From Language to Language-ish: How Brain-Like is an LSTM's Representation of Nonsensical Language Stimuli?

Maryam Hashemzadeh[1,2,*], Greta Kaufeld[3,4], Martha White[1,2], Andrea E. Martin[3,5], and Alona Fyshe[1,2,6]

[1] Department of Computing Science, University of Alberta, Alberta, Canada
[2] Alberta Machine Intelligence Institute (Amii), Alberta, Canada
[3] Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands
[4] International Max Planck Research School for Language Sciences, Nijmegen, The Netherlands
[5] Donders Institute for Brain, Cognition, and Behaviour, Radboud University, Nijmegen, The Netherlands
[6] Department of Psychology, University of Alberta, Alberta, Canada
[*] *Corresponding author, hashemza@ualberta.ca*
{*whitem,alona*}*@ualberta.ca,* {*greta.kaufeld, andrea.martin*}*@mpi.nl*

Abstract

The representations generated by many models of language (word embeddings, recurrent neural networks and transformers) correlate to brain activity recorded while people read. However, these decoding results are usually based on the brain's reaction to syntactically and semantically sound language stimuli. In this study, we asked: how does an LSTM (long short term memory) language model, trained (by and large) on semantically and syntactically intact language, represent a language sample with degraded semantic or syntactic information? Does the LSTM representation still resemble the brain's reaction? We found that, even for some kinds of nonsensical language, there is a statistically significant relationship between the brain's activity and the representations of an LSTM. This indicates that, at least in some instances, LSTMs and the human brain handle nonsensical data similarly.

1 Introduction

When people read or listen to language, brain imaging studies have shown us that the brain's activity correlates to LSTM (long short term memory) state representations for the same text (Jain and Huth, 2018; Toneva and Wehbe, 2019). In those studies (and others like them) the stimuli used to test for this correlation was based on language with no errors.[1] This implies that during the processing of within-distribution data (i.e. well-formed sentences/stories), LSTMs and the human brain show similar representational patterns. But what happens when language is out-of-distribution (e.g. nonsensical sentences or pseudo-words)? Can we expect that an LSTM will still compute contextual states in a way that resembles how the human brain reacts? I.e. is there a correlation between LSTM representations and neural activity when the stimuli is not a predictable language sample? Answering these questions could provide evidence that an LSTM is able to generalize to new data in a human-like way, even when the new data is unlike anything it encountered during training. Our answers could also help psycholinguists reason about the efficacy of nonsensical sentences and pseudo-words as syntax-only stimuli controls.

Here we use brain imaging data (Electroencephalography, EEG) collected in three conditions, *Sentence*: well-formed grammatical sentences, *Jabberwocky*: pseudo-word sentences that preserved word order, morphosyntax, and sentential prosody without lexical or compositional semantics, and *Word-list*: the words of the *Sentence* condition in a pseudo-random order without sentence prosody, syntax, or compositional meaning. We ran a character-level LSTM model on the stimuli, and trained a *decoding model* to predict the LSTM's internal representations from EEG signals. Using data from the *Sentence* condition, we corroborated previous results and showed that LSTM representations are correlated with brain activity for within-distribution language. But, when it came to nonsensical language stimuli, it was unclear if LSTM representations would still correlate to brain activity. Our original hypothesis was that LSTM representations for out-of-distribution language would no longer correlate to brain activity. However, we found that our decoding model worked quite well even when all content words of the stimuli were pseudo-words (*Jabberwocky*).

To summarize, we show that:

[1] Nonsensical language is often used when measuring Event Related Potentials. Here we speak of decoding studies only.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 645–656
November 16 - 20, 2020.

- Our decoding models work well in both the *Sentence* and *Jabberwocky* conditions, but not in the *Word-list* condition.
- The syntactic signatures available in *Sentence* and *Jabberwocky* LSTM representations are similar, and can be predicted from either the *Sentence* or *Jabberwocky* EEG.
- For some LSTM representations, the decoding model's *weight maps* generalize between *Jabberwocky* and *Sentence* EEG data.
- From our results, we can infer which LSTM representations encode semantic and/or syntactic information. We confirm using syntactic and semantic probing tasks.

Our results show that there are similarities between the way the brain and an LSTM represent stimuli from both the *Sentence* (within-distribution) and *Jabberwocky* (out-of-distribution) conditions.

2 Materials and Methods

2.1 Data description

Our data was originally collected to contrast the brain's response to language samples that vary the amount of semantic and syntactic information (Kaufeld et al., 2020). The dataset consists of EEG recordings (64 channels, 500 Hz sampling rate) of 27 native Dutch speakers (9 males; mean age= 23). The participants listened to a native Dutch speaker in three conditions: *Sentence*, *Jabberwocky*, and *Word-list*. Each condition has 80 sentences, and all *Sentence* and *Jabberwocky* stimuli sentences share the same grammatical structure.

The *Sentence* stimuli contain two coordinate clauses and a conjunction with the structure *[Adj N V N Conj Det Adj N V N]*, and contain lexical semantics, compositional semantics, and syntax. *Word-list* consists of the same ten words as *Sentence* but in a pseudo-random order with infeasible syntactic structures (either *[V V Adj Adj Det Conj N N N N]*, or *[N N N N Det Conj V V Adj Adj]*). The *Word-list* condition leaves orthography/phonology intact and contains lexical semantics, but not compositional semantics or syntax. For *Jabberwocky*, words from the *Sentence* condition are replaced with pseudo-words created with the Wuggy generator (Keuleers and Brysbaert, 2010). Crucially, the ***Jabberwocky* pseudo-words appear in the same order as the corresponding words in the *Sentence* condition**. The Wuggy generator alters words in a way that obeys the phonotactic and morphosyntactic constraints of a language, but eliminates semantic meaning. The *Jabberwocky* condition contains syntax (and morphosyntax, which is preserved by Wuggy). Amongst psycholinguists and cognitive neuroscientists, it is widely accepted that Jabberwocky does not contain lexical or compositional semantics, and a Jabberwocky condition is often used to control for semantics (Humphries et al., 2006; Fedorenko et al., 2012; Friederici et al., 2000). Anecdotally, native Dutch speakers typically cannot guess the true word when presented with the pseudo-word.

