu

Abstract

Many statistical models for natural language processing exist, including context-based neural networks that (1) model the previously seen context as a latent feature vector, (2) integrate successive words into the context using some learned representation (embedding), and (3) compute output probabilities for incoming words given the context. On the other hand, brain imaging studies have suggested that during reading, the brain (a) continuously builds a context from the successive words and every time it encounters a word it (b) fetches its properties from memory and (c) integrates it with the previous context with a degree of effort that is inversely proportional to how probable the word is. This hints to a parallelism between the neural networks and the brain in modeling context (1 and a), representing the incoming words (2 and b) and integrating it (3 and c). We explore this parallelism to better understand the brain processes and the neural networks representations. We study the alignment between the latent vectors used by neural networks and brain activity observed via Magnetoencephalography (MEG) when subjects read a story. For that purpose we apply the neural network to the same text the subjects are reading, and explore the ability of these three vector representations to predict the observed word-by-word brain activity.

Our novel results show that: before a new word i is read, brain activity is well predicted by the neural network latent representation of context and the predictability decreases as the brain integrates the word and changes its own representation of context. Secondly, the neural network embedding of word i can predict the MEG activity when word i is presented to the subject, revealing that it is correlated with the brain's own representation of word i. Moreover, we obtain that the activity is predicted in different regions of the brain with varying delay. The delay is consistent with the placement of each region on the processing pathway that starts in the visual cortex and moves to higher level regions. Finally, we show that the output probability computed by the neural networks agrees with the brain's own assessment of the probability of word i, as it can be used to predict the brain activity after the word i's properties have been fetched from memory and the brain is in the process of integrating it into the context.

1 Introduction

Natural language processing has recently seen a surge in increasingly complex models that achieve impressive goals. Models like deep neural networks and vector space models have become popular to solve diverse tasks like sentiment analysis and machine translation. Because of the complexity of these models, it is not always clear how to assess and compare their performances as they might be useful for one task and not the other. It is also not easy to interpret their very high-dimensional and mostly unsupervised representations. The brain is another computational system that processes language. Since we can record brain activity using neuroimaging, we propose a new direction that promises to improve our understanding of both how the brain is processing language and of what the neural networks are modeling by aligning the brain data with the neural networks representations.

In this paper we study the representations of two kinds of neural networks that are built to predict the incoming word: recurrent and finite context models. The first model is the Recurrent Neural Network Language Model (Mikolov et al., 2011) which uses the entire history of words to model context. The second is the Neural Probabilistic Language Model (NPLM) which uses limited context constrained to the recent words (3 grams or 5 grams). We trained these models on a large Harry Potter fan fiction corpus and we then used them to predict the words of chapter 9 of *Harry Potter and the Sorcerer's Stone* (Rowling, 2012). In parallel, we ran an MEG experiment in which 3 subject read the words of chapter 9 one by one while their brain activity was recorded. We then looked for the alignment between the word-by-word vectors produced by the neural networks and the word-by-word neural activity recorded by MEG.

Our neural networks have 3 key constituents: a hidden layer that summarizes the history of the previous words ; an embeddings vector that summarizes the (constant) properties of a given word and finally the output probability of a word given

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 233–243,
October 25-29, 2014, Doha, Qatar.

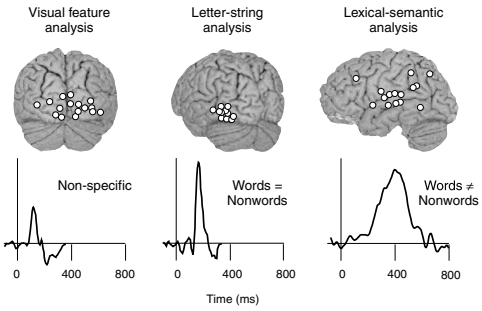

Figure 1: Cortical dynamics of silent reading. This figure is adapted from (Salmelin, 2007). Dots represent projected sources of activity in the visual cortex (left brain sketch) and the temporal cortex (right brain sketch). The curves display the mean time course of activation in the depicted source areas for different conditions. The initial visual feature analysis in the visual cortex at $\sim$100 ms is non-specific to language. Comparing responses to letter strings and other visual stimuli reveals that letter string analysis occurs around 150 ms. Finally comparing the responses to words and nonwords (made-up words) reveals lexical-semantic analysis in the temporal cortex at $\sim$200-500ms.

the context. We set out to find the brain analogs of these model constituents using an MEG decoding task. We compare the different models and their representations in terms of how well they can be used to decode the word being read from MEG data. We obtain correspondences between the models and the brain data that are consistent with a model of language processing in which brain activity encodes story context, and where each new word generates additional brain activity, flowing generally from visual processing areas to more high level areas, culminating in an updated story context, and reflecting an overall magnitude of neural effort influenced by the probability of that new word given the previous context.

1.1 Neural processes involved in reading

Humans read with an average speed of 3 words per second. Reading requires us to perceive incoming words and gradually integrate them into a representation of the meaning. As words are read, it takes 100ms for the visual input to reach the visual cortex. 50ms later, the visual input is processed as letter strings in a specialized region of the left visual cortex (Salmelin, 2007). Between 200-500ms, the word's semantic properties are processed (see Fig. 1). Less is understood about the cortical dynamics of word integration, as multiple theories exist (Friederici, 2002; Hagoort, 2003).

Magnetoencephalography (MEG) is a brain-imaging tool that is well suited for studying language. MEG records the change in the magnetic field on the surface of the head that is caused by a large set of aligned neurons that are changing their firing patterns in synchrony in response to a stimulus. Because of the nature of the signal, MEG recordings are directly related to neural activity and have no latency. They are sampled at a high frequency (typically 1kHz) that is ideal for tracking the fast dynamics of language processing.

