Structural Supervision Improves Learning of Non-Local Grammatical Dependencies

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

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

Abstract

State-of-the-art LSTM language models trained on large corpora learn sequential contingencies in impressive detail and have been shown to acquire a number of non-local grammatical dependencies with some success. Here we investigate whether supervision with hierarchical structure enhances learning of a range of grammatical dependencies, a question that has previously been addressed only for subject-verb agreement. Using controlled experimental methods from psycholinguistics, we compare the performance of word-based LSTM models versus two models that represent hierarchical structure and deploy it in left-to-right processing: Recurrent Neural Network Grammars (RNNGs) (Dyer et al., 2016) and a incrementalized version of the Parsing-as-Language-Modeling configuration from Charniak et al. (2016). Models are tested on a diverse range of configurations for two classes of non-local grammatical dependencies in English—*Negative Polarity* licensing and *Filler–Gap Dependencies*. Using the same training data across models, we find that structurally-supervised models outperform the LSTM, with the RNNG demonstrating best results on both types of grammatical dependencies and even learning many of the *Island Constraints* on the filler–gap dependency. Structural supervision thus provides data efficiency advantages over purely string-based training of neural language models in acquiring human-like generalizations about non-local grammatical dependencies.

1 Introduction

Long Short-Term Memory Recurrent Neural Networks (LSTMs) (Hochreiter and Schmidhuber, 1997) have achieved state of the art language modeling performance (Jozefowicz et al., 2016) and have been shown to indirectly learn a number of non-local grammatical dependencies, such as subject-verb number agreement and filler-gap licensing (Linzen et al., 2016; Wilcox et al., 2018), although they fail to learn others, such as Negative Polarity Item and anaphoric pronoun licensing (Marvin and Linzen, 2018; Futrell et al., 2018). LSTMs, however, require large amounts of training data and remain relatively uninterpretable. One model that attempts to address both these issues is the Recurrent Neural Network Grammar (Dyer et al., 2016). RNNGs are generative models, which represent hierarchical syntactic structure and use neural control to deploy it in left-to-right processing. They can achieve state-of-the-art broad-coverage scores on language modeling and phrase structure parsing tasks, learn Noun Phrase headedness (Kuncoro et al., 2016), and outperform linear models at learning subject-verb number agreement (Kuncoro et al., 2018).

In this work, we comparatively evaluate LSTMs, RNNGs and a third model trained using syntactic supervision—similar to the Parsing-as-Language-Modeling configuration from Charniak et al. (2016)—by conducting side-by-side tests on two novel English grammatical dependencies, deploying methodology from psycholinguistics. In this paradigm, the language models are fed with hand-crafted sentences, designed to draw out behavior that belies whether they have learned the underlying syntactic dependency. For example, Linzen et al. (2016) and Kuncoro et al. (2018) assessed how well neural language models were able to learn subject-verb number agreement by feeding the prefix *The keys to the cabinet...* If the model assigns a relatively higher probability to the grammatical plural verb *are* than the ungrammatical singular *is* it can be said to have learned the agreement dependency. Here, we investigate two non-local dependencies that remain untested for RNNGs: Negative Polarity Item (NPI) licensing is the dependency between a negative licensor—

Proceedings of NAACL-HLT 2019, pages 3302–3312
Minneapolis, Minnesota, June 2 - June 7, 2019.

such as *not* or *none*—and a Negative Polarity Item such as *any* or *ever*. The filler–gap dependency is the dependency between a filler—such as *who* or *what*—and a gap, which is an empty syntactic position. Both dependencies have been shown to be learnable by LSTMs trained on large amounts of data (Wilcox et al., 2018; Marvin and Linzen, 2018). Here, we investigate whether, after controlling for size of the training data, explicit hierarchical representation results in learning advantages.

2 Methodology

2.1 Neural Language Models

Recurrent Neural Network LMs model a sentence in a purely sequential basis, without explicitly representing the latent syntactic structure. We use the LSTM architecture in Hochreiter and Schmidhuber (1997), deploying a 2-layer LSTM language model with hidden layer size 256, input embedding size 256, and dropout rate 0.3. We refer to this model as the "LSTM" model in the following sections.

