Using Priming to Uncover the Organization of Syntactic Representations in Neural Language Models

Grusha Prasad
Johns Hopkins University
grusha.prasad@jhu.edu

Marten van Schijndel
Cornell University
mv443@cornell.edu

Tal Linzen
Johns Hopkins University
tal.linzen@jhu.edu

Abstract

Neural language models (LMs) perform well on tasks that require sensitivity to syntactic structure. Drawing on the syntactic priming paradigm from psycholinguistics, we propose a novel technique to analyze the representations that enable such success. By establishing a gradient similarity metric between structures, this technique allows us to reconstruct the organization of the LMs' syntactic representational space. We use this technique to demonstrate that LSTM LMs' representations of different types of sentences with relative clauses are organized hierarchically in a linguistically interpretable manner, suggesting that the LMs track abstract properties of the sentence.

1 Introduction

Neural networks trained on text alone, without explicit syntactic supervision, have been surprisingly successful in tasks that require sensitivity to sentence structure. The difficulty of interpreting the learned neural representations that underlie this success has motivated a range of analysis techniques, including diagnostic classifiers (Giulianelli et al., 2018; Conneau et al., 2018; Shi et al., 2016), visualization of individual neuron activations (Kádár et al., 2017; Qian et al., 2016), ablation of individual neurons or sets of neurons (Lakretz et al., 2019) and behavioral tests of generalization to infrequent or held out syntactic structures (Linzen et al., 2016; Weber et al., 2018; McCoy et al., 2018); for reviews, see Belinkov and Glass (2019) and Alishahi et al. (2019).

This paper expands the toolkit of neural network analysis techniques by drawing on the **syntactic priming** paradigm, a central tool in psycholinguistics for analyzing human syntactic representations (Bock, 1986). This paradigm is based on the empirical finding that people tend to reuse syntactic structures that they have recently produced or encountered. For example, English provides two roughly equivalent ways to express a transfer event:

(1) a. The boy threw the ball to the dog.
     b. The boy threw the dog the ball.

When readers encounter one of these variants in the text more frequently than the other, they expect that future transfer events will more likely be expressed using the frequent construction than the infrequent one. For example, after reading sentences like (1a) (the **prime**), readers expect sentences like (2a), which shares syntactic structure with the prime, to occur with a greater likelihood than the alternative variant like (2b) which does not (Wells et al., 2009).[1]

(2) a. The lawyer sent the letter to the client.
     b. The lawyer sent the client the letter.

We use the priming paradigm to analyze neural network language models (LMs), systems that define a probability distribution over the nth word of a sentence given its first $n-1$ words. Building on paradigms that determine whether the LM's expectations are consistent with the syntactic structure of the sentence (Linzen et al., 2016), we measure the extent to which a LM's expectation for a specific syntactic structure is affected by recent experience with related structures. We prime a fully trained model with a structure by adapting it to a small number of sentences containing that structure (van Schijndel and Linzen, 2018). We then measure the change in surprisal (negative log probability) after adaptation when the LM is tested either on sentences with the same struc-

---

[1] Wells et al. (2009) measured priming effects for relative clauses, not dative constructions. For work on priming in production with dative constructions, see Kaschak et al. (2011).

ture or sentences with different but related structures. The degree to which one structure primes another provides a graded similarity metric between the model's representations of those structures (cf. Branigan and Pickering 2017), which allows us to investigate how the representations of sentences with these structures are organized.

As a case study, we applied this technique to investigate how recurrent neural network (RNN) LMs represent sentences with relative clauses (RCs). We found that the representations of these sentences are organized in a linguistically interpretable manner: sentences with a particular type of RC were most similar to other sentences with the same type of RC in the LMs' representation space. Furthermore, sentences with different types of RCs were more similar to each other than sentences without RCs. We demonstrate that the similarity between sentences was not driven merely by specific words that appeared in the sentence, suggesting that the LMs tracked abstract properties of the sentence. This ability to track abstract properties decreased as the training corpus size increased. Finally, we tested the hypothesis that LMs' accuracy on agreement prediction (Marvin and Linzen, 2018) would increase with the LMs' ability to track more abstract properties of the sentence, but did not find evidence for this hypothesis.

2 Background

2.1 Syntactic predictions in neural LMs

We build on paradigms that use LM probability estimates for words in a given context as a measure of the model's sensitivity to the syntactic structure of the sentence (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018). If a language model assigns a higher probability to a verb form that agrees in number with the subject (*the boy... writes*) than a verb form that does not (*the boy... write*), we can infer that the model encodes information about the agreement features of nouns and verbs (that is, the difference between singular and plural) and has correctly identified the subject that corresponds to this verb. This reasoning has been extended beyond subject-verb agreement to study whether the predictions of neural LMs are sensitive to a range of other syntactic dependencies, including negative polarity items (Jumelet and Hupkes, 2018), filler-gap dependencies (Wilcox et al., 2018) and reflexive pronoun binding (Futrell et al., 2019).