In this work, we are interested in the mechanism of human text understanding as the meaning of incoming words is fetched from memory and integrated with the context. Interestingly, this is analogous to neural network models of language that are used to predict the incoming word. The mental representation of the previous context is analogous to the latent layer of the neural network which summarizes the relevant context before seeing the word. The representation of the meaning of a word is analogous to the embedding that the neural network learns in training and then uses. Finally, one common hypotheses is that the brain integrates the word with inversely proportional effort to how predictable the word is (Frank et al., 2013). There is a well studied response known as the N400 that is an increase of the activity in the temporal cortex that has been recently shown to be graded by the amount of surprisal of the incoming word given the context (Frank et al., 2013). This is analogous to the output probability of the incoming word from the neural network.

Fig. 2 shows a hypothetical activity in an MEG sensor as a subject reads a story in our experiment, in which words are presented one at a time for 500ms each. We conjecture that the activity in time window a, i.e. before word i is understood, is mostly related to the previous context before seeing word i. We also conjecture that the activity in time window b is related to understanding word i and integrating it into the context, leading to a new representation of context in window c.

Using three types of features from neural networks (hidden layer context representation, output probabilities and word embeddings) from three different models of language (one recurrent model and two finite context models), we therefore set to predict the activity in the brain in different time windows. We want to align the brain data with the various model constituents to understand where and when different types of processes are computed in the brain, and simultaneously, we want to

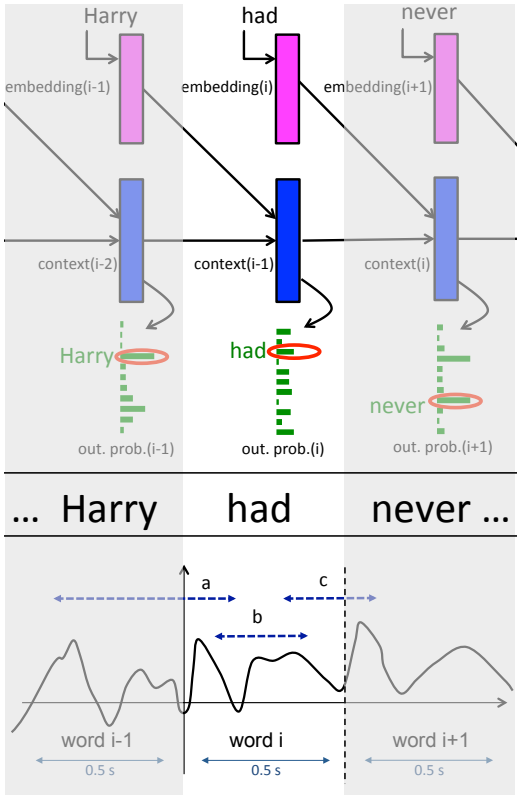

Figure 2: [Top] Sketch of the updates of a neural network reading chapter 9 after it has been trained. Every word corresponds to a fixed embedding vector (magenta). A context vector (blue) is computed before the word is seen given the previous words. Given the context vector, the probability of every word can be computed (symbolized by the histogram in green). We only use the output probability of the actual word (red circle). [Bottom] Hypothetical activity in an MEG sensor when the subject reads the corresponding words. The time periods approximated as a, b and c can be tested for information content relating to: the context of the story before seeing word i (modeled by the context vector at i), the representation of the properties of word i (the embedding of word i) and the integration of word i into the context (the output probability of word i). The periods drawn here are only a conjecture on the timings of such cognitive events.

use the brain data to shed light on what the neural network vectors are representing.

Related work

Decoding cognitive states from brain data is a recent field that has been growing in popularity. Most decoding studies that study language use functional Magnetic Resonance Imaging (fMRI), while some studies use MEG. MEG's high temporal resolution makes it invaluable for looking at the dynamics of language understanding. (Sudre et al., 2012) decode from MEG the word a subject is reading. The authors estimate from the MEG data the semantic features of the word and use these as an intermediate step to decode what the word is. This is in principle similar to the classification approach we follow, as we will also use the feature vectors as an intermediate step for word classification. However the experimental paradigm in (Sudre et al., 2012) is to present to the subjects single isolated words and to find how the brain represents their semantic features; whereas we have a much more complex and "naturalistic" experiment in which the subjects read a non-artificial passage of text, and we look at processes that exceed individual word processing: the construction of the meanings of the successive words and the prediction/integration of incoming words.

In (Frank et al., 2013), the amount of surprisal that a word has given its context is used to predict the intensity of the N400 response described previously. This is the closest study we could find to our approach. This study was concerned with analyzing the brain processes related only to surprisal while we propose a more integral account of the processes in the brain. The study also didn't address the major contribution we propose here, which is to shed light on the inner constituents of language models using brain imaging.

1.2 Recurrent and finite context neural networks

Similar to standard language models, neural language models also learn probability distributions over words given their previous context. However, unlike standard language models, words are represented as real-valued vectors in a high dimensional space. These word vectors, referred to as *word embeddings*, can be different for input and output words, and are learned from training data. Thus, although at training and test time, the input and output to the neural language models are *one-hot* representation of words, it is their embeddings that are used to compute word probability distributions. After training the embedding vectors are fixed and it is these vectors that we will use later on to predict MEG data. To predict MEG data, we will also use the latent vector representations of context that these neural networks produce, as well as the probability of the current word given the context. In this section, we will describe how recurrent neural network language models and feedforward neural probabilistic language models compute word probabilities. In the interest of space, we keep this description brief, and for details, the reader is requested to refer to the original papers describing these models.