Recurrent Neural Network Grammars (Dyer et al., 2016) predict joint probability of a sentence as well as its syntactic parse. RNNGs contain three sub-components, all of which are LSTMS: the *neural stack*, which keeps track of the current parse, the *output buffer*, which keeps track of previously-seen terminals and the *history of actions*. At each timestep the model can take three different actions: NT, which introduces a non-terminal symbol—such as a VP or NP—onto the stack; SHIFT, which places a terminal symbol onto the top of the stack, or REDUCE. REDUCE pops terminal symbols (words) off the stack until a non-terminal phrasal boundary is encountered; it then combines the terminals into a single representation via a bidirectional-LSTM and pushes the newly-reduced constituent back onto the stack. By reducing potentially unbounded constituents within the neural stack, the RNNG is able to create structural adjacency between co-dependent words that may be linearly distal. Following Dyer et al. (2016), we use 2-layer LSTMs with 256 hidden layer size for the stack-LSTM, action LSTM, and terminal LSTM, and dropout rate 0.3.

ActionLSTM: It is the combination of the neural stack and the REDUCE function that may give the RNNG an advantage over purely sequential models (such as LSTMS) or models that deploy syntactic supervision without explicit notions of compositionality. In order to assess the gains from explicitly modeling compositionality, we compare the previous two models against an incrementalized version of the Parsing-as-Language-Modeling configuration presented in Charniak et al. (2016). In this model, we strip an RNNG of its *neural stack* and *output buffer*, and train it to jointly predict the action sequence of a parse tree as well as the upcoming word. The action space of the model contains a set of non-terminal nodes (NT), terminal generations (GEN), as well as a (REDUCE) action, which functions only as a generic phrasal boundary marker. The model was trained using embedding size 256, dropout 0.3, and was able to achieve a parsing F1 score of 92.81 on the PTB, which is only marginally better than the performance of the original architecture on the same test set, as reported in Kuncoro et al. (2016). We will refer to this model as the "ActionLSTM" model in the following sections.

All three models are trained on the training-set portion of the English Penn Treebank standardly used in the parsing literature (PTB; sections 2-21), which consists of about **950,000 tokens** of English language news-wire text (Marcus et al., 1993). The RNNG and Action models get supervision from syntactic annotation–crucially, only constituent boundaries and major syntactic categories, with functional tags and empty categories stripped away—whereas the LSTM language model only uses the sequences of terminal words. We train the models until performance converges on the held-out PTB development-set data.

2.2 Psycholinguistic Assessment Paradigm

2.2.1 Surprisal

The **surprisal**, or negative log-conditional probability, $S(x_i)$ of a sentence's i^{th} word x_i, tells us how strongly x_i is expected in context and is also known to correlate with human processing difficulty (Smith and Levy, 2013; Hale, 2001; Levy, 2008). For sentences out of context, surprisal is:

$$S(x_i) = -\log p(x_i|x_1 \ldots x_{i-1})$$

We investigate a model's knowledge of a grammatical dependency, which is the co-variance between an upstream *licensor* and a downstream *licensee*, by measuring the effect that an upstream licensor has on the surprisal of a downstream licensee. The idea is that grammatical licensors

should set up an expectation for the licensee thus reducing its surprisal compared to minimal pairs in which the licensor is absent. We derive the word surprisal from the LSTM language model by directly computing the negative log value of the predicted conditional probability $p(x_i|x_1 \ldots x_{i-1})$ from the softmax layer.

Following the method in Hale et al. (2018) for estimating word surprisals from RNNG, we use word-synchronous beam search (Stern et al., 2017) to find a set of most likely incremental parses and sum their forward probabilities to approximate $P(x_1, \ldots x_i)$ and $P(x_1, \ldots x_{i-1})$ for computing the surprisal. We set the action beam size to 100 and word beam size to 10. We ensured that the correct incremental RNNG parses were present on the beam immediately before and throughout the material over which surprisal was calculated through manual spot inspection; the correct parse was almost always at the top of the beam.