2.2 Syntactic priming in humans

Syntactic priming has been used to study whether the representations of two sentences have shared structure. For example, (1a) (repeated below as (3)) shares the structure VP → V NP PP with (4a) but not (4b).

(3) The boy threw the ball to the dog.

(4) a. The renowned chef made some wonderful pasta for the guest.
     b. The renowned chef made the guest some wonderful pasta.

If (3) primes (4a) more than it primes (4b), we can infer that the representations of (3) are more similar to that of (4a) than to that of (4b). Since (4b) and (4a) differ only in their structure, this difference in similarity must be driven by structural information in the representations of the sentences (for reviews, see Mahowald et al. 2016 and Tooley and Traxler 2010).

Although priming studies have traditionally measured the priming effect on the sentence immediately following the prime, more recent studies have demonstrated that the effects of syntactic priming can be cumulative and long-lasting: sentences with a shared structure S_X become progressively easier to process when preceded by n sentences with the same structure S_X than when preceded by n sentences with a different structure S_Y (Kaschak et al., 2011; Wells et al., 2009).[2] In conjunction with the finding that words that are consistent with a probable syntactic parse are easier to process than words consistent with less probable parses (Hale, 2001; Levy, 2008), the increased ease of processing in cumulative priming studies can be interpreted as evidence that, with increased exposure to a structure, participants begin to expect that structure with a greater probability (Chang et al., 2006).

Cumulative priming allows us to study how sentences are related to each other in the human (or LM) representation space in the same way that non-cumulative priming does: when participants (or LMs) are exposed to sentences with structure S_X, if there is a greater decrease in surprisal when they are tested on other sentences with S_X than when they are tested on other sentences with S_Y, we can infer that the representations of sentences with S_X are more similar to each other than to the

---

[2]In studies looking at non-cumulative priming, $n = 1$.

Abstract structure	Example
Unreduced Object RC	The conspiracy that the employee welcomed divided the beautiful country.
Reduced Object RC	The conspiracy the employee welcomed divided the beautiful country.
Unreduced Passive RC	The conspiracy that was welcomed by the employee divided the beautiful country.
Reduced Passive RC	The conspiracy welcomed by the employee divided the beautiful country.
Active Subject RC	The employee that welcomed the conspiracy quickly searched the buildings.
PS/ORC-matched Coordination	The conspiracy welcomed the employee and divided the beautiful country.
ASRC-matched Coordination	The employee welcomed the conspiracy and quickly searched the buildings.

Table 1: Examples of sentences generated using templates containing the seven abstract structures we analyzed (optional elements, which only occur in a subset of the examples, are indicated in grey).

representations of sentences with S_Y.

2.3 LM adaptation as cumulative priming

Van Schijndel and Linzen (2018) modeled cumulative priming in recurrent neural networks (RNNs) by adapting fully trained RNN LMs to new stimuli — i.e. taking a fully trained RNN LM and continuing to train it on a small set of sentences (cf. Grave et al. 2017; Krause et al. 2017; Chowdhury and Zamparelli 2019). They demonstrated that when an RNN LM was adapted to a small number of sentences with a shared syntactic structure, the surprisal for novel sentences with that structure decreased, enabling them to infer that the LM's representations of sentences contained information about that structure.

3 Similarity between syntactic structures in RNN LM representational space

Following the assumptions in Section 2.2, we define a similarity metric between two structures S_X and S_Y in an LM's representation space by adapting the LM to sentences with S_X and measuring the change in surprisal for sentences with S_Y — i.e. measuring to what extent sentences with S_X prime sentences with S_Y. We use the notation $\mathbb{A}(Y \mid X)$ to refer to this change in surprisal[3], where X and Y are non-lexically-overlapping sets of sentences whose members share the structures S_X and S_Y respectively. If we assume that S_X and S_Y are similar to each other in the LM's representation space, then $\mathbb{A}(Y \mid X) > 0$ — i.e., encountering sentences with S_X causes the LM to assign a higher probability to sentences with S_Y. On the other hand, if we assume that S_X and S_Y are unrelated to each other, then $\mathbb{A}(Y \mid X) = 0$ — i.e., encountering sentences with S_X does not cause the LM to change its probability for sentences with

---

[3] $\mathbb{A}$ is shorthand for adaptation.