Stimuli examples:

- *Sentence*: Lange mannen bouwen huisjes en de lieve honden brengen planken. (Tall men build houses and the sweet dogs bring boards.)
- *Jabberwocky*: Lalve wanzen botren raasjes en de reeve rorden brargen sponken.
- *Word-list*: planken mannen huisjes honden de en bouwen brengen lange lieve

In the *Jabberwocky* condition the determiners and conjunctions are not pseudo-words. To fairly compare the conditions, we removed these words from all three conditions during our analyses. Due to the nature of spoken language, the time-duration each of word/pseudo-word differs. To account for this, we considered the first 400 ms of EEG after word/pseudo-word onset.

To improve the EEG's signal to noise ratio, we average the EEG recording for a given sentence across all subjects. Though this reduces participant-specific signal, we have found it to be the best way to decode from EEG data. For this data, models trained on only one subject did not perform above chance. For each word of each stimulus sentence S, we concatenated the recording from every electrode into one vector $R_t \in \mathbb{R}^{1\times D}$ where D is the total number of readings across all sensors (here $D = 12800$: 64 sensors $\times$ 200 time points).

2.2 Decoding model

The aim of a decoding model is to find a mapping function $f(R_t) \rightarrow g(S_{1:t})$ between an EEG recording R_t of the brain's response to word w_t and a language model's representation of stimulus $S_{1:t}$ (the words of a sentence up to and including word w_t). Our methodology closely followed (Jain and Huth, 2018). We instantiate our mapping function in two steps:

1. $g(S_{1:t}) \in \mathbb{R}^{1\times P}$: an LSTM's P-dimensional representation for word w_t, conditioned on context $w_1, \ldots w_{t-1}$.

2. $f(R_t)$: a regularized linear regression to map the EEG signal R_t to $g(S_{1:t})$.

Figure 1 shows a schematic of the decoding model.

1- Derive LSTM representations ($g(S)$): The *Jabberwocky* stimuli are made of pseudo-words, so we needed a language model that can handle out-of-vocabulary input. We used the state-of-the-art character-level LSTM language model proposed by Kim et al. (2016), but used three LSTM layers based on previous decoding work (Jain and Huth, 2018). This LSTM operates on the characters of incoming words (so it can handle pseudo-words), but it produces predictions at the word level. Each input character has its own embedding, which are concatenated and fed to convolutional layers. The convolved values are passed to a highway network, whose output is fed through three stacked LSTM layers before predicting the next word. In the decoding analyses that follow, the $g(S)$ vectors we analyze are (1) the concatenation of the character embeddings called *Embedding* layer, (2) the concatenation of the Convolutional layers called *Conv*, and ($\{$3-5$\}$) the three LSTM layers called *LSTM1-3*. We will use the term *LSTM* to refer to the full character-based model, *LSTM representation* to refer to any of the $g(S)$ vectors types, and *LSTM layer* or *LSTM1-3* to refer specifically to the LSTM layers within the larger LSTM model.

We trained the LSTM on one million sentences from Dutch Wikipedia. We set the number of epochs to 40, batch size is 50, and sequence length is 20. We used a stochastic gradient descent optimizer with sparse categorical cross-entropy loss. The initial learning rate is 0.8 with inverse time decay rate 0.5.

For Dutch Wikipedia, the average test perplexity of our model is 108.12. When the inputs are the *Sentence* stimuli, the average perplexity is higher: 317.91. This is likely because the coordinate clauses within each stimulus are only 4 words long, which reduces the effective context. When the inputs are the *Jabberwocky* pseudo-words and the outputs are the corresponding *Sentence* next word, the perplexity is 325.12. These *Sentence* and *Jabberwocky* perplexities are not significantly different ($p = 0.967$). We calculated the average perplexity on the *Word-list* stimuli to be 1008.23, which indicates that (as expected) the network cannot predict the next words in the *Word-list* stimuli. This also shows that while the *Jabberwocky* and *Sentence* perplexities are higher than on Wikipedia, they are much lower than for stimuli with no contextual information.

For comparison, we also experimented with non-contextual word embeddings from Grave et al. (2018). This 300-dimensional model is pre-trained on Dutch Wikipedia using Continuous Bag of Words (CBOW) with position-weights.

2- Regularized linear regression ($f(R)$): We used ridge regression to test if the EEG data correlates with the word/pseudo-word representations. The regression function $f(R_t)$ is a linear transformation of R_t to predict the P-dimensional $g(S_{1:t})$: $f(R_t) = R_t\beta$ where $\beta \in \mathbb{R}^{D\times P}$.

2.3 Measuring model accuracy

We used Monte Carlo (MC) cross-validation to evaluate our decoding models. MC cross-validation affords a more stable estimate of model accuracy, and allows for statistically-sound comparisons of model performance. During each of our 200 MC samples, we swept the regression regularization parameter among the values in range $[0.1, 200]$ using 5-fold cross-validation on the training data only.

We use a 2 vs. 2 classification test to assess the performance of the learned model (Mitchell et al., 2008; Fyshe et al., 2019). During each cross-validation trial we randomly create groups of two from the held-out samples. Using a model fit to the training data, we produce predicted representations for the held-out samples. For simplicity, let $y_t^i = g(S_{1:t}^i)$ be the contextual representation for word w_t of sentence i. Then, for each group of 2 test samples $(S_{1:t1}^i, S_{1:t2}^j)$, we perform a *2 vs. 2 test* using the true representations (y_{t1}^i, y_{t2}^j) and predicted representations $(\widehat{y}_{t1}^i, \widehat{y}_{t2}^j)$. The 2 vs. 2 test compares the sum of cosine similarity for correctly matched the true and predicted vectors:

$$cos(y_{t1}^i, \widehat{y}_{t1}^i) + cos(y_{t2}^j, \widehat{y}_{t2}^j), \quad (1)$$

to the sum of cosine similarity of the mismatched vectors:

$$cos(y_{t1}^i, \widehat{y}_{t2}^j) + cos(y_{t2}^j, \widehat{y}_{t1}^i). \quad (2)$$

If Eq 1 is greater than Eq 2, the 2 vs. 2 test passes. 2 vs. 2 accuracy is the percentage of correct 2 vs. 2 tests, and chance 2 vs. 2 accuracy is 0.5. In addition to 2 vs. 2 accuracy, we also report mean-squared-error of the learned model in Appendix B.