2.2.2 Wh-Licensing Interaction

Unlike NPI, licensing, the filler—gap dependency is the covariance between a piece of extant material, a filler, and a piece of *absent* material, a gap. Here, we employ the methodology from Wilcox et al. (2018), which introduces the **Wh-Licensing Interaction**. To compute the wh-licensing interaction for a sentence, Wilcox et al. (2018) construct four variants, given in (1), that exhibit the four possible combinations of fillers and gaps for a specific syntactic position. The underscores are for presentational purposes only and were not included in experimental materials.

(1) a. I know that the lion devoured the gazelle at sunrise. [-FILLER -GAP]
b. *I know what the lion devoured the gazelle at sunrise. [+FILLER -GAP]
c. *I know that the lion devoured __ at sunrise. [-FILLER +GAP]
d. I know what the lion devoured __ at sunrise. [+FILLER +GAP]

If a filler sets up an expectation for a gap, then filled syntactic positions should be more surprising in the context of a filler than in a minimally-different, non-filler variants. We measure this expectation by calculating the difference of surprisal between (1-b) and (1-a). Similarly, if gaps require fillers to be licensed, transitions from transitive verbs to adjunct clauses that skip an obligatory argument should be less surprising in the context of a filler than in minimally-different, non-filler variants. We measure this expectation by computing the difference in surprisal between (1-c) and (1-d). Because the filler–gap dependency is a two-way interaction, the wh-licensing interaction consists of the difference of these two differences, which is given in (2).

(2) $(S(\text{1-b}) - S(\text{1-a})) - (S(\text{1-c}) - S(\text{1-d}))$

For basic filler—gap dependencies, we expect the presence of a filler to set up a global expectation for a gap, thus we measure the summed licensing interaction across the entire embedded clause, which we expect to be significantly above zero if the model is learning the dependency. Our experimental materials include only vocabulary items within the PTB, avoiding the need for Out of Vocabulary handling. We determine statistical significance using a mixed-effects linear regression model, using sum-coded conditions (Baayen et al., 2008). For within-model comparison we use surprisal as the dependent variable and experimental conditions as predictors; for between-model comparison, we use wh-licensing interaction as the dependent variable with model type and experimental conditions as predictors. All figures depict by-item means, with error bars representing 95% confidence intervals, computed by subtracting out the within-item means from each condition as advocated by Masson and Loftus (2003). The strength of a wh-licensing interaction can be interpreted as either its mean size in bits, or as its mean size normalized by its standard deviation across items. The latter is Cohen's d, rooted in signal-detection theory (**?**); because all our experiments involve similar number of items, it is roughly proportional to the size of wh-interaction relative to the size of the associated confidence interval.[1]

3 Negative Polarity Item Licensing

In English, Negative Polarity Items (NPIs), such as *any*, *ever* must be in the SCOPE of a negative LICENSOR such as *no*, *none*, or *not* (**?**Ladusaw, 1979). Crucially, the scope of a licensor is characterized structurally, not in purely linear terms; for present purposes, a sufficient approximation is that an NPI is in the proper scope of a licensor if it is c-commanded by it. Thus while *ever* in (3-b) and (3-d) is grammatical because it is licensed by *no* in the main-clause subject, *ever* is ungrammatical in (3-c) despite the linearly preceding *no*, be-

[1] All of our experiments were pre-registered online at `http://aspredicted.org/blind.php?x={xd9cw9, 3xv2du, jd384m, cy6zp6, 2hk4gf, zt73qt, f9pk9f, ab9f3h, yt6pi4}`

cause inside a subject-modifying relative clause is not a valid position for an NPI licensor; we call this a DISTRACTOR position.

(3) a. *__The__ senator that supported the measure has ever found any support from her constituents.
b. __No__ senator that supported the measure has ever found any support from her constituents.
c. *__The__ senator that supported __no__ measure has ever found any support from her constituents.
d. __No__ senator that supported no measure has ever found any support from her constituents.

Learning of NPI licensing conditions by LSTM language models trained on large corpora has previously been investigated by Marvin and Linzen (2018) and Futrell et al. (2018). Futrell et al. found that the language models of both Gulordava et al. (2018) and Jozefowicz et al. (2016) (hereafter called 'Large Data LSTMs') learned a contingency between licensors and NPIs: the NPIs in examples like (3) were lower-surprisal when linearly preceded by negative licensors. However, both papers reported that these models failed to constrain the contingency along the correct structural lines: negative NPI surprisal was decreased at least as much by a preceding negative distractor as by a negative licensor.