S_Y.

4 Experimental setup

4.1 Syntactic structures

We analyzed five types of RCs. In an active subject RC, the gap is in the subject position of the embedded clause:[4]

(5) My cousin that _ liked the book ...

In a passive subject RC (*passive RCs*), the gap is in the subject position of the embedded clause, and the embedded verb is passive. In English, passive RCs can be unreduced (6a) or reduced (6b):

(6) a. The book that _ was liked by my cousin ...
     b. The book _ liked by my cousin ...

In an object RC the gap is in the object position of the embedded clause. In English, object RCs can be unreduced (7a) or reduced (7b):

(7) a. The book that my cousin liked _ ...
     b. The book my cousin liked _ ...

Finally, we also included two additional conditions with verb coordination: one with nearly identical word order and lexical content as active subject RCs ((8); ASRC-matched Coordination), and another with nearly identical word order and lexical content as passive RCs and object RCs ((9); PS/ORC-matched Coordination).[5]

(8) My cousin liked the book and ...

(9) The book liked my cousin and ...

---

[4] We illustrate the location of the gap with underscores here, but the underscores were not included in the LM's input.

[5] In order to maintain the same word order as in object and passive RCs, the subject of the coordinated verb phrases is an NP that tends to fill the object position in other sentences (e.g, "the equation"). Therefore, many of the sentences in this condition are implausible (e.g., "The equation reviewed the physicists and challenged the method.")
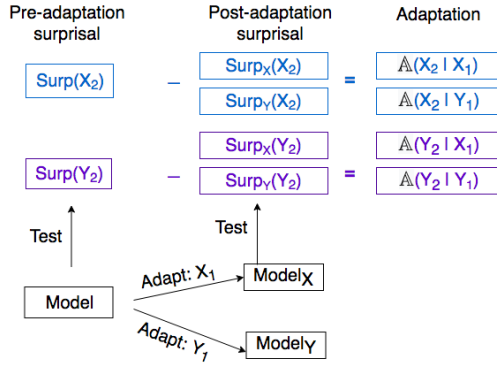
Figure 1: A schematic for calculating the similarity between two structures S_X and S_Y in an LM's representation space. X_1, X_2 and Y_1, Y_2 are non-lexically-overlapping sets of sentences with S_X and S_Y respectively. $Model_X$ and $Model_Y$ refer to versions of a fully trained model that have been adapted to either X_1 or Y_1 respectively. $Surp_X()$ and $Surp_Y()$ are functions that return the surprisal of sentences for $Model_X$ and $Model_Y$.

These conditions enable us to measure whether sentences with different types of RCs are more similar to each other in an LM's representation space than they are to lexically matched sentences without RCs.

4.2 Adaptation and test sets

We generated sentences from seven templates, one for each of the syntactic structures of interest. The slots were filled with 223 verbs, 164 nouns, 24 adverbs and 78 adjectives such that the semantic plausibility of the combination of nouns, verbs, adverbs and adjectives was ensured. The seven variants of every sentence had nearly identical lexical items (see Table 1).[6] We used these templates to generate five experimental lists — each list comprised of a pair of adaptation and test sets with minimal lexical overlap between them (only function words and some modifiers were shared). Each adaptation set contained 20 sentences and each test set contained 50.

In order to infer that any decrease in surprisal is caused by adaptation to an abstract syntactic structure, we need to ensure that the models are not adapting to properties of the sentence that are unrelated to the abstract structure of interest. Con-

sider a LM adapted to (10) and tested on (11):

(10) The conspiracy that the employee welcomed divided the country.

(11) The proposal that the receptionist managed shocked the CEO.

When the LM is adapted to sentences such as (10), it could adjust its expectations about several properties of the sentence, some more linguistically interesting than others. For instance, it could learn that there are three determiners in the sentence, that the third word of the sentence is *that*, that sentences have nine words, that every verb is preceded by a noun, and so on and so forth. If there is a decrease in surprisal when a model is adapted to (10) and tested on (11), it is unclear if this is because the model learned to expect object relative clauses or if it learned to expect any of the other mentioned properties.

To minimize the likelihood that the adaptation effects are driven by irrelevant properties of the sentence, we introduced several sources of variability to our templates: nouns could either be singular or plural, noun phrases could be optionally modified by an adjective, adjectives were optionally modified with an intensifier and verb phrases were optionally modified with adverbs which could occur either pre-verbally or post-verbally (details in the Supplementary Materials).[7]

4.3 Models

We used 75 of the LSTM language models trained by van Schijndel et al. (2019); these LMs varied in the number of hidden units per layer (100, 200, 400, 800, 1600) and the number of tokens they were trained on (2 million, 10 million or 20 million). For each training corpus size, van Schijndel and Linzen trained models on five disjoint subsets of the WikiText-103 corpus, to ensure that the results generalized across different training sets.