To test for statistical significance, we used permutation tests. The LSTM representations for the stimuli were randomly shuffled such that the true representations $g(S_t)$ were no longer correctly

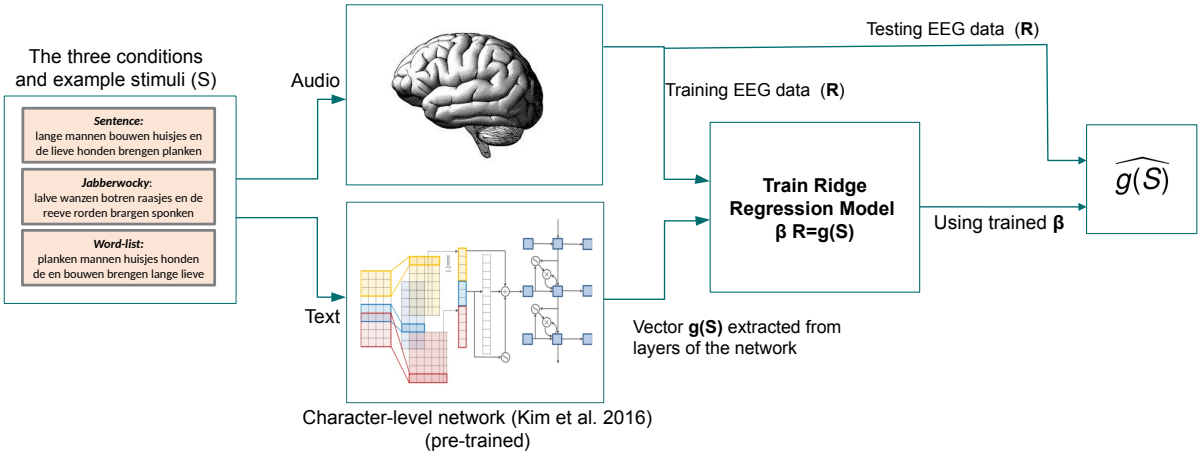


Figure 1: Decoding model. Each stimulus sentence is fed to a pre-trained language model to create a non-linear context-based representation. The hidden representations for a sentence (S) are extracted from each layer $g(S)$. Our ridge regression model is trained to use the EEG signal R to predict $g(S)$.

matched to the EEG data. We then trained and tested our decoding models as described above using > 1000 random permutations. These results represent the expected distribution of 2 vs. 2 accuracy when there is no relationship between the EEG data and the LSTM representations. From that (null) distribution we can compute p-values for our observed accuracy on the un-permuted representations. We correct for multiple comparisons using the Benjamini-Hochberg-Yekutieli False Discovery Rate (FDR) procedure (Benjamini and Yekutieli, 2001) using $\alpha = 0.05$.

For our models to perform above chance, there must be correlates of particular aspects of language (such as semantics or syntax) present in the brain activation data R, and in the corresponding contextual representation ($g(S)$). Furthermore, our decoding model assumes a linear relationship between R and $g(S)$. If our models do not perform above chance, any of the above conditions may be violated; our analyses are not designed to differentiate between the failure cases.

3 Results

We were interested in comparing the representations generated by an LSTM to that of the human brain, in response to both within- and out-of-distribution language. Our *Sentence* stimuli, which represent within-distribution language, contain semantic and syntactic information. We used two kinds of out-of-distribution stimuli: *Jabberwocky*, which was designed to have syntactic information only, and *Word-list*, which has only semantic information. We attempted to learn a mapping from EEG to LSTM representations (to test if the LSTM and brain handle the stimuli similarly). To begin, we examined the difference in the semantic and syntactic information encoded by each of the LSTM representations. Then, we developed analyses to test for a similarity in the representation of semantic and syntactic information across the experimental conditions. We investigated using the following questions:

1. Is there a difference in the semantic/syntactic information captured by the LSTM representations? (Probing tasks)
2. Can we learn a mapping from the EEG data to the LSTM representations in the *Sentences, Jabberwocky*, or *Word-list* conditions? Is there a difference in performance across the different LSTM representations? (Analysis 1: test for semantic and/or syntactic information)
3. If there is syntactic information present in the *Sentences* and *Jabberwocky* LSTM representations, is it exchangeable? (Analysis 2: swap the $g(S)$ conditions)
4. Do the actual *patterns* learned by the decoder generalize to EEG from the other condition? (Analysis 3: swap R at *test* time only)

The EEG analyses are summarized in Table 1.

3.1 Probing tasks

Previous work has found that LSTM layers encode differing amounts of information about semantic meaning and syntactic structure (McCann et al., 2017; Peters et al., 2018). To investigate the behavior of our LSTM, we used several probing task benchmarks. Because there are more avail-

Analysis	Case	Train EEG	Train $g(S)$	Test EEG	Test $g(S)$
1	1	*Sen*	*Sen*	*Sen*	*Sen*
	2	*Jab*	*Jab*	*Jab*	*Jab*
	3	*WL*	*WL*	*WL*	*WL*
2	1	*Sen*	*Jab*	*Sen*	*Jab*
	2	*Jab*	*Sen*	*Jab*	*Sen*
3	1	*Sen*	*Sen*	*Jab*	*Sen*
	2	*Jab*	*Jab*	*Sen*	*Jab*

Table 1: Data description for each analysis. Sen: Sentence, Jab: Jabberwocky, WL: Word-list. Analysis 1: EEG & $g(S)$ from the same condition. Analysis 2: $g(S)$ swapped between conditions. Analysis 3: EEG swapped between conditions *at test time only*.

