Neural Language Models as Psycholinguistic Subjects: Representations of Syntactic State

Richard Futrell[1], **Ethan Wilcox**[2], **Takashi Morita**[3,4], **Peng Qian**[5], **Miguel Ballesteros**[6], and **Roger Levy**[5]

[1]Department of Language Science, UC Irvine, rfutrell@uci.edu
[2]Department of Linguistics, Harvard University, wilcoxeg@g.harvard.edu
[3]Primate Research Institute, Kyoto University, tmorita@alum.mit.edu
[4]Department of Linguistics and Philosophy, MIT
[5]Department of Brain and Cognitive Sciences, MIT, {pqian,rplevy}@mit.edu
[6]IBM Research, MIT-IBM Watson AI Lab, miguel.ballesteros@ibm.com

Abstract

We investigate the extent to which the behavior of neural network language models reflects incremental representations of syntactic state. To do so, we employ experimental methodologies which were originally developed in the field of psycholinguistics to study syntactic representation in the human mind. We examine neural network model behavior on sets of artificial sentences containing a variety of syntactically complex structures. These sentences not only test whether the networks have a representation of syntactic state, they also reveal the specific lexical cues that networks use to update these states. We test four models: two publicly available LSTM sequence models of English (Jozefowicz et al., 2016; Gulordava et al., 2018) trained on large datasets; an RNN Grammar (Dyer et al., 2016) trained on a small, parsed dataset; and an LSTM trained on the same small corpus as the RNNG. We find evidence for basic syntactic state representations in all models, but only the models trained on large datasets are sensitive to subtle lexical cues signalling changes in syntactic state.

1 Introduction

It is now standard practice in NLP to derive sentence representations using neural sequence models of various kinds (Elman, 1990; Sutskever et al., 2014; Goldberg, 2017; Peters et al., 2018; Devlin et al., 2018). However, we do not yet have a firm understanding of the precise content of these representations, which poses problems for interpretability, accountability, and controllability of NLP systems. More specifically, the success of neural sequence models has raised the question of whether and how these networks learn robust syntactic generalizations about natural language, which would enable robust performance even on data that differs from the peculiarities of the training set.

Here we build upon recent work studying neural language models using experimental techniques that were originally developed in the field of psycholinguistics to study language processing in the human mind. The basic idea is to examine language models' behavior on targeted sentences chosen to probe particular aspects of the learned representations. This approach was introduced by Linzen et al. (2016), followed more recently by others (Bernardy and Lappin, 2017; Enguehard et al., 2017; Gulordava et al., 2018), who used an agreement prediction task (Bock and Miller, 1991) to study whether RNNs learn a hierarchical morphosyntactic dependency: for example, that *The key to the cabinets...* can grammatically continue with *was* but not with *were*. This dependency turns out to be learnable from a language modeling objective (Gulordava et al., 2018). Subsequent work has extended this approach to other grammatical phenomena, with positive results for filler–gap dependencies (Chowdhury and Zamparelli, 2018; Wilcox et al., 2018) and negative results for anaphoric dependencies (Marvin and Linzen, 2018).

In this work, we consider syntactic representations of a different kind. Previous studies have focused on relationships of **dependency**: one word licenses another word, which is tested by asking whether a language model favors one (grammatically licensed) form over another in a particular context. Here we focus instead on whether neural language models show evidence for incremental **syntactic state** representations: whether behavior of neural language models reflects the kind of generalizations that would be captured using a stack-based incremental parse state in a symbolic grammar-based model. For example, during the underlined portion of Example (1), an incremental language model should represent and maintain the knowledge that it is currently inside a subordinate clause, implying (among other things) that a full main clause must follow.

Proceedings of NAACL-HLT 2019, pages 32–42
Minneapolis, Minnesota, June 2 - June 7, 2019.

(1) As the doctor studied the textbook, the nurse walked into the office.

In this work, we use a targeted evaluation approach (Marvin and Linzen, 2018) to elicit evidence for syntactic state representations from language models. That is, we examine language model behavior on artificially constructed sentences designed to expose behavior that is crucially dependent on syntactic state representations. In particular, we study complex subordinate clauses and garden path effects (based on main-verb/reduced-relative ambiguities and NP/Z ambiguities). We ask three general questions: (1) Is there basic evidence for the representation of syntactic state? (2) What textual cues does a neural language model use to infer changes to syntactic state? (3) Do the networks maintain knowledge about syntactic state over long spans of complex text, or do the syntactic state representations degrade?

Among neural language models, we study both generic sequence models (LSTMs), which have no explicit representation of syntactic structure, and an RNN Grammar (RNNG) (Dyer et al., 2016), which explicitly calculates Penn Treebank-style context-free syntactic representations as part of the process of assigning probabilities to words. This comparison allows us to evaluate the extent to which explicit representation of syntactic structure makes models more or less sensitive to syntactic state. RNNGs have been found to outperform LSTMs not only in overall test-set perplexity (Dyer et al., 2016), but also in modeling long-distance number agreement in Kuncoro et al. (2018) for certain model configurations; our work extends this comparison to a variety of syntactic state phenomena.