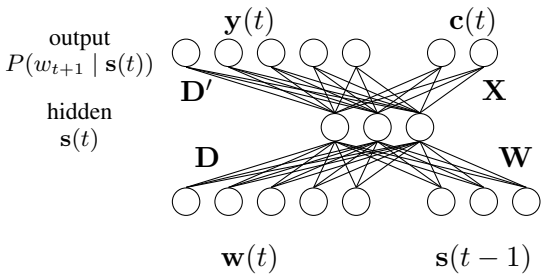

Figure 3: Recurrent neural network language model.

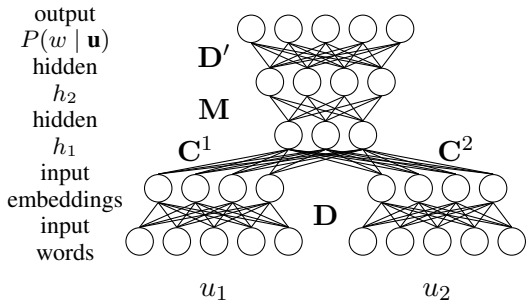

Figure 4: Neural probabilistic language model

Recurrent Neural Network Language Model

Unlike standard feedforward neural language models that only look at a fixed number of past words, recurrent neural network language models use all the previous history from position 1 to $t-1$ to predict the next word. This is typically achieved by *feedback* connections, where the hidden layer activations used for predicting the word in position $t-1$ are fed back into the network to compute the hidden layer activations for predicting the next word. The hidden layer thus stores the history of all previous words. We use the RNNLM architecture as described in Mikolov (2012), shown in Figure 3. The input to the RNNLM at position t are the one-hot representation of the current word, $\mathbf{w}(t)$, and the activations from the hidden layer at position $t-1$, $\mathbf{s}(t-1)$. The output of the hidden layer at position $t-1$ is

$$\mathbf{s}(t) = \phi\left(\mathbf{D}\mathbf{w}(t) + \mathbf{W}\mathbf{s}(t-1)\right),$$

where $\mathbf{D}$ is the matrix of input word embeddings, $\mathbf{W}$ is a matrix that transforms the activations from the hidden layer in position $t-1$, and ϕ is a sigmoid function, defined as $\phi(x) = \frac{1}{1+\exp(-x)}$, that is applied elementwise. We need to compute the probability of the next word $\mathbf{w}(t+1)$ given the hidden state $\mathbf{s}(t)$. For fast estimation of output word probabilities, Mikolov (2012) divides the computation into two stages: First, the probability distribution over *word classes* is computed, after which the probability distribution over the subset of words belonging to the class are computed. The class probability of a particular class with index m at position t is computed as:

$$P(\mathbf{c}_m(t) \mid \mathbf{s}(t)) = \frac{\exp\left(\mathbf{s}(t)\mathbf{X}\mathbf{v}_m\right)}{\sum_{c=1}^{C}\left(\exp\left(\mathbf{s}(t)\mathbf{X}\mathbf{v}_c\right)\right)},$$

where $\mathbf{X}$ is a matrix of class embeddings and $\mathbf{v_m}$ is a one-hot vector representing the class with index m. The normalization constant is computed over all classes C. Each class specifies a subset V' of words, potentially smaller than the entire vocabulary V. The probability of an output word l at position $t+1$ given that its class is m is defined as:

$$P(\mathbf{y}_l(t+1) \mid \mathbf{c}_m(t), \mathbf{s}(t)) = \frac{\exp\left(\mathbf{s}(t)\mathbf{D}'\mathbf{v}_l\right)}{\sum_{k=1}^{V'}\left(\exp\left(\mathbf{s}(t)\mathbf{D}'\mathbf{v}_k\right)\right)},$$

where $\mathbf{D}'$ is a matrix of output word embeddings and $\mathbf{v}_l$ is a one hot vector representing the word with index l. The probability of the word $\mathbf{w}(t+1)$ given its class c_i can now be computed as:

$$P(\mathbf{w}(t+1) \mid \mathbf{s}(t)) = P(\mathbf{w}(t+1) \mid c_i, \mathbf{s}(t))\, P(c_i \mid \mathbf{s}(t)).$$

Neural Probabilistic Language Model

We use the feedforward neural probabilistic language model architecture of Vaswani et al. (2013), as shown in Figure 4. Each context $\mathbf{u}$ comprises a sequence of words $\mathbf{u}_j$ ($1 \le j \le n-1$) represented as one-hot vectors, which are fed as input to the neural network. At the output layer, the neural network computes the probability $P(w \mid \mathbf{u})$ for each word w, as follows.

The output of the first hidden layer h_1 is

$$h_1 = \phi\left(\sum_{j=1}^{n-1}\mathbf{C}^j\mathbf{D}\mathbf{u}_j + \mathbf{b}_1\right),$$

where $\mathbf{D}$ is a matrix of input word embeddings which is shared across all positions, the $\mathbf{C}^j$ are the context matrices for each word in $\mathbf{u}$, $\mathbf{b}_1$ is a vector of biases with the same dimension as h_1, and ϕ is applied elementwise. Vaswani et al. (2013) use rectified linear units (Nair and Hinton, 2010) for

the hidden layers h_1 and h_2, which use the activation function $\phi(x) = \max(0, x)$.