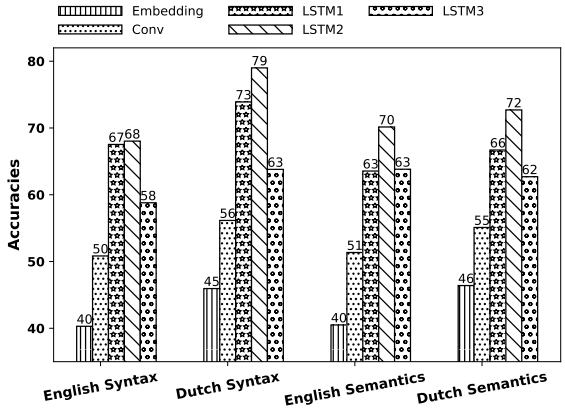


Figure 2: Average accuracies for the semantic/syntactic probing tasks using LSTM representations from Dutch or English LSTM language models.

able benchmarks for English, we also trained an identical LSTM architecture using English Penn Treebank (PTB) (Marcus et al., 1993), and checked the probing task results for consistency against the Dutch results. The English semantic and syntactic probing tasks are from Conneau et al. (2018), and the Dutch from Eichler et al. (2019). A description of each task is given in the Appendix (Table 2).

For each probing task we trained an MLP classifier with 2 hidden layers of 100 units. The MLP input is the average of the LSTM representations for a sentence, and the output is the predicted class of the sentence (e.g. past tense verb). Note that the sentences here are not from our stimuli, but rather from the probing tasks themselves.

The average accuracies for the English and Dutch probing tasks are shown in Fig. 2, and individual task accuracies appear in Table 3 of Appendix A. We were reassured to see the performance of the English and Dutch LSTMs show similar patterns. Both the Embedding and Conv layers perform poorly on the semantic and syntactic tasks. We see the strongest evidence for syntax in LSTM1 and LSTM2, and the strongest evidence for semantics in LSTM2.

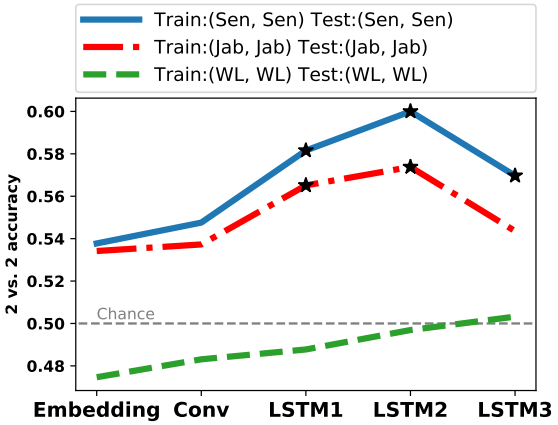


Figure 3: Analysis 1 (Test for semantic and syntactic information): 2 vs. 2 accuracy with $g(S)$/EEG from the same condition. The x-axis denotes LSTM representation ($g(S)$). Legend denotes EEG/LSTM representations used for train/test: (EEG condition, LSTM condition). "*Sen*": *Sentence*, "*Jab*": *Jabberwocky* , "*WL*": *Word-list*. $\star$: above chance ($p < 0.05$, FDR corrected).

3.2 Test for semantic and/or syntactic information (Analysis 1)

To test for the correlation of semantic and/or syntactic information between the EEG and LSTM representations, we measured the accuracy of a decoding model trained with data from the same condition. This is Analysis 1 from Table 1, and results are in Fig. 3.

Based on the probing results, for the *Sentence* stimuli we expected to see highest performance for LSTM2 (contains semantic and syntactic information), and somewhat lower performance for LSTM1 (strong syntax performance, but lower semantics). For the *Jabberwocky* condition, we expected to see strongest performance for the syntactically rich LSTM1 and LSTM2. For the *Word-list* condition, we were unsure if the contextual representations would work at all, given that the random ordering of words removes the sentence's context.

In the *Sentences* condition, the accuracy is statistically above chance for LSTM layers 1-3 (0.581, 0.600, and 0.569 respectively, $p < 0.05$, FDR corrected). This matched our predictions based on the probing tasks, and shows that LSTM3 has sufficient syntactic/semantic information for the decoding task. In the *Jabberwocky* condition, only

the accuracies of the LSTM1 and LSTM2 are statistically above chance with (0.565 and 0.573 respectively, $p < 0.05$, FDR corrected), which again matched our predictions based on the probing tasks. The *Sentence* condition conveys both semantic and syntactic information, and so the decoding model produces higher accuracy than the *Jabberwocky* condition, which lacks semantics. For both *Jabberwocky* and *Sentence* conditions, LSTM2 shows accuracy higher than LSTM1 and LSTM3, which is consistent with previous decoding work showing that middle LSTM layers outperformed early and late layers (Jain and Huth, 2018; Toneva and Wehbe, 2019).

When the decoding model is trained on data from the *Word-list* condition, no representation performs significantly different from chance ($p > 0.05$, FDR corrected). Because of this poor performance, Analyses 2 and 3 do not consider the *Word-list* condition. The accuracies for the Embedding and Conv layers are not significantly above chance for any condition ($p > 0.05$, FDR corrected).

We also trained our decoding models with non-contextual CBOW representations, and found the 2 vs. 2 accuracy to be 0.55 for the *Sentence* condition, and 0.54 for the *Word-list* condition, neither of which are above chance. Since the *Jabberwocky* stimuli are pseudo-words, we cannot test the 2 vs. 2 accuracy using this word-level model.

3.3 Swap the $g(S)$ conditions (Analysis 2)

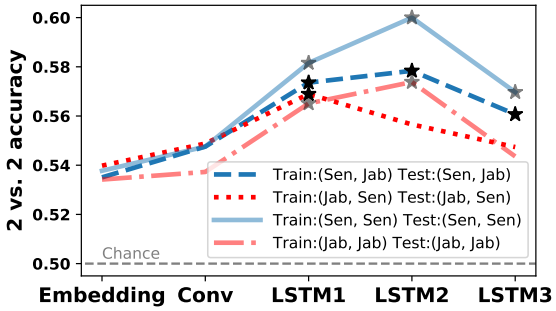
Analysis 1 showed that some LSTM representations could be decoded in the *Sentence* and *Jabberwocky* conditions. This tells us there is a relationship between the information in some LSTM representations and the corresponding EEG data. But, the syntactic signatures that contribute to that relationship could be condition-specific. That is, the syntactic EEG signals driven by *Jabberwocky* could be fundamentally different from those in the *Sentence* condition.