2 General methods

We investigate neural language model behavior primarily by studying the **surprisal**, or log inverse probability, that a language model assigns to each word in a sentence:

$$S(x_i) = -\log_2 p(x_i|h_{i-1}),$$

where x_i is the current word or character, h_{i-1} is the model's hidden state before consuming x_i, the probability is calculated from the network's softmax activation, and the logarithm is taken in base 2, so that surprisal is measured in bits. Surprisal is equivalent to the pointwise contribution to the language modeling loss function due to a word.

In psycholinguistics, the common practice is to study reaction times per word (for example, reading time as measured by an eyetracker), as a measure of the word-by-word difficulty of online language processing. These reading times are often taken to reflect the extent to which humans expect certain words in context, and may be generally proportional to surprisal given the comprehender's probabilistic language model (Hale, 2001; Levy, 2008; Smith and Levy, 2013; Futrell and Levy, 2017). In this study, we take language model surprisal as the analogue of human reading time, using it to probe the neural networks' expectations about what words will follow in certain contexts. There is a long tradition linking RNN performance to human language processing (Elman, 1990; Christiansen and Chater, 1999; MacDonald and Christiansen, 2002) and grammaticality judgments (Lau et al., 2017), and RNN surprisals are a strong predictor of human reading times (Frank and Bod, 2011; Goodkind and Bicknell, 2018). RNNGs have also been used as models of human online language processing (Hale et al., 2018).

2.1 Experimental methodology

In each experiment presented below, we design a set of sentences such that the word-by-word surprisal values will show evidence for syntactic state representations. The idea is that certain words will be surprising to a language model only if the model has a representation of a certain syntactic state going into the word. We analyze word-by-word surprisal profiles for these sentences using regression analysis. Except where otherwise noted, all statistics are derived from linear mixed-effects models (Baayen et al., 2008) with sum-coded fixed-effect predictors and maximal random slope structure (Barr et al., 2013). This method lets us factor out by-item variation in surprisal and focus on the contrasts between conditions.

2.2 Models tested

We study the behavior of four models of English: two LSTMs trained on large data, an an RNNG and an LSTM trained on matched, smaller data (the Penn Treebank). The models are summarized in Table 1. All models are trained on a language modeling objective.

Our first LTSM is the model presented in Jozefowicz et al. (2016) as "BIG LSTM+CNN Inputs", which we call "JRNN", which was trained on the One Billion Word Benchmark (Chelba et al., 2013) with two hidden layers of 8196 units each

Model	Architecture	Training data	Data size (tokens)	Reference
JRNN	LSTM	One Billion Word	$\sim$ 800 million	Jozefowicz et al. (2016)
GRNN	LSTM	Wikipedia	$\sim$ 90 million	Gulordava et al. (2018)
RNNG	RNN Grammar	Penn Treebank	$\sim$ 1 million	Dyer et al. (2016)
TinyLSTM	LSTM	Penn Treebank	$\sim$ 1 million	—

Table 1: Models tested, by architecture, training data, and training data size.

and CNN character embeddings as input. The second large LSTM is the model described in the supplementary materials of Gulordava et al. (2018), which we call "GRNN", trained on 90 million tokens of English Wikipedia with two hidden layers of 650 hidden units each.

Our RNNG is trained on syntactically labeled Penn Treebank data (Marcus et al., 1993), using 256-dimensional word embeddings for the input layer and 256-dimensional hidden layers, and dropout probability 0.3. Next-word predictions are obtained through hierarchical softmax with 140 clusters, obtained with the greedy agglomerative clustering algorithm of Brown et al. (1992). We estimate word surprisals using word-synchronous beam search (Stern et al., 2017; Hale et al., 2018): at each word w_i a beam of incremental parses is filled, the summed forward probabilities (Stolcke, 1995) of all candidates on the beam is taken as a lower bound on the prefix probability: $P_{\text{min}}(w_{1\ldots i})$, and the surprisal of the i-th word in the sentence is estimated as $\log \frac{P_{\text{min}}(w_{1\ldots i})}{P_{\text{min}}(w_{1\ldots i-1})}$. Our action beam is size 100, and our word beam is size 10. Finally, to disentangle effects of training set from model architecture, we use an LSTM trained on string data from the Penn Treebank training set, which we call TinyLSTM. For TinyLSTM we use 256-dimensional word-embedding inputs and hidden layers and dropout probability 0.3, just as with the RNNG.