The output of the second layer h_2 is

$$h_2 = \phi\left(\mathbf{M}h_1 + \mathbf{b}_2\right),$$

where $\mathbf{M}$ is a weight matrix between h_1 and h_2 and $\mathbf{b}_2$ is a vector of biases for h_2. The probability of the output word is computed at the output softmax layer as:

$$P(w \mid \mathbf{u}) = \frac{\exp\left(\mathbf{v}_w \mathbf{D}' h_2 + \mathbf{b}^T \mathbf{v}_w\right)}{\sum_{w'=1}^{V} \exp\left(\mathbf{v}_{w'} \mathbf{D}' h_2 + \mathbf{b}^T \mathbf{v}_{w'}\right)},$$

where $\mathbf{D}'$ is the matrix of output word embeddings, $\mathbf{b}$ is a vector of biases for every output word and $\mathbf{v}_w$ its the one hot representation of the word w in the vocabulary.

2 Methods

We describe in this section our approach. In summary, we trained the neural network models on a Harry Potter fan fiction database. We then ran these models on chapter 9 of *Harry Potter and the Sorcerer's Stone* (Rowling, 2012) and computed the context and embedding vectors and the output probability for each word. In parallel, 3 subjects read the same chapter in an MEG scanner. We build models that predict the MEG data for each word as a function of the different neural network constituents. We then test these models with a classification task that we explain below. We detect correspondences between the neural network components and the brain processes that underlie reading in the following fashion. If using a neural network vector (e.g. the RNNLM embedding vector) allows us to classify significantly better than chance in a given region of the brain at a given time (e.g. the visual cortex at time 100-200ms), then we can hypothesize a relationship between that neural network constituent and the time/location of the analogous brain process.

2.1 Training the Neural Networks

We used the freely available training tools provided by Mikolov (2012)[1] and Vaswani et al. (2013)[2] to train our RNNLM and NPLM models used in our brain data classification experiments. Our training data comprised around 67.5 million words for training and 100 thousand words for validation from the Harry Potter fan fiction database (http://harrypotterfanfiction.com). We restricted the vocabulary to the top 100 thousand words which covered all but 4 words from Chapter 9 of *Harry Potter and the Sorcerer's Stone*.

For the RNNLM, we trained models with different hidden layers and learning rates and found the RNNLM with 250 hidden units to perform best on the validation set. We extracted our word embeddings from the input matrix $\mathbf{D}$ (Figure 3). We used the default settings for all other hyper parameters.

We trained 3-gram and 5-gram NPLMs with 150 dimensional word embeddings and experimented with different number of units for the first hidden layer (h_1 in Figure 4), and different learning rates. For both the 3-gram and 5-gram models, we found 750 hidden units to perform the best on the validation set and chose those models for our final experiments. We used the output word embeddings $\mathbf{D}'$ in our experiments. We visually inspected the nearest neighbors in the 150 dimensional word embedding space for some words and didn't find the neighbors from $\mathbf{D}'$ or $\mathbf{D}$ to be distinctly better than each other. We leave the comparison of input and output embeddings on brain activity prediction for future work.

2.2 MEG paradigm

We recorded MEG data for three subjects (2 females and one male) while they read chapter 9 of *Harry Potter and the Sorcerer's Stone* (Rowling, 2012). The participants were native English speakers and right handed. They were chosen to be familiar with the material: we made sure they had read the Harry Potter books or seen the movies series and were familiar with the characters and the story. All the participants signed the consent form, which was approved by the University of Pittsburgh Institutional Review Board, and were compensated for their participation.

The words of the story were presented in rapid serial visual format (Buchweitz et al., 2009): words were presented one by one at the center of the screen for 0.5 seconds each. The text was shown in 4 experimental blocks of ∼11 minutes. In total, 5176 words were presented. Chapter 9 was presented in its entirety without modifications and each subject read the chapter only once.

One can think of an MEG machine as a large helmet, with sensors located on the helmet that

[1] http://rnnlm.org/

[2] http://nlg.isi.edu/software/nplm

record the magnetic activity. Our MEG recordings were acquired on an Elekta Neuromag device at the University of Pittsburgh Medical Center Presbyterian Hospital. This machine has 306 sensors distributed into 102 locations on the surface of the subject's head. Each location groups 3 sensors or two types: one magnometer that records the intensity of the magnetic field and two planar gradiometers that record the change in the magnetic field along two orthogonal planes[3].

Our sampling frequency was 1kHz. For preprocessing, we used Signal Space Separation method (SSS, (Taulu et al., 2004)), followed by its temporal extension (tSSS, (Taulu and Simola, 2006)).

For each subject, the experiment data consists therefore of a 306 dimensional time series of length ∼45 minutes. We averaged the signal in every sensor into 100ms non-overlapping time bins. Since words were presented for 500ms each, we therefore obtain for every word $p = 306 \times 5$ values corresponding to 306 vectors of 5 points.

2.3 Decoding experiment

To find which parts of brain activity are related to the neural network constituents (e.g. the RNNLM context vector), we run a prediction and classification experiment in a 10-fold cross validated fashion. At every fold, we train a linear model to predict MEG data as a function of one of the feature sets, using 90% of the data. On the remaining 10% of the data, we run a classification experiment.

MEG data is very noisy. Therefore, classifying single word waveforms yields a low accuracy, peaking at 60%, which might lead to false negatives when looking for correspondences between neural network features and brain data. To reveal informative features, one can boost signal by either having several repetitions of the stimuli in the experiment and then averaging (Sudre et al., 2012) or by combining the words into larger chunks (Wehbe et al., 2014). We chose the latter because the former sacrifices word and feature diversity.