To test if the syntactic signatures in the *Sentence* and *Jabberwocky* conditions are exchangeable (i.e. similar in some way), we examined the accuracy of the decoding model in two cases: 1) using the EEG signals from the *Sentence* condition to predict the $g(S)$ vectors from the *Jabberwocky* stimuli, and 2) using the EEG signals from the *Jabberwocky* condition to predict the $g(S)$ vectors from the *Sentences* stimuli (see Table 1, Analysis 2). Because the *Jabberwocky* LSTM representations do not contain semantic information, this analysis will also tell us the degree to which the *Sentences* EEG/LSTM results in Analysis 1 leveraged semantic information. Because it is so central to this analysis, we again note that **the *Jabberwocky* stimuli are composed of pseudo-words derived from the *Sentence* stimuli, and the word order is maintained.** That is, the first word of sentence 1 in the *Jabberwocky* condition is a pseudo-word transformation of the first word from sentence 1 of the *Sentence* condition. Thus, we can interchange the corresponding representational vectors.

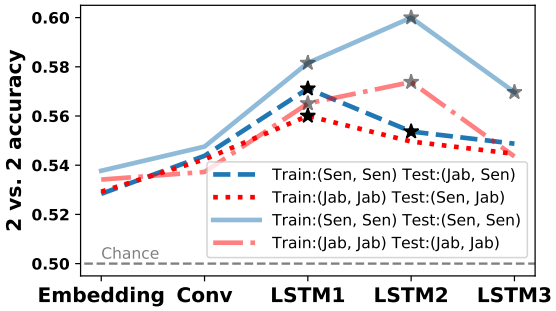
In Fig. 4a we see that the EEG signals from the *Sentence* condition can be used to predict the *Jabberwocky* LSTM representations (case 1). The accuracies for LSTM1-3 are 0.573, 0.578, and 0.560 which are all above chance ($p < 0.05$, FDR corrected). For the most part, the accuracies for case 1 are lower than the results from case 1 in Analysis 1 (EEG/LSTM representations from the *Sentence* condition), and we find there is a significant difference in the performance of LSTM2 ($p = 0.0006$). This is consistent with the hypothesis that *Jabberwocky* LSTM representations contain only syntactic information. Interestingly, the 2 vs. 2 accuracy when using *Sentence* EEG and *Jabberwocky* LSTM representations is higher than Analysis 1, where *Jabberwocky* EEG was paired with *Jabberwocky* LSTM representations. This is evidence that the syntactic information encoded in the *Sentence* EEG signals may be less noisy.

In case 2, when we use the *Jabberwocky* EEG to predict the *Sentence* LSTM representations, only the first LSTM layer shows above chance accuracy (0.568, $p < 0.05$, FDR corrected). This implies that the EEG signals from the *Jabberwocky* condition are not significantly correlated with the syntactic information in LSTM2 and LSTM3 vectors derived from *Sentence* stimuli. However, LSTM1 seems to encode syntactic information that is exchangeable.

Though we did not explicitly test the correlation of the LSTM vectors for the *Sentence* and *Jabberwocky* conditions, Analysis 2 provides evidence that the two may encode correlated syntactic information. In addition, recall that the LSTM fed *Jabberwocky* can predict the next word of the corresponding *Sentence* stimuli with perplexity close to that of an LSTM fed *Sentence* stimuli. That predictability is another piece of evidence that the representations share information that could be lever-



(a) Analysis 2 (Swap $g(S)$ vectors): Solid lines show the 2 vs. 2 accuracy of the decoding model that uses the *Sentences* EEG signals to predict the $g(S)$ vectors from the *Jabberwocky* stimuli, and vice versa.



(b) Analysis 3 (Swap R at test time): Solid lines show the 2 vs. 2 accuracy of the decoding model trained with EEG data and LSTM representations from the same condition, but tested with EEG data from the other condition.

Figure 4: Results from Analysis 2 and 3. Analysis 1 results appear as dashed lines. The x-axis denotes LSTM representation ($g(S)$). Legend denotes EEG/LSTM representations used for train/test: (EEG condition, LSTM condition). "*Sen*": *Sentence*, "*Jab*": *Jabberwocky* , "*WL*": *Word-list*. $\star$: above chance ($p < 0.05$, FDR corrected).

aged in across the two decoding tasks.

3.4 Swap R at *test* time only (Analysis 3)

This analysis tests if a trained decoding model can generalize to EEG data from the other condition. For example, can a model trained with EEG signals and LSTM representations both from the *Sentence* condition still predict the *Sentence* vectors when tested on EEG from the *Jabberwocky* condition? This is Analysis 3 in Table 1. If the *pattern* leveraged to predict LSTM representations is similar across the two conditions, the 2 vs. 2 accuracy will remain above chance.

In Fig. 4b, for case 1 (train on *Sentence* EEG, test on *Jabberwocky* EEG), the accuracies of the LSTM1 (0.571) and LSTM2 (0.553) are statistically above chance ($p < 0.05$, FDR corrected). Thus, the model trained using *Sentence* EEG can predict *Sentence* vectors from the corresponding *Jabberwocky* EEG. This implies that the brain's representation for the syntax in both the *Sentence* and *Jabberwocky* conditions takes a similar form, at least with respect to the syntactic information represented in LSTM1 and LSTM2. However, the performance of LSTM2 here is significantly lower than the performance of LSTM2 in case 1 of Analyses 1 and 2 ($p = 0.0001, p = 0.0005$ respectively). In fact, the performance for LSTM2 has dropped by a very large margin compared to Analysis 1, presumably because the semantic information leveraged in Analysis 1 is not available in the *Jabberwocky* EEG.

For case 2, (trained on *Jabberwocky* EEG/LSTM representations, but tested on *Sentence* EEG), only LSTM1 can be predicted with above chance 2 vs. 2 accuracy (0.560 with $p = 0.001$). So, as we saw in case 1, the LSTM1 model does generalize to EEG from the other condition. But, the same cannot be said for LSTM2, which is not significantly above chance in this case. That LSTM2 generalizes in one direction (case 1) but not the other (case 2) implies that the *Jabberwocky* EEG data is noisier, leading to a less robust model.