3 Subordinate clauses

We begin by studying subordinate clauses, a key example of a construction requiring stack-like representation of syntactic state. In such constructions, as shown in Example (1), a **subordinator** such as "as" or "when" serves as a cue that the following clause is a subordinate clause, meaning that it must be followed by some main (matrix) clause. In an incremental language model, this knowledge must be maintained and carried forward while processing the words inside subordinate clause. A grammar-based symbolic language model (e.g., Stolcke, 1995; Manning and Carpenter, 2000) would maintain this knowledge by keeping track of syntactic rules representing the incomplete subordinate clause and the upcoming main clause in a stack data structure. Psycholinguistic research has clearly demonstrated that humans maintain representations of this kind in syntactic processing (Staub and Clifton, 2006; Lau et al., 2006; Levy et al., 2012). Here we ask whether the string completion probabilities produced by neural language models show evidence of the same knowledge.

We can detect the knowledge of syntactic state in this case by examining whether the network licenses and requires a matrix clause following the subordinate clause. These expectations can be detected by examining surprisal differences between sentences of the form in Example (2):

(2) a. As the doctor studied the textbook, the nurse walked into the office. [SUBordinator, MATRIX]
b. *As the doctor studied the textbook. [SUB, NO-MATRIX]
c. ?The doctor studied the textbook, the nurse walked into the office. [NO-SUBordinator, MATRIX]
d. The doctor studied the textbook. [NO-SUB, NO-MATRIX]

If the network *licenses* a matrix clause following the subordinate clause—and maintains knowledge of that licensing relationship throughout the clause, from the subordinator to the comma—then this should be manifested as lower surprisal at the matrix clause in (2-a) as compared to (2-c). We call this the **matrix licensing effect**: the surprisal of the condition [SUB, MATRIX] minus [NOSUB, MATRIX], which will be negative if there is a licensing effect. If the network *requires* a following matrix clause, then this will be manifested as higher surprisal at the matrix clause for (2-b) compared with (2-d). We call this the **no-matrix penalty effect**: the surprisal of [SUB,NOMATRIX] minus [NOSUB, NOMATRIX], which will be positive if there is a penalty.

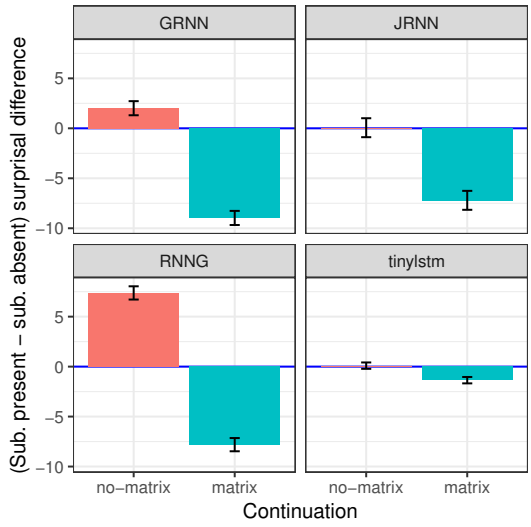

Figure 1: Effect of subordinator absence/presence on surprisal of continuations. Red: no-matrix penalty effect. Blue: matrix licensing effect. In this and all other figures, unless otherwise noted, **error bars** represent 95% confidence intervals of the *contrasts between conditions shown*, computed from the standard error of the by-item and by-condition mean surprisals after subtracting out the by-item means (Masson and Loftus, 2003).

We designed 23 experimental items on the pattern of (2) and calculated difference in the sum surprisal of the words in the matrix clause.[1] Figure 3 shows the matrix licensing effect (in blue) and the no-matrix penalty effect (in red), averaged across items. For all models, we see a facilitative matrix licensing effect ($p < .001$ for all models), smallest in TinyLSTM. However, we only find a significant no-matrix penalty for GRNN and the RNNG ($p < .001$ in both): the other models do not significantly penalize an ungrammatical continuation ($p = .9$ for JRNN; $p = .5$ for TinyLSTM). That is, JRNN and TinyLSTM give no indication that (2-b) is less probable than (2-c).

We found that all models at least partially represent the licensing relationship between a subordinate and matrix clause. However, in order to fully represent the syntactic requirements induced by a subordinator, it seems that a model needs either large amounts of data (as in GRNN) or explicit representation of syntax (as in the RNNG, as opposed to TinyLSTM).

[1]Note that it would not be sufficient to look at surprisal only at the punctuation token, because the comma could indicate the beginning of a conjoined NP.

3.1 Maintenance and degradation of syntactic state

The foregoing results show that neural language models use the presence of a subordinator as a cue to the onset of a subordinate clause, and that they maintain knowledge that they are in a subordinate clause throughout the intervening material up to the comma. Now we probe the ability of models to maintain this knowledge over long spans of complex intervening material. To do so, we use sentences on the template of (2) and add intervening material modifying the NPs in the subordinate clause. To both of these NPs (in subject and object position), we add modifiers of increasing syntactic complexity: PPs, subject-extracted relative clauses (SRCs), and object-extracted relative clauses (ORCs), as shown in Figure 2. We study the extent to which these modifiers weaken the language models' expectations about the upcoming matrix clause.

As a summary measure of the strength of language models' expectations about an upcoming matrix clause, we collapse the two measures of the previous section into one: the **matrix licensing interaction**, consisting of the difference between the no-matrix penalty effect and the matrix licensing effect (the two bars in Figure 1). A similar measure was used to detect filler–gap dependencies by Wilcox et al. (2018).