At testing, we therefore repeat the following 300 times. Two sets of words are chosen randomly from the test fold. To form the first set, 20 words are sampled without replacement from the test sample (unseen by the classifier). To form the second set, the k^{th} word is chosen randomly from all words in the test fold having the same length as the k^{th} word of the first set. Since every fold of the data was used 9 times in the training phase and once in the testing phase, and since we use a high number of randomized comparisons, this averages out biases in the accuracy estimation. Classifying sets of 20 words improves the classification accuracy greatly while lowering its variance and makes it dissociable from chance performance. We compare only between words of equal length, to minimize the effect of the low level visual features on the classification accuracy.

After averaging out the results of multiple folds, we end up with average accuracies that reveal how related one of the models' constituents (e.g. the RNNLM context vector) is to brain data.

2.3.1 Annotation of the stimulus text

We have 9 sets of annotations for the words of the experiment. Each set j can be described as a matrix $\mathbf{F}_j$ in which each row i corresponds to the vector of annotations of word i. Our annotations correspond to the 3 model constituents for each of the 3 models: the hidden layer representation before word i, the output probability of word i and the learned embeddings for word i.

2.3.2 Classification

In order to align the brain processes and the different constituents of the different models, we use a classification task. The task is to classify the word a subject is reading out of two possible choices from its MEG recording. The classifier uses one type of feature in an intermediate classification step. For example, the classifier learns to predict the MEG activity for any setting of the RNNLM hidden layer. Given an unseen MEG recording for an unknown word i and two possible story words i' and i'' (one of which being the true word i), the classifier predicts the MEG activity when reading i' and i'' from their hidden layer vectors. It then assigns the label i' or i'' to the word recording i depending on which prediction is the closest to the recording. The following are the detailed steps of this complex classification task. However, for the rest of the paper the most useful point to keep in mind is that the main purpose of the classification is to find a correspondence between the brain data and a given feature set j.

1. Normalize the columns of $\mathbf{M}$ (zero mean, standard deviation = 1). Pick feature set $\mathbf{F}_j$ and normalize its columns to a minimum of 0 and a maximum of 1.

[3] In this paper, we treat these three different sensors as three different dimensions without further exploiting their physical properties.

2. Divide the data into 10 folds, for each fold b:

   (a) Isolate $\mathbf{M}^b$ and $\mathbf{F}_j^b$ as test data. The remainder $\mathbf{M}^{-b}$ and $\mathbf{F}_j^{-b}$ will be used for training[4].

   (b) Subtract the mean of the columns of $\mathbf{M}^{-b}$ from $\mathbf{M}^b$ and $\mathbf{M}^{-b}$ and the mean of the columns of $\mathbf{F}_j^{-b}$ from $\mathbf{F}_j^b$ and $\mathbf{F}_j^{-b}$

   (c) Use ridge regression to solve
   $$\mathbf{M}^{-b} = \mathbf{F}_j^{-b} \times \boldsymbol{\beta}_j^t$$
   by tuning the λ parameter to every one of the p output dimensions independently. λ is chosen via generalized cross validation (Golub et al., 1979).

   (d) Perform a binary classification. Sample from the set of words in b a set c of 20 words. Then sample from b another set of 20 words such that the k^{th} word in c and d have the same number of letters. For every sample (c,d):

      i. predict the MEG data for c and d as: $\mathbf{P}^c = \mathbf{F}_j^c \times \mathbf{\Gamma}_j^b$ and $\mathbf{P}^d = \mathbf{F}_j^d \times \mathbf{\Gamma}_j^b$

      ii. assign to $\mathbf{M}^c$ the label c or d depending on which of $\mathbf{P}^c$ or $\mathbf{P}^d$ is closest (Euclidean distance).

      iii. assign to $\mathbf{M}^d$ the label c or d depending on which of $\mathbf{P}^c$ or $\mathbf{P}^d$ is closest (Euclidean distance).

3. Compute the average accuracy.

2.3.3 Restricting the analysis spatially: a searchlight equivalent

We adapt the searchlight method (Kriegeskorte et al., 2006) to MEG. The searchlight is a discovery procedure used in fMRI in which a cube is slid over the brain and an analysis is performed in each location separately. It allows to find regions in the brain where a specific phenomenon is occurring. In the MEG sensor space, for every one of the 102 sensor locations ℓ, we assign a group of sensors g_ℓ. For every location ℓ, we identify the locations that immediately surround it in any direction (Anterior, Right Anterior, Right etc...) when looking at the 2D flat representation of the location of the sensors in the MEG helmet (see Fig. 9 for an illustration of the 2D helmet). g_ℓ therefore contains the 3 sensors at location ℓ and at the neighboring locations. The maximum number of sensors in a group is 3×9. The locations at the edge of the helmet have fewer sensors because of the missing neighbor locations.

[4]The rows from $\mathbf{M}^{-b}$ and $\mathbf{F}_j^{-b}$ that correspond to the five words before or after the test set are ignored in order to make the test set independent.

2.3.4 Restricting the analysis temporally

Instead of using the entire time course of the word, we can use only one of the corresponding 100ms time windows. Obtaining a high classification accuracy using one of the time windows and feature set j means that the analogous type of information is encoded at that time.