4.4 Calculating the adaptation effect (AE)

For every structure, we computed the similarity between that structure and every other structure (including itself) as described in Section 3. This process is schematized in Figure 1. The surprisal values were averaged across the entire sentence.[8]

---

[6]Since the main verb of the sentence was constrained to be semantically plausible with the subject of the sentence, it often varied between active subject RC and ASRC-matched coordination on the one had and all other conditions on the other.

[7]The Supplementary Materials, the templates and and code for all the analyses along with the data can be found on GitHub: https://github.com/grushaprasad/RNN-Priming

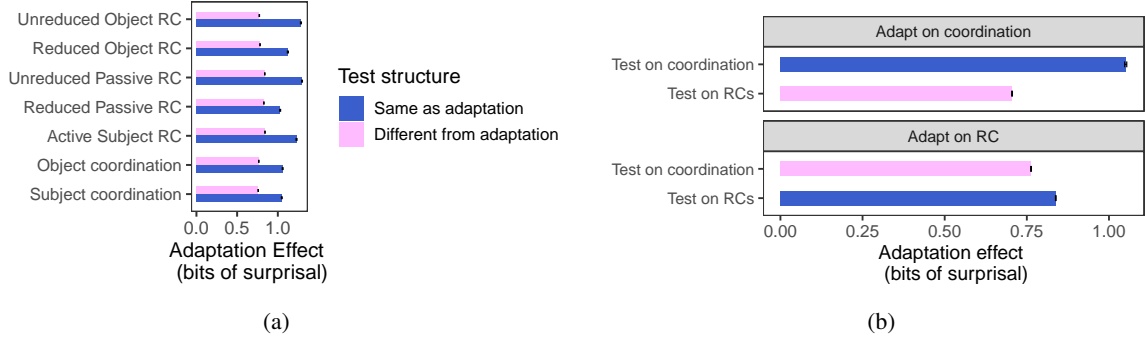[8]Unknown words were excluded from this average.

Figure 2: The adaptation effect averaged across all 75 models when (a) they were adapted to each of the structures and tested on either the same structure (blue, bottom) or different structure (pink, top) and (b) they were adapted to RCs and tested on non-RCs or vice versa (pink bars); or when they were adapted to RCs or non-RCs and tested on other RCs or and non-RCs respectively (blue bars). Greater values indicate more similarity between adaptation and test structures. Error bars reflect 95% CIs.

We found that $\mathbb{A}(B \mid A)$ was proportional to the surprisal of B prior to adaptation (see Supplementary Materials). As a consequence, for three structures X, Y and Z, $\mathbb{A}(Y \mid X)$ could be greater than $\mathbb{A}(Z \mid X)$ merely because Y was a more surprising structure to begin with than Z. In order to remove this confound, we first fit a linear regression model predicting $\mathbb{A}(Y \mid X)$ from the surprisal of Y prior to adaptation ($Surp(Y)$):

$$\mathbb{A}(Y \mid X) = \beta_0 + \beta_1 Surp(Y) + \epsilon$$

We then regressed out the linear relationship between $\mathbb{A}(Y \mid X)$ and $Surp(Y)$ as follows:

$$AE(Y \mid X) = \mathbb{A}(Y \mid X) - \beta_1 Surp(Y)$$
$$= \beta_0 + \epsilon$$

Since $Surp(Y)$ was centered around its mean, β_0 reflects the mean of $\mathbb{A}(Y \mid X)$ when $Surp(Y)$ is equal to the mean surprisal of all sentences prior to adaptation. The term ϵ reflects any variance in $\mathbb{A}(Y \mid X)$ that is not predicted by $Surp(Y)$. By summing these two terms together, $AE(Y \mid X)$ reflects the change in surprisal for Y after adapting to X that is independent of $Surp(Y)$.

4.5 Statistical analyses

We used linear mixed effects models (Pinheiro et al., 2000) to test for statistical significance; all of the results reported below were highly significant. Details about the statistical analyses can be found in the Supplementary Materials.

5 Results

5.1 Validating AE as a similarity metric

As discussed in Section 2.3, under the adaptation-as-priming paradigm, we would expect sentences

that share the same specific structure to be more similar to each other than lexically matched sentences that do not share the structure.[9] In other words, if X_1 and X_2 are non-lexically-overlapping sets of sentences with shared structure S_X, and Y_2 is a set of sentences with structure S_Y, but is lexically matched with X_2, then we would expect $AE(X_2 \mid X_1) > AE(Y_2 \mid X_1)$. We found this prediction to be true for all of our seven structures (Figure 2a), thus validating our similarity metric.