Figure 3 shows the strength of the matrix licensing interaction given sentences with various modifiers inserted. For the large LSTMs, GRNN exhibits a strong interaction when the intervening material is short and syntactically simple, and the interaction gets progressively weaker as the intervening material becomes progressively longer and more complex ($p < 0.001$ for subject postmodifiers and $p < 0.01$ object postmodifiers). The other models show less interpretable behavior.

Our results indicate that at least some large LSTMs, along with the RNNG, are capable of maintaining a representation of syntactic state over spans of complex intervening material. Quantified as a licensing interaction, this representation of syntactic state exhibits the most clearly understandable behavior in GRNN, which shows a graceful degradation of syntactic expectations as the complexity of intervening material increases. The representation is maintained most strongly in the RNNG, except for one particular construction (object-position SRCs).

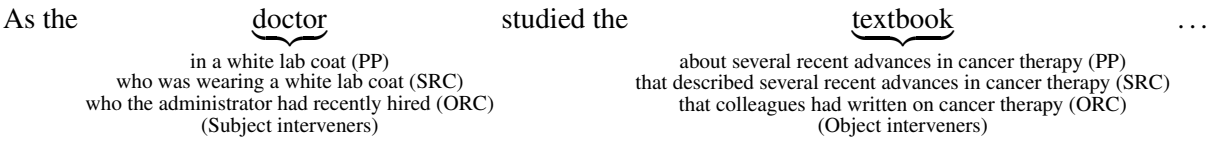

Figure 2: Scheme for lengthening the subordinate clause in Section 3.1.

4 Garden path effects

The major phenomenon that has been used to probe incremental syntactic representations in humans is **garden path effects**. Garden path effects arise from local ambiguities, where a context leads a comprehender to believe one parse is likely, but then a disambiguating word forces her to drastically revise her beliefs, resulting in high surprisal/reading time at the disambiguating word. In effect, the comprehender is "led down the garden path" by a locally likely but ultimately incorrect parse (Bever, 1970). Garden-pathing in LSTMs has recently been demonstrated by van Schijndel and Linzen (2018a,b) in the context of modeling human reading times.

Garden path effects allow us to detect representations of syntactic state because if a person or language model shows a garden path effect at a word, that means that the person or model had some belief about syntactic state which was disconfirmed by that word. In psycholinguistics, these effects have been used to study the question of what information determines people's beliefs about likely parses given locally ambiguous contexts: for example, whether factors such as world knowledge play a role (Ferreira and Clifton, 1986; Trueswell et al., 1994).

Here we study two major kinds of local ambiguities inducing garden path effects. For each ambiguity, we ask two main questions. First, whether the network shows the basic garden path effect, which would indicate that it had a syntactic state representation that made a disambiguating word surprising. Second, whether the network is sensitive to subtle lexical cues to syntactic structure which may modulate the size of the garden path effect: this question allows us to determine what information the network uses to determine the beginnings and endings of certain syntactic states.

4.1 NP/Z Ambiguity

The **NP/Z ambiguity**[2] refers to a local ambiguity in sentences of the form given in Example (3).

(3)a. When the dog scratched the vet with his new assistant **took off** the muzzle. [TRANSITIVE, NOCOMMA]
b. When the dog scratched, the vet with his new assistant **took off** the muzzle. [TRANSITIVE, COMMA]
c. When the dog struggled the vet with his new assistant **took off** the muzzle. [INTRANSITIVE, NOCOMMA]
d. When the dog struggled, the vet with his new assistant **took off** the muzzle. [INTRANSITIVE, COMMA]

When a comprehender reads the underlined phrase "the vet with his new assistant" in (3-a), she may at first believe that this phrase is the direct object of the verb "scratched" inside the subordinate clause. However, upon reaching the verb "took off", she realizes that the underlined phrase was not in fact an object of the verb "scratched", rather it was the subject of a new clause, and the subordinate clause in fact ended after the verb "scratched". The key region of the sentence where the garden path disambiguation happens—called the **disambiguator**—is the phrase "took off", marked in bold.

While a garden path should obtain in (3-a), no such garden path should exist for (3-b), because a comma clearly demarcates the end of the subordinate clause. Therefore a basic garden path effect would be indicated by the difference in surprisal at the disambiguator for (3-a) minus (3-b). Furthermore, if a comprehender is sensitive to the relationship between verb argument structure and clause boundaries, then there should be no garden path in (3-c), because the verb "struggled" is INTRANSITIVE: it cannot take an object in English, so an incremental parser should never be misled into believing that "the vet..." is its object. This lexical information about syntactic structure is subtle enough that there has been controversy about whether even humans are sensitive to it in online processing (Staub, 2007).

4.1.1 NP/Z Garden Path Effect

We tested whether neural language models would show the basic garden path effect and if this ef-

[2]For Noun Phrase/Zero ambiguity. At first the embedded verb appears to take an NP object, but later it turns out that it was a zero (null) object.