Syntactic supervision might plausibly facilitate learning of NPI licensing conditions. We tested this following the method of Futrell et al. (2018), constructing 27 items on the design of in (3), with two variants: one included *ever* and omitting *any*, and one including *any* and omitting *ever*. Figure 1, left panel, shows the results. For the RNNG and the ActionLSTM, negative licensors and distractors alike reduced surprisal of both NPIs ($p < 0.05$ for the RNNG, $p < 0.001$ for the ActionLSTM). For the LSTM, negative licensors and distractors alike reduced surprisal of *ever* (both $p < 0.01$), but not *any*. This may seem surprising as *any* is considerably more frequent than *ever* (123 vs. 727 instances in the training data), but *any*'s non-NPI uses (e.g., *I will eat anything fried*) may complicate its learning.

From Figure 1 it is also apparent that the RNNG and ActionLSTM show signs of stronger NPI licensing effects from negation in the licensor position than in the distractor position, at least for *ever*. To quantify this, we follow Marvin and Linzen (2018) in computing item-mean classification accuracies, with classification being considered correct if the NPI is assigned higher probability in context for (3-b) than for (3-c). Results are shown in Figure 1, right panel. No

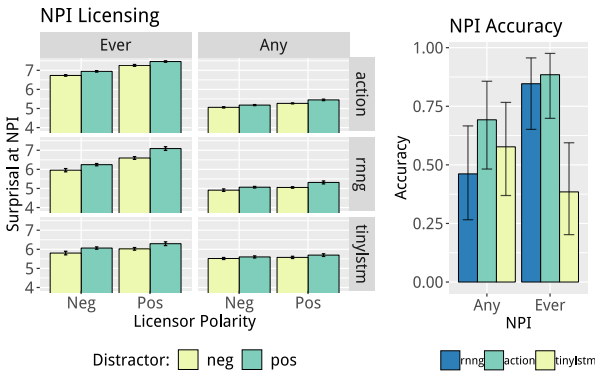

Figure 1: NPI Licensing at left: Y-axis shows surprisal at the NPI, x-axis indicates polarity of the c-commanding licensor, and color indicates distractor polarity. Licensing accuracy at right: Y-axis shows classification accuracy, x-axis indicates the NPI tested, and color indicates the model. Error bars represent 95% binomial confidence intervals.

model is significantly above chance for *any*, but for *ever* the syntactically supervised models perform much better: The RNNG reaches 85% performance, and the ActionLSTM 88%, both significantly above chance ($p < 0.001$ by binomial test for each), and are not significantly different from each other, but both better than the LSTM ($p < 0.01$ for the RNNG/LSTM; $p < 0.001$ for the ActionLSTM/LSTM by Fisher's exact test). To our knowledge this is the first demonstration of a language model learning the licensing conditions for an NPI without direct supervision.

Overall, we find that syntactic supervision facilitates the contingency of NPIs on a negative licensor in context, but is not sufficient for clean generalization of the structural conditions on NPI licensing with the training dataset used here.

4 Filler–Gap Dependencies

The dependency between a FILLER, which is a wh-word such as *who* or *what*, and a GAP, which is an empty syntactic position, is characterized by a number of properties, some of which were tested for large data LSTMs by Wilcox et al. (2018). Here we investigate the effect of syntactic supervision on filler–gap dependency learning. Syntactic annotation of the dependency itself is stripped from the training data (Figure 2), so syntactic supervision can play only an indirect facilitatory role for the models' neural learning mechanisms.

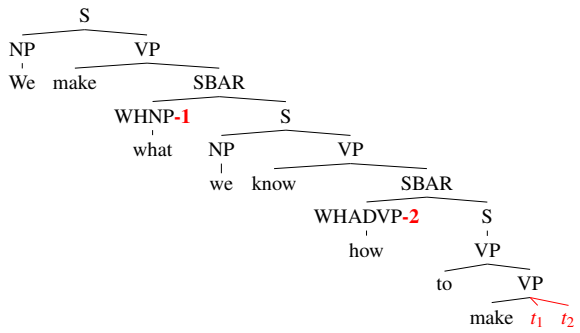

Figure 2: Example of filler–gap dependency representation in the Penn Treebank. Non-local dependency annotation indicated in bold, red font is stripped from the training data, so that the RNNG must learn about the filler-gap dependency purely through neural generalization.