5.2 Similarity between sentences with different types of VP coordination

Our two coordination conditions were structurally identical to each other but varied in their semantic plausibility — the sentences in PS/ORC-matched coordination condition were often semantically implausible whereas sentences in ASRC-matched condition were always semantically plausible (see footnote 5). If sentences that were structurally similar were close together irrespective of semantic plausibility, then we expect sentences with coordination to be more similar to each other than lexically matched sentences with RCs. Consistent with this prediction, the adaptation effect for models adapted to one type of coordination was greater when the models were tested on sentences with the other type of coordination than when they were tested on sentences with RCs (top panel of Figure 2b).

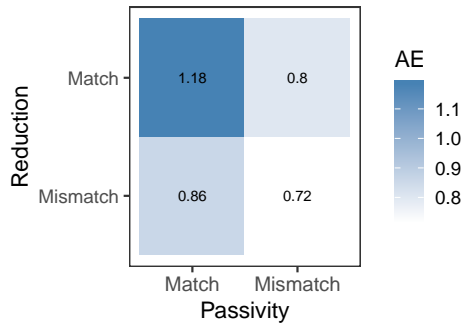[9]By lexically matched we mean that all content words were shared between sentences.

Figure 3: The adaptation effect when models adapted to sentences with reduced and unreduced RCs are tested on sentences that match only in reduction (top right), match only in passivity (bottom right), match in both reduction and passivity (top left) or sentences that match in neither (bottom right).

5.3 Similarity between sentences with different types of RCs

Unlike sentences with coordination, sentences with different types of RCs differ from each other at a surface level (see Table 1). However, at a more abstract level they all share a common property: a gap. If the RNN LMs were keeping track of whether or not a sentence contained a gap, we would expect sentences with different types of RCs to be more similar to each other in the RNN LMs' representation space than lexically matched sentences without a gap. In other words, if RC_X and RC_Y are two different types of RCs and $Coord_Y$ is a sentence with verb coordination lexically matched with RC_Y, then we would expect $AE(RC_Y \mid RC_X) > AE(Coord_Y \mid RC_X)$.

Consistent with this prediction, the adaptation effect for models adapted to RCs was greater when they were tested on sentences with other types of RCs than when they were tested on sentences with coordination (bottom panel of Figure 2b). This suggests that the LMs do keep track of whether or not a sentence contains a gap, even though this property is not overtly indicated by a lexical item that is shared across all types of RCs.

5.4 Similarity between sentences belonging to different sub-classes of RCs

The different types of RCs we tested can be divided into sub-classes based on at least two linguistically interpretable features: reduction and passivity. Reduction distinguishes reduced passive and object RCs on the one hand from unreduced passive and object RCs on the other. Passivity distinguishes reduced and unreduced passive RCs on the one hand from reduced and unreduced object RCs on the other. The LMs could be tracking either, both or none of these features.

We probed whether the LMs track these features by comparing the similarity between sentences that share one feature but not the other, with the similarity between sentences that share neither feature. If the adaptation effect is greater when there is a match in one feature than when there is a match in neither of the features, we can infer that the LMs track whether sentences have that feature. We found that the LMs track both of these features (Figure 3).

Additionally, we probed which of the features contributes more towards the similarity between sentences by comparing the similarity between sentences that match only in passivity with sentences that match only in reduction. When the adaptation and test sets matched only in passivity, the adaptation effect was slightly (but significantly) greater than when the adaptation and test sets matched only in reduction (Figure 3). In other words, in the LMs' representation space, (12) is more similar to (13) than it is to (14), suggesting that passivity contributes more towards the similarity between sentences than reduction.

(12) The conspiracy the employee welcomed divided the country.

(13) The conspiracy that the employee welcomed divided the country.

(14) The conspiracy welcomed by the employee divided the country.

This result is both intuitive and linguistically interpretable — the edit distance between reduced and unreduced RCs is smaller than the that between object and passive RCs; the syntax tree for (12) is also more similar to (13) than it is to (14).

5.5 What properties of sentences drive the similarity between them?

Our analyses so far have demonstrated that sentences that belong to linguistically interpretable classes (e.g., sentences that match in reduction) are more similar to each other in the LMs' representation space than they are to sentences that do not belong to those classes (e.g., sentences that do not match in reduction). However, it is unclear what properties of the sentences are driving this similarity between members of the class. For al-

most all of the linguistically interpretable classes we considered, all sentences belonging to a class shared at least some, if not all, function words. The only exception was the class of all RCs, where the property shared by all sentences in this class (the presence of a gap) was not overtly observable. Therefore, it is possible that the similarity between members of most of the classes we tested was being driven entirely by the presence of these function words.