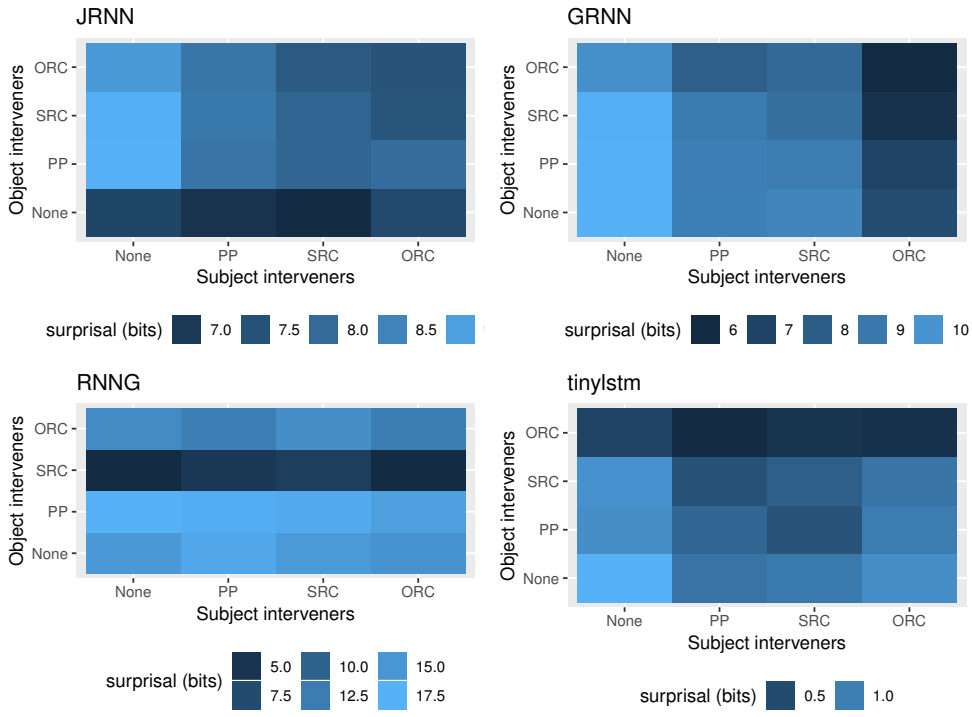

Figure 3: Size of matrix clause licensing interaction (see text) given various intervening elements in the subordinate clause. Note that the heatmaps are on different scales across models.

fect would be modulated by verb transitivity. We constructed 32 items based of the same structure as (3), based on materials from Staub (2007), manipulating the transitivity of the embedded verb ("scratched" vs. "struggled"), and the presence of a disambiguating comma at the end of the subordinate clause. An NP/Z garden path effect would show up as increased surprisal at the main verb "took off" in the absence of a comma. If the networks use the transitivity of the embedded verb as a cue to clause structure, and maintain that information over the span of six words between the embedded verb and the main verb, then there should be a garden path effect for the transitive verb, but not for the intransitive verb. More generally we would expect a *stronger* garden path given the transitive verb than given the intransitive verb.

Figure 4 shows the mean surprisals at the disambiguator for all four models, for both transitive and intransitive embedded verbs. The overall per-region surprisals, averaged over words in each region, are shown in Figure 5. We see that a garden path effect exists in all models (though very small in TinyLSTM): all models show significantly higher surprisal at the main verb when the disambiguating comma is absent ($p < .001$ for all models). However, only the large LSTMs appear to be sensitive to the transitivity of the embedded verb, showing a smaller garden path effect for intransitive verbs. Statistically, there is a significant interaction of comma presence and verb transitivity only in GRNN and JRNN (GRNN: $p < .01$; JRNN: $p < .001$; RNNG: $p = .3$, TinyLSTM: $p = .3$).

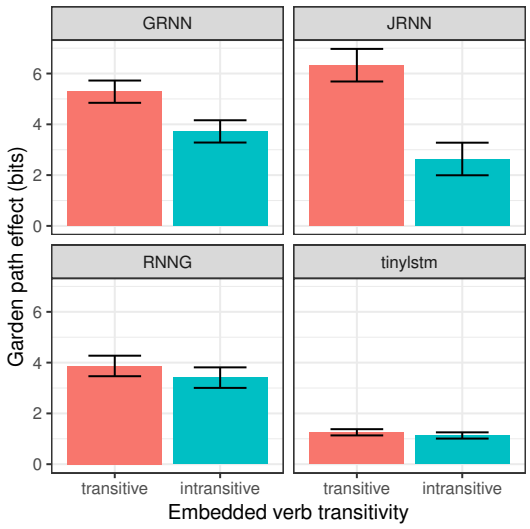

Figure 4: Average garden path effect (surprisal at disambiguator in NO-COMMA condition minus COMMA condition) by model and embedded verb transitivity.