2.3.5 Classification accuracy by time and region

The above steps compute whole brain accuracy using all the time series. In order to perform a more precise spatio-temporal analysis, one can use only one time window m and one location ℓ for the classification. This can answer the question of when and where different information is represented by brain activity. For every location, we will use only the columns corresponding to the time point m for the sensors belonging to the group g_ℓ. Step (d) of the classification procedure is changed as such:

(d) Perform a binary classification. Sample from the set of words in b a set c of 20 words. Then sample from b another set of 20 words such that the k^{th} word in c and d have the same number of letters. For every sample (c,d), and for every setting of $\{m, \ell\}$:

   i. predict the MEG data for c and d as: $\mathbf{P}^c_{\{m,\ell\}} = \mathbf{F}_j^c \times \mathbf{\Gamma}^b_{j,\{m,\ell\}}$ and $\mathbf{P}^d_{\{m,\ell\}} = \mathbf{F}_j^d \times \mathbf{\Gamma}^b_{j,\{m,\ell\}}$

   ii. assign to $\mathbf{M}^c_{\{m,\ell\}}$ the label c or d depending on which of $\mathbf{P}^c_{\{m,\ell\}}$ or $\mathbf{P}^d_{\{m,\ell\}}$ is closest (Euclidean distance).

   iii. assign to $\mathbf{M}^d_{\{m,\ell\}}$ the label c or d depending on which of $\mathbf{P}^c_{\{m,\ell\}}$ or $\mathbf{P}^d_{\{m,\ell\}}$ is closest (Euclidean distance).

2.3.6 Statistical significance testing

We determine the distribution for chance performance empirically. Because the successive word samples in our MEG and feature matrices are not independent and identically distributed, we break the relationship between the MEG and feature matrices by shifting the feature matrices by large delays (e.g. 2000 to 2500 words) and we repeat the classification using the delayed matrices. This simulates chance performance more fairly than a permutation test because it keeps the time structure of the matrices. It was used in (Wehbe et al.,

2014) and inspired by (Chwialkowski and Gretton, 2014). For every $\{m, \ell\}$ setting we can therefore compute a standardized z-value by subtracting the mean of the shifted classifications and dividing by the standard deviation. We then compute the p-value for the true classification accuracy being due to chance. Since the three p-values for the three subjects for a given $\{m, \ell\}$ are independent, we combine them using Fisher's method for independent test statistics (Fisher, 1925). The statistics we obtain for every $\{m, \ell\}$ are dependent because they comprise nearby time and space windows. We control the false discovery rate using (Benjamini and Yekutieli, 2001) to adjust for the testing at multiple locations and time windows. This method doesn't assume any kind of independence or positive dependence.

3 Results

We present in Fig. 5 the accuracy using all the time windows and sensors. In Fig. 6 we present the classification accuracy when running the classification at every time window exclusively. In Fig. 9 we present the accuracy when running the classification using different time windows and groups of sensors centered at every one of the 102 locations.

It is important to lay down some conventions to understand the complex results in these plots. To recap, we are trying to find parallels between model constituents and brain processes. We use:

- a subset of the data (for example the time window 0-100ms and all the sensors)
- one type of feature (for example the hidden context layer from the NPLM 3g model)

and we obtain a classification accuracy A. If A is low, there is probably no relationship between the feature set and the subset of data. If A is high, it hints to an association between the subset of data and the mental process that is analogous to the feature set. For example, when using all the sensors and time window 0-100ms, along with the NPLM 3g hidden layer, we obtain an accuracy of 0.70 (higher than chance with p $< 10^{-14}$, see Fig. 6). Since the NPLM 3g hidden layer summarizes the context of the story before seeing word i, this suggests that the brain is still processing the context of the story before word i between 0-100ms.

Fig. 6 shows the accuracy for different types of features when using all of the time points and all the sensors to classify a word. We can see

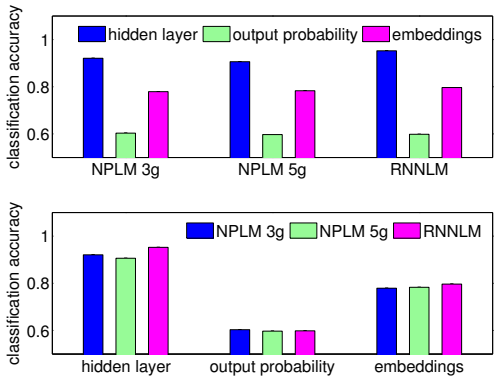

Figure 5: Average accuracy using all time windows and sensors, grouped by model (top) and type of feature (bottom). All accuracies are significantly higher than chance ($p < 10^{-8}$).

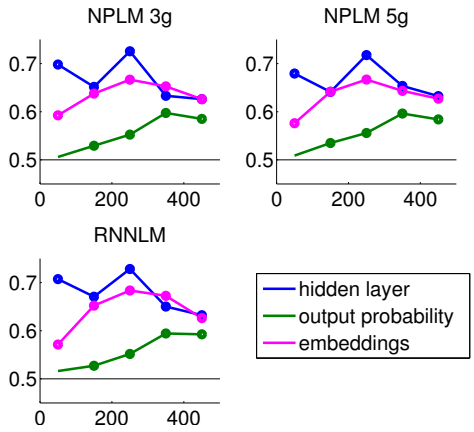

Figure 6: Average accuracy in different time windows when using different types of features as input to the classifier, for different models. Accuracy is plotted in the center of the respective time window. Points marked with a circle are significantly higher than chance accuracy for the given feature set and time window after correction.

similar classification accuracies for the three types of models, with RNNLM ahead for the hidden layer and embeddings and behind for the output probability features. The hidden layer features are the most powerful for classification. Between the three types of features, the hidden layer features are the best at capturing the information contained in the brain data, suggesting that most of the brain activity is encoding the previous context. The embedding features are the second best. Finally the output probability have the smallest accuracies. This makes sense considering that they capture much less information than the other two high dimensional descriptive vectors, as they do not represent the complex properties of the words, only a numerical assessment of their likelihood.