In order to test whether the similarity between members of classes was indeed being driven by the presence of shared function words, we compared the representation space of the models we tested in the previous sections (henceforth *trained models*) with the representation space of models trained on no data (henceforth *baseline models*). Since the baseline models were only ever exposed to the 20 sentences in the adaptation set and there was no lexical overlap in content words between adaptation and test sets, any similarity between sentences in the representation space of these models would be driven by the presence of function words. If the similarity between sentences in the representation space of the trained models was being driven by factors other than the presence of function words, we would expect this similarity to be greater than the similarity between these sentences in the representation space of the baseline models.

We cannot directly use adaptation effect to compare the similarity between sentences in the representation spaces of trained models and baseline models, however: models trained on more data are likely to have stronger priors and are therefore less likely to drastically change their representations after 20 sentences than models trained on less data. In order to mitigate this issue, we defined a distance measure between sentences that belong to a class and sentences that do not belong to a class S_X as follows (see Figure 4 for a schematic):

$$\mathbb{D}(S_X, \neg S_X) = \frac{AE(X_2 \mid X_1)}{AE(\neg X_2 \mid X_1)}$$

This value would be greater than one if sentences that belonged to a class were more similar to each other than they were to sentences that did not belong to the class. Since the strength of prior belief would affect sentences that belong to the class the same way it would affect sentences that do not belong to the class, the effect would cancel out.

We measured the distance between members and non-members for three linguistically inter-

Figure 4: A schematic of how $\mathbb{D}(RC, \neg RC)$ is calculated. For any given row, the black square indicates the specific structure the models were adapted to, the blue squares indicate other structures that belong to the same linguistically defined class as the black square and the pink squares indicate the structures that do not belong to this linguistically defined class. In calculating the distance, we first calculated the proportion between the mean adaptation effect for the blue squares and the mean adaptation effect for pink squares for each row. We then averaged across the proportion for each row to arrive at one number.

pretable classes: sentences which contained the same type of RC, sentences that matched in their reduction or sentences that contained any type of RC. In our baseline models, for all three classes, sentences that belonged to one of these classes were more similar to each other than sentences that did not belong to that class (Figure 5a). This was surprising for the class of sentences that contained any type of RC because there was no function word that was shared by all sentences in this class. We hypothesize that this is because sentences without RCs always contained the word *and*, whereas sentences with RCs never did.

In cases where members of the class shared at least some function words, the distance between sentences that belonged to the class and sentences that did not for the trained models was greater than that for the baseline models. This suggests that the similarity between sentences in the representation space of trained models was being driven by factors other than the mere presence of function words. However, somewhat surprisingly, as the number of training tokens increased, the distance between members and non-members decreased.

In the case where the members of the class did not share any function words, the distance between sentences that belonged to the class and sentences that did not belong to the class did not differ be-
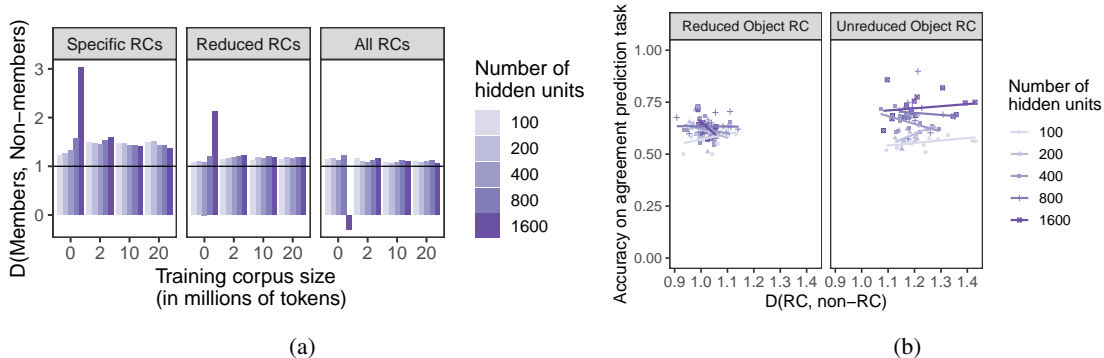
Figure 5: (a) Effect of hidden layer size and corpus size on the distance between sentences with specific RCs and sentences without (left), between sentences that match in reduction and sentences that do not (middle) and between sentences with RCs and sentences without (right). The solid black line indicates the point at which sentences that belong to a particular class are equally similar to other sentences that belong to that class and sentences that do not. (b) Agreement prediction accuracy on reduced object RCs and unreduced object RCs as a function of $\mathbb{D}(RC, \neg RC)$

tween the trained models and the baseline models. This suggests that any similarity between sentences in the representation space of trained models was driven purely by the presence (or in this case absence) of lexical items.