All models show NP/Z garden path effects, indicating that they are sensitive to some cues indicat-

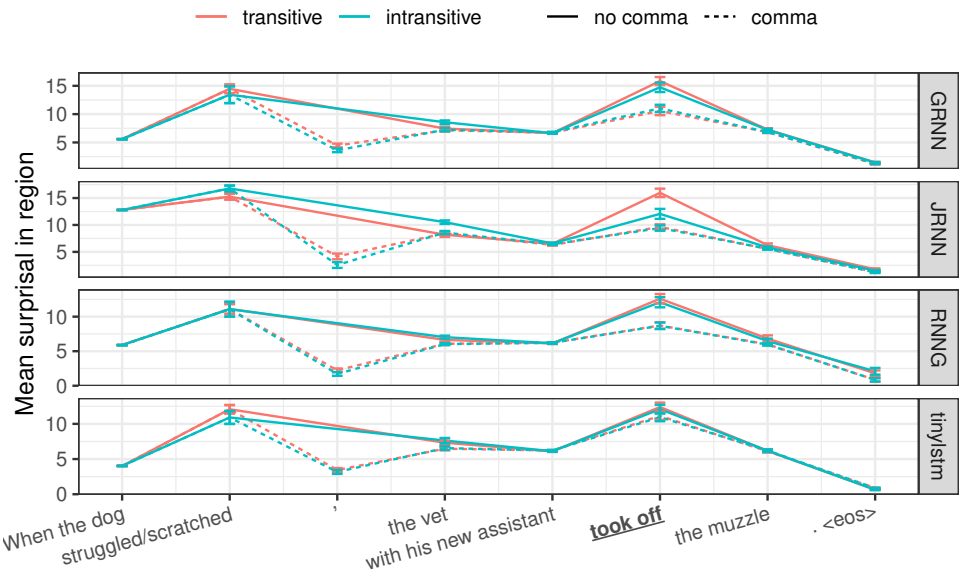

Figure 5: Region-by-region surprisal values for NP/Z garden path materials. Surprisal values are averaged across items and across words in regions. The critical region where the garden path effect is visible is the verb "took off".

ing end-of-clause boundaries. However, only the large LSTMs appear to use verb argument structure information as a cue to these boundaries. The results suggest that very large amounts of data may be necessary for current neural models to discover such fine-grained dependencies between syntactic properties of verbs and sentence structure.

4.1.2 Maintenance and degradation of state

We can probe the maintenance and degradation of syntactic state information by manipulating the length of the intervening material between the onset of the local ambiguity and the disambiguator in examples such as (3). The question is whether the networks maintain the knowledge, while processing the intervening material, that the intervening noun phrase is probably the object of the embedded verb inside a subordinate clause, or whether they gradually lose track of this information. To study this question we used materials on the pattern of (4): these materials manipulate the length of the intervening material (underlined) while holding constant the distance between the subordinator ("As") and the disambiguator (**grew**).

(4)a. As the author studying Babylon in ancient times wrote the book **grew**. [SHORT, NO-COMMA]

b. As the author studying Babylon in ancient times wrote, the book **grew**. [SHORT, COMMA]

c. As the author wrote the book describing Babylon in ancient times **grew**. [LONG, NO-COMMA]

d. As the author wrote, the book describing Babylon in ancient times **grew**. [LONG, COMMA]

If neural language models show degradation of syntactic state, then the garden path effect (measured as the difference in surprisal between the COMMA and NO-COMMA conditions at the disambiguator) will be smaller for the LONG conditions. We tested 32 sentences of the form in (4), based on materials from Tabor and Hutchins (2004). The garden path effect sizes are shown in Figure 6.

We find a significant garden effect in all models in the SHORT condition ($p < .001$ in JRNN and GRNN; $p < .01$ in the RNNG and $p = .03$ in TinyLSTM). In the long condition, we find the garden path effect in all models except TinyLSTM: ($p < .001$ in JRNN; $p < .01$ in GRNN; $p = .02$ in the RNNG; and $p = .2$ in TinyLSTM). The crucial interaction between length and comma presence (indicating that syntactic state degrades) is significant in GRNN ($p < .01$) and TinyLSTM ($p < .001$) but not JRNN ($p = .7$) nor the RNNG ($p = .6$). The pattern is reminiscent of the results on degradation of state information about subordinate clauses in Section 3, where GRNN and TinyLSTM showed the clearest evidence of degradation.

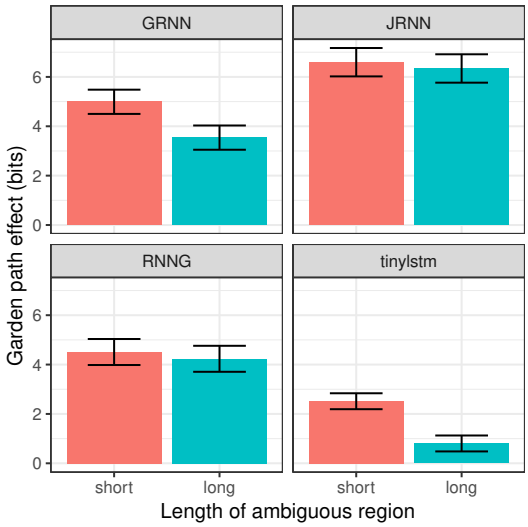

Figure 6: Average garden path effect by model and length of ambiguous region.