4 Discussion

Considering the results as a whole, several points become clear. There is a relationship between the semantic and/or syntactic information as represented by the brain and by LSTM representations, at least for the *Sentence* and *Jabberwocky* conditions. The probing results are quite consistent with the results of Analyses 1-3: LSTM1 has a strong signal for syntax, LSTM2 has syntax and semantics, and LSTM3 has some syntax and/or semantic signal, but the signal is weaker than for LSTM1-2.

LSTM1 shows only minor changes in performance in Analysis 2 and 3. So the syntactic information encoded in this layer is fairly consistent for stimuli from both the *Sentence* and *Jabberwocky* conditions, and it correlates well to either EEG data source. There is likely not much semantic information to leverage here, as the performance of models trained on *Sentence* EEG change by only a small amount in Analysis 2 and 3.

In Analysis 2 we saw similar drops in LSTM2 performance for both *Sentence* and *Jabberwocky* conditions. The drop in performance using the *Sentence* EEG could be attributed to the lack of

semantic information in the *Jabberwocky* LSTM representations. However, we see a similar size drop in performance for the *Jabberwocky* condition, which implies that there is a mismatch even in the syntactic information available in LSTM2 for the two conditions. In Analysis 3, when we swap the test data, the pattern learned to predict LSTM2 in the *Sentence* condition (leveraging semantics and syntax) is not as effective when tested on *Jabberwocky* data.

The performance of LSTM3 is harder to explain, possibly because it has weaker semantic/syntactic signal (as evidenced by the probing tasks). There is a small performance hit when training on *Sentence* EEG data in Analysis 2, but a very large drop in Analysis 3. This pattern could result if LSTM3's representations of syntax are similar for *Sentence* and *Jabberwocky* stimuli, but the brain showed differing representations for the syntactic information in the two conditions. Then, it is possible that only the *Sentence* EEG would correlate to the syntactic information in LSTM3.

We wondered if there could be another explanation for our ability to decode in the *Jabberwocky* condition. One possibility is that the EEG and LSTM layers contain a correlate of the position in a sentence (1st word, 2nd word, etc.), and our models are using that information to decode (7/8 2 vs. 2 tests will use words at different positions). To test for this possibility, we trained a classifier to predict the ordering of two random words selected from a sentence, as suggested by Adi et al. (2016). The input to the classifier is the LSTM representation of the two words at their positions in a sentence, and the output is a binary decision for which of the two words appears sooner in the sentence. A model trained using our LSTM and the *Sentence* stimuli produced 80% accuracy on this task. Thus, we cannot say unequivocally that our results are not due in some part to positional information. However, our probing results are consistent with there being semantic/syntactic information in the representations, and those results are very consistent with the decoding analysis. This is strong evidence that our results are not entirely due to positional information.

We wondered also if the lexical semantics of the *Jabberwocky* stimuli could be leaking into the LSTM vectors, perhaps because the pseudo-words were repaired in the convolution step of the LSTM. Note, however, that lexical semantics are entirely intact in the *Word-list* condition, but the LSTM representations are of no use in that condition. Morphosyntax and syntax are maintained in the *Jabberwocky* condition, which appears to be enough to drive the correlation between LSTM representations and EEG recordings. The LSTM may be picking up on bi- and tri-gram signals related to morphosyntax cueing syntactic structure (Martin, 2016, 2020), but more work is needed to rule out alternative explanations.

Recall that the *Sentence* and *Jabberwocky* stimuli share some orthographic/phonological information. Could our *Jabberwocky* results, and the results of Analysis 2 (swap $g(s)$), be due only to the EEG encoding phonological or orthographic information? If our models were able to leverage such information, we would expect to see comparable decoding results in Analysis 1 and the *Word-list* condition, where the stimuli are perfect orthographic matches to the EEG. However, that analysis did not produce significantly above-chance accuracy. Furthermore, if the information leveraged in Analysis 2 was at the character-level, we would expect to see significantly above-chance accuracy in the character embedding or convolutional layers. However, it is not until the first LSTM layer (where contextual information is first incorporated) that any decoding model performs significantly above chance in any condition. This is evidence that the information being leveraged is *not simply phonological or orthographic*.

Our stimuli are composed of two conjoined sentences. How much composition have Dutch listeners done by the time when they get to the conjunction word "en?" How does the processing differ between the first vs the second of the conjoined sentences? Previous work on the brain's processing of syntactic structures and coordinate clauses proposed an "active structure maintenance model", where neural activity increases as a function of syntactic complexity Pallier et al. (2011); Lau and Liao (2018). They found that neural activity in certain left-hemispheric regions indeed increased when more constituents had to be integrated, for both sentences and jabberwocky stimuli. It may be that the second coordinate constituent in our stimuli sentences elicit stronger neural activity than the first, but more analysis would be required to verify this.

5 Related Work

The first example of mapping brain responses onto corpus-derived representations appeared in Mitchell et al. (2008). This study encoded word meaning into vectors of word co-occurrence features. The authors showed that a trained linear regression model could predict fMRI activation in response to single concrete noun stimuli. From there, decoding models were shown to work with dependency-parse-based representations (Murphy et al., 2012) and with concept-relation-features extracted from topic models (Pereira et al., 2013). Anderson et al. (2017) demonstrated that decoding models can learn the pattern of the brain's response to abstract concepts/nouns.

Some of the first examples of decoding language *in context* were from Wehbe et al. (2014a) and de Heer et al. (2017). The first models used a combination of (non-contextual) corpus-derived, acoustically-derived and/or hand-coded representations. Several groups then began to experiment with encoding models based on *contextual* language representations, like those in recurrent neural network (RNN) language models (Wehbe et al., 2014b; Jain and Huth, 2018; Toneva and Wehbe, 2019). These models showed that vectors incorporating contextual information could be decoded from brain imaging data, and contextual models actually outperformed non-contextual word vectors. We confirmed those findings here.