Location of Gap	All Fillers	'Who'	'What'
All Positions	13907	1888	660
Subject Position	6632	1510	236
Object Position	2080	12	332
Indirect Object Position	57	0	6

Table 1: Filler—Gap Dependency Statistics for the Penn Treebank training data (used for both models).

4.1 Flexibility of Gap Position

The filler–gap dependency is flexible: a filler can license a gap in any of a number of syntactic positions, including the argument positions of subject, object, and indirect object, as illustrated in (4), as well as in other positions (e.g. the adjunct position for *how* in Figure 2).

(4) a. I know who _ introduced the accountant to the guests after lunch.
b. I know who the CEO introduced _ to the guests after lunch.
c. I know who the CEO introduced the accountant to _ after lunch.

These gap positions differ in frequency, however (Table 1): the majority (63.1%) are in some argument structure position, of which the vast majority (75.6%) are subject position (mostly subject-extracted relative clauses), 23.7% are object position, and 0.7% are indirect object position.

Using the wh-interaction measure described in Section 2.2, Wilcox et al. (2018) showed that large-data LSTMs learn filler–gap dependencies for all three argument positions, with the size of the wh-interaction generally largest for subject gaps and smallest for indirect-object gaps. Table 1 suggests that this gradation may reflect frequency of learning signal, with the dependency being learned more robustly the more frequent the extraction type. We applied the same method, adapting Wilcox et al.'s items to the smaller training dataset. The results can be seen in the upper-left panel of Figure 3.

All three models learn the filler-gap dependency for subject and object positions, and there is suggestive but inconclusive evidence for learning in the rare indirect object position. We see stronger dependency learning for more frequent gap types, as was found for large data LSTMs, and the supervised models show a much stronger wh-licensing effect than the LSTM.

4.2 Syntactic Hierarchy

As with NPIs, the filler–gap dependency is subject to a number of hierarchical, structural constraints. The most basic of these constraints is that the filler must be "above" the gap in the appropriate structural sense (to a first approximation, the filler must *c*-command the gap, though see e.g. **?** for qualifications). Hence *who* in (5-a) is a legitimate extraction from the relative clause, but (5-b) is ungrammatical as the gap is in the matrix clause, above the filler.

(5) a. The policeman who the criminal shot _ with his gun shocked the jury during the trial.
b. *The policeman who the criminal shot the politician with his gun shocked _ during the trial.

A model that properly generalizes this constraint on the filler–gap dependency should *not* show a wh-interaction for cases like (5-b): an undischarged *who* filler should not make the matrix-clause gap particularly more expected. As far as we are aware, no prior work has investigated this property of the filler–gap dependency in language models; we do so here. Because the context in (5) does not allow for an immediate *that* clause initiation for the –FILLER condition as in (1), we instantiate this condition by contrasting the +FILLER,+GAP condition of (5-b) with the variants in (6), where the *who* filler is immediately discharged as the RC verb's extracted subject:

(6) a. *The policeman who _ knows that the criminal shot the politician with his gun shocked _ during the trial. -FILLER,+GAP
b. *The policeman who the criminal shot the politician with his gun shocked the jury during the trial. +FILLER,–GAP
c. The policeman who _ knows that the criminal shot the politician with his gun shocked the jury during the trial. -FILLER,–GAP

We created 22 items following the templates of (5-a) (*Subject* condition) and (5-b) (*Matrix* condition); results are shown in the top-right panel of 3. The supervised models show a large