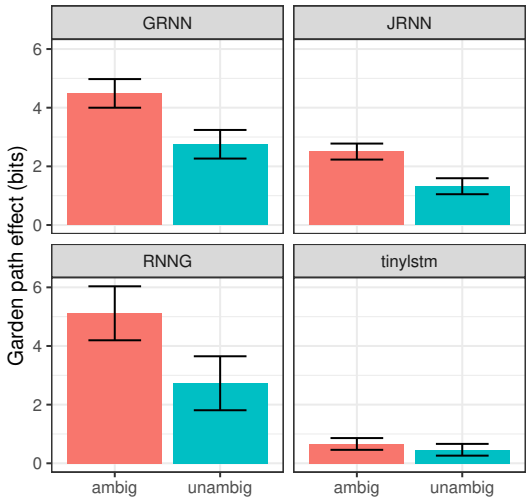

Figure 7: Garden path effect size for MV/RR ambiguity by model and verb-form ambiguity.

Note that the pattern found here is the opposite of the pattern of human reading times. Humans appear to show "digging-in" effects: the longer the span of time between the introduction of a local ambiguity and its resolution, the larger the garden path effect (Tabor and Hutchins, 2004; Levy et al., 2009).

4.2 Main Verb/Reduced Relative Ambiguity

Next we turn to garden path effects induced by the classic **Main Verb/Reduced Relative (MV/RR) ambiguity**, in which a word is locally ambiguous between being the main verb of a sentence or introducing a **reduced relative clause** (reduced RC: a relative clause with no explicit complementizer, headed by a passive-participle verb). That ambiguity can be maintained over a long stretch of material:

(5)a. The woman brought the sandwich from the kitchen **tripped** on the carpet. [REDUCED, AMBIGuous]
b. The woman who was brought the sandwich from the kitchen **tripped** on the carpet. [UNREDUCED, AMBIG]
c. The woman given the sandwich from the kitchen **tripped** on the carpet. [REDUCED, UNAMBIGuous]
d. The woman who was given the sandwich from the kitchen **tripped** on the carpet. [UNREDUCED, UNAMBIG]

In Example (5-a), the verb "brought" is initially analyzed as a main verb phrase, but upon reaching the verb "tripped"—the disambiguator in this case—the reader must re-analyze it as an RC. The garden path should be eliminated in sentences such as (5-b), the UNREDUCED condition, where the words "who was" clarify that the verb "brought" is part of an RC, rather than the main verb of the sentence. Therefore we quantify the garden path effect as the surprisal at the disambiguator for the REDUCED minus UNREDUCED conditions.

There is another possible cue that the initial verb is the head of an RC: the morphological form of the verb. In examples such as (5-c), the the verb "given" is unambiguously in its past-participle form, indicating that it cannot be the main verb of the sentence. If a language model is sensitive to morphological cues to syntactic structure, then it should either not show a garden path effect in this UNAMBIGuous condition, or it should show a reduced garden path effect.

We constructed 29 experimental items following the template of (5). Figure 7 shows the garden path effect sizes by model and verb-form ambiguity. All networks show the basic garden path effect ($p < .001$ in JRNN, GRNG, and RNNG; $p < 0.01$ in TinyLSTM). However, the garden path effect in TinyLSTM is much smaller than the other models: RC reduction causes an additional .3 bits of surprisal at the disambiguating verb, as compared to 2.8 bits in the RNNG, 1.9 in JRNN, and 3.6 in GRNN (TinyLSTM's garden path effect is significantly smaller than each other model at $p < 0.001$).

If the network is using the morphological form

Phenomenon	GRNN	JRNN	RNNG	TinyLSTM
Subordination	✓✓	✓✗	✓✓	✓✗
NP/Z Garden Path	✓✓	✓✓	✓✗	✓✗
MV/RR Garden Path	✓✓	✓✓	✓✓	✓✗

Table 2: Summary of results by model and phenomenon. The first check mark indicates basic evidence of syntactic state representation. The second check mark indicates the ability to capture more fine-grained phenomena: for subordination, the no-matrix penalty effect; for the NP/Z garden path, the effect of verb transitivity; and for the MV/RR garden path, the effect of verb morphology.

of the verb as a cue to syntactic structure, then it should show the garden path effect more strongly in the AMBIG condition than the UNAMBIG condition. The large language models and the RNNG do show this pattern: at the critical main-clause verb, surprisal is superadditively highest in the reduced ambiguous condition (the dotted blue line; a positive interaction between the reduced and ambiguous conditions is significant in the three models at $p < 0.001$). However, TinyLSTM does not show evidence for superadditive surprisal for the ambiguous verbform and the reduced RC ($p = .45$).