Fig. 6 shows the accuracy when using different windows of time exclusively, for the 100ms time

windows starting at $0, 100 \ldots 400$ms after word presentation. We can see that using the embedding vector becomes increasingly more useful for classification until 300-400ms, and then its performance starts decreasing. This results aligns with the following hypothesis: the word is being perceived and understood by the brain gradually after its presentation, and therefore the brain representation of the word becomes gradually similar to the neural network representation of the word (i.e. the embedding vector). The output probability feature accuracy peaks at a later time than the embeddings accuracy. Obtaining a higher than chance accuracy at time window m using the output probability as input to the classifier suggests strongly that the brain is integrating the word at time window m, because it is responding differently for predictable and unpredictable words[5]. The integration step happens after the perception step, which is probably why the output probability curves peak later than the embeddings curves.

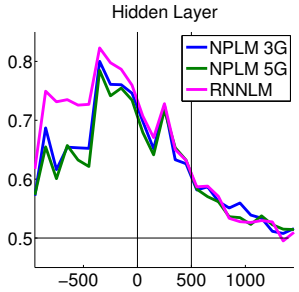

Figure 7: Average accuracy in time for the different hidden layers. The analysis is extended to the time windows before and after the word is presented, the input feature is restricted to be the hidden layer before the central word is seen. The first vertical bar indicates the onset of the word, the second one indicates the end of its presentation.

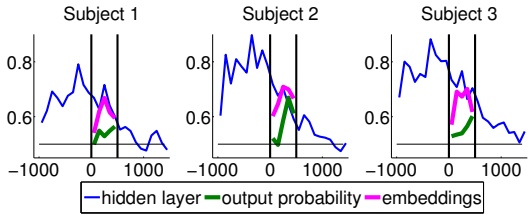

Figure 8: Accuracy in time when using the RNNLM features for each of the three subjects.

To understand the time dynamics of the hidden layer accuracy we need to see a larger time scale than the word itself. The hidden layer captures the context before word i is seen. Therefore it seems reasonable that the hidden layer is not only related to the activity when the word is on the screen, but also related to the activity before the word is presented, which is the time when the brain is integrating the previous words to build that context. On the other hand, as the word i and subsequent words are integrated, the context starts diverging from the context of word i (computed before seeing word i). We therefore ran the same analysis as before, but this time we also included the time windows before and after word i in the analysis, while maintaining the hidden layer vector to be the context before word i is seen. We see the behavior we predicted in the results: the context before seeing word i becomes gradually more useful for classification until word i is seen, and then it gradually decreases until it is no longer useful since the context has changed. We observe the RNNLM hidden layer has a higher classification accuracy than the finite context NPLMs. This might be due to the fact that the RNNLM has a more complete representation of context that captures more of the properties of the previous words.

To show the consistency of the results, we plot as illustration the three curves we obtain for each subject for the RNNLM (Fig. 8). The patterns seem very consistent indicating the phenomena we described can be detected at the subject level.

We now move on to the spatial decomposition of the analysis. When the visual input enters the brain, it first reaches the visual cortex at the back of the head, and then moves anteriorly towards the left and right temporal cortices and eventually the frontal cortex. As it flows through these areas, it is processed to higher levels of interpretations. In Fig. 9, we plot the accuracy for different regions of the brain and different time windows for the RNNLM features. To make the plots simpler we multiplied by zero the accuracies which were not significantly higher than chance. We expand a few characteristic plots. We see that in the back of the head the embedding features have an accuracy that seems to peak very early on. As we move forward in the brain towards the left and right temporal cortices, we see the embeddings accuracy peaking at a later time, reflecting the delay it takes for the information to reach this part of the brain. The output probability start being useful for classification after the embeddings, and specifically in the left temporal cortex which is the cite where the N400

[5]the fact that we can classify accurately during windows 300-400ms indicates that the classifier is taking advantage of the N400 response discussed in the introduction