Though there are fewer decoding models trained on EEG, there are a few recent examples. Hale et al. (2018) showed that the operations performed by an RNN-grammar trained to parse sentences correlated to EEG collected while people listened to a story. Schwartz and Mitchell (2019) found connections between bi-LSTM representations and the ERPs (event related potentials) more classically used to study language in the brain. Our work adds to the new body of work showing that EEG can be a powerful data source in this space.

6 Conclusion and future work

In this study, we explored the correlation of a character-level LSTM with the brain's response for two kinds of out-of-distribution language. The *Jabberwocky* condition used pseudo-word translations of the *Sentence* stimuli (ablate semantics, preserve syntax). The *Word-list* stimuli was a pseudo-random re-ordering of the words in each of the *Sentence* stimuli (ablate syntax, preserve semantics). We ran a character-based LSTM to create contextual embeddings for the stimuli of each condition. Our linear-regression decoding models were trained to predict the various LSTM representations from the EEG signals.

Our results showed that the LSTM layers of this character-based LSTM do in fact correlate with EEG signals in both the *Sentence* and *Jabberwocky* conditions, but not in the *Word-list* condition. By training models with various alterations to the data, we were able to determine which LSTM representations carry semantic and syntactic information. We verified those results using a probing task on our Dutch LSTM, as well as an identical model trained on English.

There are multiple avenues for future work. For example, Dutch has a fairly transparent phoneme-grapheme correspondence; would our results still hold for a language with deeper orthography? We were surprised to find that some LSTM representations resembled the *Jabberwocky* EEG signals. Are there other examples of out-of-distribution language where this relationship holds? And, perhaps more interestingly, where it does not hold? Finding ways in which the brain's representations differ from an LSTM could help us to build models closer to the true nature of human language processing.

Acknowledgments

AF and MW are supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants, and hold Canada CIFAR (Canadian Institute for Advanced Research) AI Chairs. The computational work was supported in part by infrastructure made available by WestGrid and Compute Canada. AEM was supported by the Max Planck Research Group "Language and Computation in Neural Systems" and by the Netherlands Organization for Scientific Research (grant 016.Vidi.188.029).

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Andrew J Anderson, Douwe Kiela, Stephen Clark, and Massimo Poesio. 2017. Visually grounded and textual semantic models differentially decode brain activity associated with concrete and abstract nouns. *Transactions of the Association for Computational Linguistics*, 5:17–30.

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

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Max Eichler, Gözde Gül Şahin, and Iryna Gurevych. 2019. LINSPECTOR WEB: A multilingual probing suite for word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 127–132, Hong Kong, China. Association for Computational Linguistics.

Evelina Fedorenko, Alfonso Nieto-Castañon, and Nancy Kanwisher. 2012. Lexical and syntactic representations in the brain: An fmri investigation with multi-voxel pattern analyses. *Neuropsychologia*, 50(4):499 – 513. Multivoxel pattern analysis and cognitive theories.

Angela D. Friederici, Martin Meyer, and D.Yves [von Cramon]. 2000. Auditory language comprehension: An event-related fmri study on the processing of syntactic and lexical information. *Brain and Language*, 74(2):289 – 300.

Alona Fyshe, Gustavo Sudre, Leila Wehbe, Nicole Rafidi, and Tom M. Mitchell. 2019. The lexical semantics of adjective–noun phrases in the human brain. *Human Brain Mapping*, 40(15):4457–4469.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan R Brennan. 2018. Finding syntax in human encephalography with beam search. *arXiv preprint arXiv:1806.04127*.

Wendy A de Heer, Alexander G Huth, Thomas L Griffiths, Jack L Gallant, and Frédéric E Theunissen. 2017. The hierarchical cortical organization of human speech processing. *Journal of Neuroscience*, 37(27):6539–6557.

Colin Humphries, Jeffrey R. Binder, David A. Medler, and Einat Liebenthal. 2006. Syntactic and semantic modulation of neural activity during auditory sentence comprehension. *Journal of Cognitive Neuroscience*, 18(4):665–679.

Shailee Jain and Alexander Huth. 2018. Incorporating context into language encoding models for fmri. In *Advances in Neural Information Processing Systems*, pages 6628–6637.

Greta Kaufeld, Hans Rutger Bosker, Sanne Ten Oever, Phillip M. Alday, Antje S. Meyer, and Andrea E. Martin. 2020. Linguistic structure and meaning organize neural oscillations into a content-specific hierarchy. *Accepted at Journal of Neuroscience*.

Emmanuel Keuleers and Marc Brysbaert. 2010. Wuggy: A multilingual pseudoword generator. *Behavior research methods*, 42(3):627–633.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Ellen Lau and Chia-Hsuan Liao. 2018. Linguistic structure across time: Erp responses to coordinated and uncoordinated noun phrases. *Language, Cognition and Neuroscience*, 33(5):633–647.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Andrea E. Martin. 2016. Language processing as cue integration: Grounding the psychology of language in perception and neurophysiology. *Frontiers in Psychology*, 7:120.

Andrea E. Martin. 2020. A compositional neural architecture for language. *Journal of Cognitive Neuroscience*, pages 1–20.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Tom M Mitchell, Svetlana V Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L Malave, Robert A Mason, and Marcel Adam Just. 2008. Predicting human brain activity associated with the meanings of nouns. *science*, 320(5880):1191–1195.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Selecting corpus-semantic models for neurolinguistic decoding. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 114–123. Association for Computational Linguistics.

Christophe Pallier, Anne-Dominique Devauchelle, and Stanislas Dehaene. 2011. Cortical representation of the constituent structure of sentences. *Proceedings of the National Academy of Sciences*, 108(6):2522–2527.

Francisco Pereira, Matthew Botvinick, and Greg Detre. 2013. Using wikipedia to learn semantic feature representations of concrete concepts in neuroimaging experiments. *Artificial intelligence*, 194:240–252.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Dan Schwartz and Tom Mitchell. 2019. Understanding language-elicited eeg data by predicting it from a fine-tuned language model. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 43–57.

Mariya Toneva and Leila Wehbe. 2019. Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain). *arXiv preprint arXiv:1905.11833*.