The three large LSTMs and the RNNG replicate the key human-like garden-path disambiguation effect due to to ambiguity in verb form. But strikingly, even when the participial verbform is unambiguous, there is still a significant garden path effect in all models ($p < 0.01$ in all models except TinyLSTM, where $p = .08$). Apparently, these networks treat an unambiguous passive-participial verb as only a *noisy cue* to the presence of an RC.

5 General Discussion and Conclusion

In all models studied, we found clear evidence of basic incremental state syntactic representation. However, models varied in how well they fully captured the effects of such state and the potentially subtle lexical cues indicating the beginnings and endings of such states: only the large LSTMs could sometimes reliably infer clause boundaries from verb argument structure (Section 4.1) and morphological verb-form (Section 4.2), and only GRNN and the RNNG fully captured the proper behavior of subordinate clauses. The results are summarized in Table 2. We suggest that representation of course-grained syntactic structure requires either syntactic supervision or large data, while exploiting fine-grained lexical cues to structure requires large data.

More generally, we believe that the psycholinguistic methodology employed in this paper provides a valuable lens on the internal representations of black-box systems, and can form the basis for more systematic tests of the linguistic competence of NLP systems. We make all experimental items, results, and analysis scripts available online at `github.com/langprocgroup/nn_syntactic_state`.

References

R. Haraald Baayen, D.J. Davidson, and D.M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412.

Dale J. Barr, Roger P. Levy, Christoph Scheepers, and Harry J Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3):255–278.

Jean-Philippe Bernardy and Shalom Lappin. 2017. Using deep neural networks to learn syntactic agreement. *Linguistic Issues in Language Technology*, 15:1–15.

Thomas G. Bever. 1970. The cognitive basis for linguistic structures. In J. R. Hayes, editor, *Cognition and the Development of Language*. Wiley, New York.

Kathryn Bock and Carol A Miller. 1991. Broken agreement. *Cognitive Psychology*, 23(1):45–93.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144.

Morten H. Christiansen and Nick Chater. 1999. Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, 23(2):157–205.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

J.L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Emile Enguehard, Yoav Goldberg, and Tal Linzen. 2017. Exploring the syntactic abilities of RNNs with multi-task learning. *arXiv preprint arXiv:1706.03542*.

Fernanda Ferreira and Charles Clifton. 1986. The independence of syntactic processing. *Journal of Memory and Language*, 25(3):348–368.

Stefan L. Frank and Rens Bod. 2011. Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological Science*, 22(6):829–834.

Richard Futrell and Roger Levy. 2017. Noisy-context surprisal as a human sentence processing cost model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 688–698, Valencia, Spain.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, UT. Association for Computational Linguistics.

K. Gulordava, P. Bojanowski, E. Grave, T. Linzen, and M. Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of NAACL*.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan R Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, Melbourne, Australia.

John T. 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 and Language Technologies*, pages 1–8.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv*, 1602.02410.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1426–1436.

Ellen Lau, Clare Stroud, Silke Plesch, and Colin Phillips. 2006. The role of structural prediction in rapid syntactic analysis. *Brain & Language*, 98:74–88.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: a probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Roger Levy, Evelina Fedorenko, Mara Breen, and Ted Gibson. 2012. The processing of extraposed structures in English. *Cognition*, 122(1):12–36.

Roger P Levy, Florencia Reali, and Thomas L Griffiths. 2009. Modeling the effects of memory on human online sentence processing with particle filters. In *Advances in neural information processing systems*, pages 937–944.

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.

Maryellen C. MacDonald and Morten H. Christiansen. 2002. Reassessing working memory: Comment on Just and Carpenter (1992) and Waters and Caplan (1996). *Psychological Review*, 109(1):35–54.

Christopher D Manning and Bob Carpenter. 2000. Probabilistic parsing using left corner language models. In *Advances in probabilistic and other parsing technologies*, pages 105–124. Springer.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

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. Association for Computational Linguistics.

Michael EJ Masson and Geoffrey R Loftus. 2003. Using confidence intervals for graphically based data interpretation. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 57(3):203.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*.

Marten van Schijndel and Tal Linzen. 2018a. Modeling garden path effects without explicit hierarchical syntax. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*.

Marten van Schijndel and Tal Linzen. 2018b. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Adrian Staub. 2007. The parser doesn't ignore intransitivity, after all. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 33(3):550.

Adrian Staub and Charles Clifton. 2006. Syntactic prediction in language comprehension: Evidence from *either . . . or*. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 32(2):425–436.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700. Association for Computational Linguistics.

Andreas Stolcke. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Whitney Tabor and Sean Hutchins. 2004. Evidence for self-organized sentence processing: Digging-in effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(2):431.

John C. Trueswell, Michael K. Tanenhaus, and S. Garnsey. 1994. Semantic influences on parsing: Use of thematic role information in syntactic ambiguity resolution. *Journal of Memory and Language*, 33:285–318.

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. Association for Computational Linguistics.