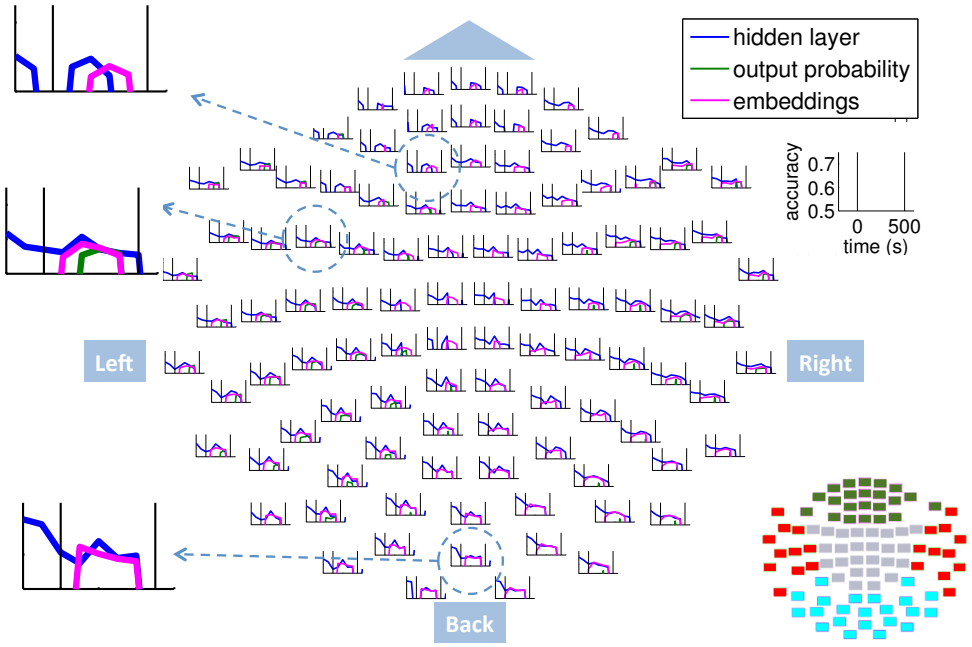

Figure 9: Average accuracy in time and space on the MEG helmet when using the RNNLM features. For each of the 102 locations the average accuracy for the group of sensors centered at that location is plotted versus time. The axes are defined in the rightmost, empty plot. Three plots have been magnified to show the increasing delay in high accuracy when using the embeddings feature, reflecting the delay in processing the incoming word as information travels through the brain. A sensor map is provided in the lower right corner: visual cortex = cyan, temporal = red, frontal = dark green.

is reported in the literature. Finally, as we reach the frontal cortex, we see that the embeddings features have an even later accuracy peak.

4 Conclusion and contributions

Novel brain data exploration We present here a novel and revealing approach to shed light on the brain processes involved in reading. This is a departure from the classical approach of controlling for a few variables in the text (e.g. showing a sentence with an expected target word versus an unexpected one). While we cannot make clear cut causal claims because we did not control for our variables, we are able to explore the data much more and offer a much richer interpretation than is possible with artificially constrained stimuli.

Comparing two models of language Adding brain data into the equation allowed us to compare the performance of the models and to identify a slight advantage for the RNNLM in capturing the text contents. Numerical comparison is however a secondary contribution of our approach. We showed that it might be possible to use brain data to understand, interpret and illustrate what exactly is being encoded by the obscure vectors that neural networks compute, by drawing parallels between the models constituents and brain processes.

Anecdotally, in the process of running the experiments, we noticed that the accuracy for the hidden layer of the RNNLM was peaking in the time window corresponding to word $i-2$, and that it was decreasing during word $i-1$. Since this was against our expectations, we went back and looked at the code and found that it was indeed returning a delayed value and corrected the features. We therefore used the brain data in order to correct a mis-specification in our neural network model. This hints if not proves the potential of our approach for assessing language models.

Future Work The work described here is our first attempt along the promising endeavor of matching complex computational models of language with brain processes using brain recordings. We plan to extend our efforts by (1) collecting data from more subjects and using various types of text and (2) make the brain data help us with training better statistical language models by using it to determine whether the models are expressive enough or have reached a sufficient degree of convergence.

Acknowledgements

This research was supported in part by NICHD grant 5R01HD07328-02. We thank Nicole Rafidi for help with data acquisition.

References

Yoav Benjamini and Daniel Yekutieli. 2001. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188.

Augusto Buchweitz, Robert A Mason, Lêda Tomitch, and Marcel Adam Just. 2009. Brain activation for reading and listening comprehension: An fMRI study of modality effects and individual differences in language comprehension. *Psychology & neuroscience*, 2(2):111–123.

Kacper Chwialkowski and Arthur Gretton. 2014. A kernel independence test for random processes. *arXiv preprint arXiv:1402.4501*.

Ronald Aylmer Fisher. 1925. *Statistical methods for research workers*. Genesis Publishing Pvt Ltd.

Stefan L Frank, Leun J Otten, Giulia Galli, and Gabriella Vigliocco. 2013. Word surprisal predicts N400 amplitude during reading. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics*, pages 878–883.

Angela D Friederici. 2002. Towards a neural basis of auditory sentence processing. *Trends in cognitive sciences*, 6(2):78–84.

Gene H Golub, Michael Heath, and Grace Wahba. 1979. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223.

Peter Hagoort. 2003. How the brain solves the binding problem for language: a neurocomputational model of syntactic processing. *Neuroimage*, 20:S18–S29.

Nikolaus Kriegeskorte, Rainer Goebel, and Peter Bandettini. 2006. Information-based functional brain mapping. *Proceedings of the National Academy of Sciences of the United States of America*, 103(10):3863–3868.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and J Cernocky. 2011. RNNLM-recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201.

Tomas Mikolov. 2012. *Statistical Language Models Based on Neural Networks*. Ph.D. thesis, Brno University of Technology.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of ICML*, pages 807–814.

Joanne K. Rowling. 2012. *Harry Potter and the Sorcerer's Stone*. Harry Potter US. Pottermore Limited.

Riitta Salmelin. 2007. Clinical neurophysiology of language: the MEG approach. *Clinical Neurophysiology*, 118(2):237–254.

Gustavo Sudre, Dean Pomerleau, Mark Palatucci, Leila Wehbe, Alona Fyshe, Riitta Salmelin, and Tom Mitchell. 2012. Tracking neural coding of perceptual and semantic features of concrete nouns. *NeuroImage*, 62(1):451–463.

Samu Taulu and Juha Simola. 2006. Spatiotemporal signal space separation method for rejecting nearby interference in MEG measurements. *Physics in medicine and biology*, 51(7):1759.

Samu Taulu, Matti Kajola, and Juha Simola. 2004. Suppression of interference and artifacts by the signal space separation method. *Brain topography*, 16(4):269–275.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation.

Leila Wehbe, Brian Murphy, Partha Talukdar, Alona Fyshe, Aaditya Ramdas, and Tom Mitchell. 2014. Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses. *in press*.