5.6 Does $\mathbb{D}(RC, \neg RC)$ predict agreement prediction accuracy?

Marvin and Linzen (2018) created a dataset that evaluated the grammaticality of the predictions of language models. Using this dataset, they showed that LSTM LMs could not accurately predict the number of the main verb if the main clause subject was modified by an object RCs (either reduced or unreduced). However, the models had better performance if the main clause was modified by an active subject RC. For example, the models were at near chance levels in predicting that (15a) should have higher probability than (15b), but were slightly better at predicting that (16a) should have higher probability than (16b):

(15) a. The farmer that the parents love swims.
b. *The farmer that the parents love swim.

(16) a. The farmer that loves the parents swims.
b. *The farmer that loves the parents swim.

One possible explanation for this poor performance is that object RCs, either reduced or unreduced, are quite infrequent (Roland et al., 2007). If the LM treats object RCs as unrelated to other RCs, there are likely very few training examples from which the models can learn about subject-verb agreement when the subject is modified by an object RC. If the LM had instead treated ob-

ject RCs as belonging to the same class as other RCs, it could learn to generalize from training examples of subject-verb agreement when the subject is modified by other RCs. This suggests the hypothesis that agreement prediction accuracy on object RCs will be higher in LMs in which the representation of object RCs is more similar to the representation of other RCs.

The similarity between object RCs and other RCs was defined as in the previous section (the proportion of blue squares to pink squares of the top two rows in Figure 4). There was an increase in accuracy as the number of hidden units increased (see Figure 5b). However, the similarity between object RCs and other types of RCs did not significantly correlate with agreement prediction; we therefore did not find any evidence for the hypothesis mentioned above.[10]

6 Discussion

Drawing on the syntactic priming paradigm from psycholinguistics, we proposed a new technique to analyze how the representations of sentences in neural language models (LMs) are organized. Applying this paradigm to sentences with relative clauses (RCs), we found that the representations of these sentences were organized in a linguistically interpretable hierarchical manner (summarized in Figure 6).

We investigated whether this hierarchical organization was driven by function words that are shared among sentences sentences or whether there was evidence that LMs were tracking more

---

[10] Similar patterns were observed for the other constructions in the dataset. See Supplementary Materials.
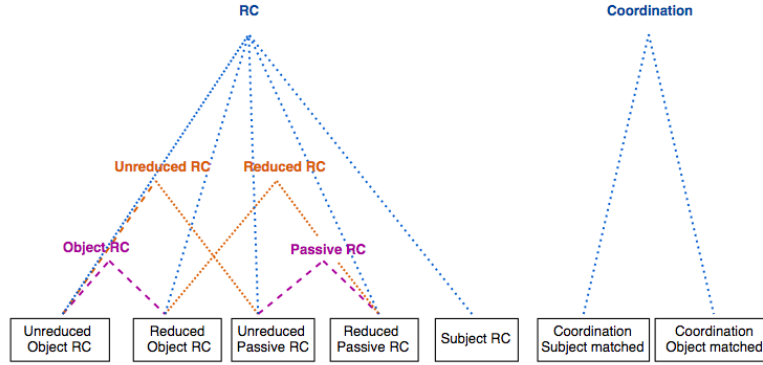
Figure 6: A schematic of how sentences belonging to different linguistically defined classes are related to each other in the LMs' representation space. Each colour indicates a different level of hierarchy.

abstract properties of the sentence. We found that for at least some linguistically interpretable classes, sentences that belonged to these classes were more similar to each other in the representation space of the LMs we tested than in the representation space of baseline LMs that were not trained on any data. This suggests that the trained LMs were capable of tracking abstract properties of the sentence.

However, for linguistically interpretable classes in which sentences shared a non-lexically observable property (e.g. presence of a gap), sentences were as similar to each other in the representation space of the LMs we tested as in the representation space of baseline LMs. Taken together, these results suggest that LMs might be able to track abstract properties of classes of sentences only if these classes also share a lexically observable property.