Leila Wehbe, Brian Murphy, Partha Talukdar, Alona Fyshe, Aaditya Ramdas, and Tom Mitchell. 2014a. Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses. *PloS one*, 9(11).

Leila Wehbe, Ashish Vaswani, Kevin Knight, and Tom Mitchell. 2014b. 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)*, pages 233–243.

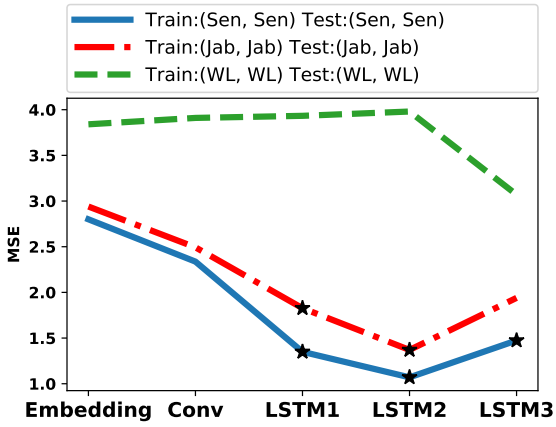


Figure 5: Analysis 1 (Test for semantic and syntactic information): MSE for $g(S)$/EEG from the same condition. The x-axis denotes LSTM representation ($g(S)$). Legend denotes EEG/LSTM representations used for train/test: (EEG condition, LSTM condition). "*Sen*": *Sentence*, "*Jab*": *Jabberwocky* , "*WL*": *Word-list*. $\star$: below chance ($p < 0.05$, FDR corrected).

A Supplemental Material: Probing task performance

Table 2 describes the probing tasks in English from Conneau et al. (2018) and in Dutch from Eichler et al. (2019). Table 3 shows probing task accuracy for both English and Dutch datasets, as measured with the character-based LSTMs proposed by Kim et al. (2016). The English model is trained on the Penn Treebank (Marcus et al., 1993), the Dutch on Dutch Wikipedia.

B Supplemental Material: Measuring model accuracy by mean-squared-error

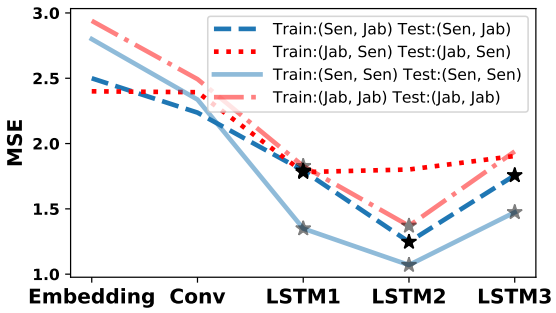
In addition to 2 vs. 2 accuracy, we also used mean-squared-error (MSE) to assess the performance of the decoding model. Figs. 5 and 6 show the results of MSE for analyses 1-3.

Table 2: Description of the probing tasks. "En" shows the English datasets and "Du" shows the Dutch datasets.

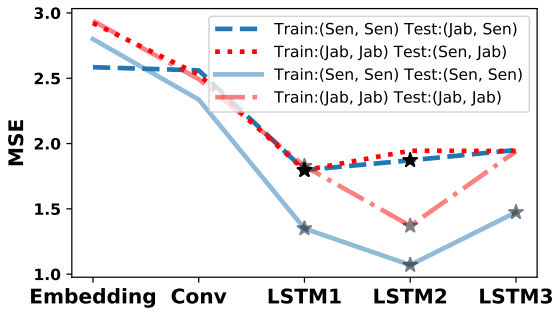
Type	Name	Description	Data
Semantic	Tense	*Tense of the main-clause verb (present/ past)*	*En/Du*
	Subject number	*Number of the subjects of the main clause*	*En*
	Object number	*Number of the direct objects of the main clause*	*En*
	Coordination inversion	*Indicate if a sentence is intact or modified*	*En*
Syntactic	Bigram shift	*Indicate having legal word orders*	*En*
	Tree depth	*Depth of the hierarchical structure of sentences*	*En*
	Top constituent	*Indicate top constituent sequence of sentences*	*En*
	Number	*Indicate singularity and plurality of nouns/adjectives/verbs*	*Du*
	Part of Speech	*Indicate the part of speech of a specific word*	*Du*

Table 3: Probing task accuracies. Each row shows the accuracies of a specific probing task described in Table 2. Columns correspond to the LSTM representation: "*Embedding*": Embedding layer, "*Conv*": concatenation of Convolutional layers, "*LSTM1-3*": an LSTM layers. "*Tense/En*" and "*Tense/Du*" denote the English and Dutch probing task for *Tense*, respectively.

Layers #	Embedding	Conv	LSTM1	LSTM2	LSTM3
Tense/En	43.2	53.2	63.2	70.7	63.9
Tense/Du	46.4	55.1	66.7	72.7	62.7
Subject number	38.8	53.5	65.5	72.1	64.3
Object number	39.5	52.1	66.8	71.7	65.8
Coord. Inv.	40.5	46.6	58.7	66.1	61.3
Bigram shift	43.1	53.1	70.8	69.4	58
Tree depth	39.3	45.6	56.3	58.6	54.3
Top constituent	38.5	53.8	75.5	76.1	64.1
Number	52.3	58.6	78.2	81.9	67.3
Part of Speech	39.6	53.7	69.8	76.1	60.3



(a) Analysis 2 (Swap $g(S)$ vectors): Solid lines show the MSE of the decoding model that uses the *Sentences* EEG signals to predict the $g(S)$ vectors from the *Jabberwocky* stimuli, and vice versa.



(b) Analysis 3 (Swap R at test time): Solid lines show the MSE of the decoding model trained with EEG data and LSTM representations from the same condition, but tested with EEG data from the other condition.

Figure 6: MSE results from Analysis 2 and 3. Analysis 1 results appear as dashed lines. The x-axis denotes LSTM representation ($g(S)$). Legend denotes EEG/LSTM representations used for train/test: (EEG condition, LSTM condition). "*Sen*": *Sentence*, "*Jab*": *Jabberwocky* , "*WL*": *Word-list*. $\star$: below chance ($p < 0.05$, FDR corrected).