Additionally, we found that the sentences belonging to linguistically interpretable classes were more similar to each other in the representation spaces of models trained on 2 million tokens than in the representation spaces for models trained on 20 million tokens. We infer from this that LMs' ability to track abstract properties of sentences decreases with an increase in the training corpus size. This suggests that if we want these LMs to track more abstract linguistic properties, training them on more data from the same distribution is unlikely to help (cf. van Schijndel et al. 2019). Future work can explore how to bias these models to track linguistically useful properties through architectural biases (Dyer et al., 2016), training on auxiliary tasks (Enguehard et al., 2017) or data augmentation (Perez and Wang, 2017).

We hypothesized that models' accuracy on subject verb agreement when preceded by object RCs would increase as the similarity between object RCs and the other types of RCs increased. However, we did not find evidence for this. This could either be because the similarity between object RCs and the other types of RCs was too weak to be useful (see Figure 5a) or because the LMs do not use this property when predicting verb agreement. Future work can disambiguate these reasons by testing models that are biased to treat sentences with object RCs and other RCs as being similar.

Finally, our method allows us to generate a similarity matrix in the LMs representation space for any given set of structures. In the future, generating a similar matrix for human representations using priming experiments and comparing these two matrices using analysis methods from cognitive neuroscience (Kriegeskorte et al., 2008) may enable us to gain insight into how human-like the LM representations are and vice versa.

7 Conclusion

We proposed a novel technique to analyze how the representations of various syntactic structures are organized in neural language models. As a case study, we applied this technique to gain insight into the representations of sentences with relative clauses in RNN language models and found that the representations of sentences were organized in a linguistically interpretable manner.

8 Acknowledgments

References

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. *Journal of Natural Language Engineering*, 25(4):543–557.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

J. Kathryn Bock. 1986. Syntactic persistence in language production. *Cognitive Psychology*, 18(3):355–387.

Holly P. Branigan and Martin J. Pickering. 2017. An experimental approach to linguistic representation. *Behavioral and Brain Sciences*, 40.

Franklin Chang, Gary S. Dell, and Kathryn Bock. 2006. Becoming syntactic. *Psychological Review*, 113(2):234.

Shammur Absar Chowdhury and Roberto Zamparelli. 2019. An LSTM adaptation study of (un)grammaticality. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 204–212, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Émile Enguehard, Yoav Goldberg, and Tal Linzen. 2017. Exploring the syntactic abilities of RNNs with multi-task learning. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 3–14, Vancouver, Canada. Association for Computational Linguistics.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017. Improving neural language models with a continuous cache. In Yoshua Bengio and Yann LeCun, editors, *Proceedings of the Fifth International Conference on Learning Representations*. International Conference on Learning Representations.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, pages 1–8, Pittsburgh, PA. Association for Computational Linguistics.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Ákos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4):761–780.

Michael P. Kaschak, Timothy J. Kutta, and John L. Jones. 2011. Structural priming as implicit learning: Cumulative priming effects and individual differences. *Psychonomic Bulletin & Review*, 18(6):1133–1139.

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. 2017. Dynamic evaluation of neural sequence models. Technical report, University of Edinburgh.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A. Bandettini. 2008. Representational similarity

analysis-connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Kyle Mahowald, Ariel James, Richard Futrell, and Edward Gibson. 2016. A meta-analysis of syntactic priming in language production. *Journal of Memory and Language*, 91:5–27.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2018. Revisiting the poverty of the stimulus: Hierarchical generalization without a hierarchical bias in recurrent neural networks. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 2093—2098, Austin, TX.

Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

José Pinheiro, Douglas Bates, et al. 2000. *Mixed-Effects Models in S and S-PLUS*. Springer Science & Business Media.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Analyzing linguistic knowledge in sequential model of sentence. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 826–835, Austin, Texas. Association for Computational Linguistics.

Douglas Roland, Fredric Dick, and Jeffrey L. Elman. 2007. Frequency of basic english grammatical structures: A corpus analysis. *Journal of Memory and Language*, 57(3):348–379.

Marten van Schijndel and Tal Linzen. 2018. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710, Brussels, Belgium. Association for Computational Linguistics.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesnt buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Kristen M Tooley and Matthew J Traxler. 2010. Syntactic priming effects in comprehension: A critical review. *Language and Linguistics Compass*, 4(10):925–937.

Noah Weber, Leena Shekhar, and Niranjan Balasubramanian. 2018. The fine line between linguistic generalization and failure in Seq2Seq-attention models. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 24–27, New Orleans, Louisiana. Association for Computational Linguistics.

Justine B. Wells, Morten H. Christiansen, David S. Race, Daniel J. Acheson, and Maryellen C. MacDonald. 2009. Experience and sentence processing: Statistical learning and relative clause comprehension. *Cognitive Psychology*, 58:250–271.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.