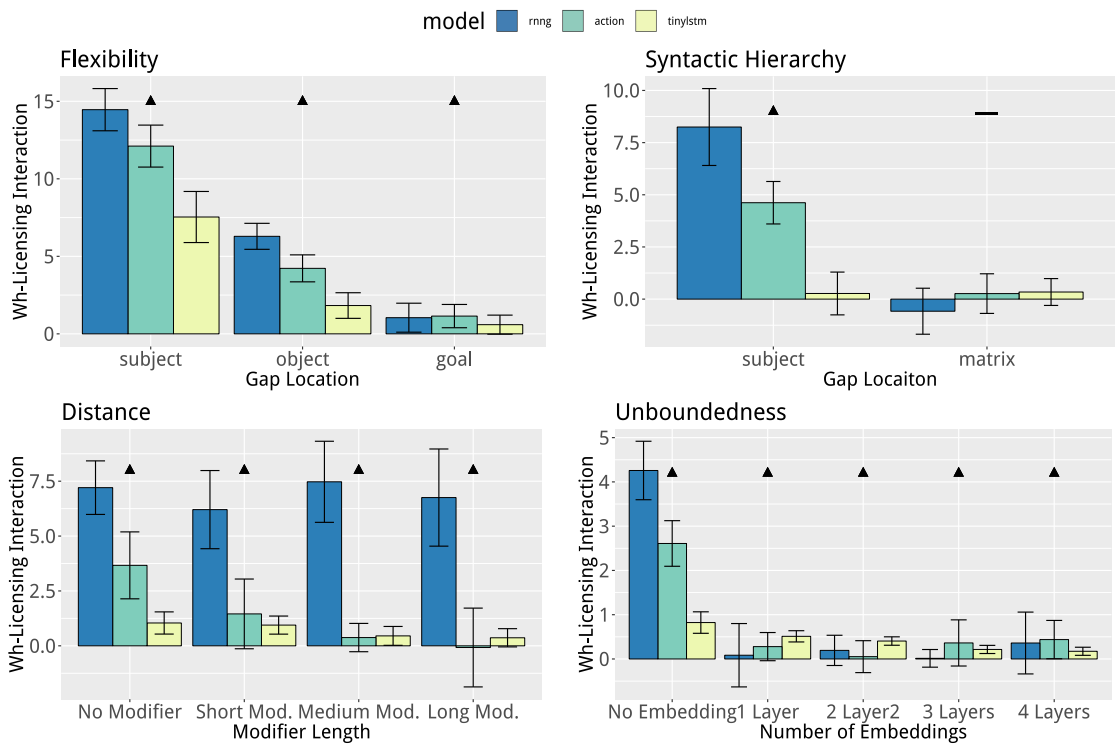

Figure 3: Model results for the basic properties of filler–gap licensing. "▲" indicates grammatical conditions in which models should display strong wh-licensing interaction, "–" indicates ungrammatical conditions in which models should display reduced wh-licensing interaction. The RNNG model significantly outperforms the LSTM model in 8/13 grammatical cases; the ActionLSTM model outperforms the LSTM model in 5/13 cases; and the RNNG outperforms the ActionLSTM model in 6/13 cases where strong licensing is expected.

wh-licensing interaction effect for a gap inside the subject-modifying relative clause—with the RNNG demonstrating more licensing interaction than the ActionLSTM—and neither model inappropriately generalizes this licensing effect to a matrix-clause gap. The LSTM shows no wh-licensing effects in either position, suggesting that syntactic supervision facilitates appropriately generalized filler-gap dependencies for subject-modifying relative clauses.[2]

4.3 Robustness to Intervening Material

For a model that learns human-like syntactic generalizations and maintains accurate phrase-like representations throughout a string, filler–gap dependencies should be robust to linearly intervening material that does not change the tree-structural relationship between the filler and the gap. Wilcox et al. (2018) found that the large-data RNNs described earlier exhibit a robust wh-interaction of this type, by introducing an optional postnominal modifier between filler and gap to sentence templates like (7), with no modification (7-a), short (3–5 word) modifiers (7-b), medium (6–8 word) modifiers (7-c), and long (8–12 word) modifiers (7-d).

(7) a. I know what your friend gave __ to Alex last weekend.

b. I know what your friend in the hat gave __ to Alex last weekend.

c. I know what your friend who you ate lunch with yesterday gave __ to Alex last weekend.

d. I know what your friend who recently took you on a walking tour of the city gave __ to Alex last weekend.

We adapted their materials for the small training dataset and tested our three models; results are shown in 3, bottom-left panel. The RNNG shows a robust licensing interaction that does not diminish with additional intervening material (all $d > 1.3$). The LSTM shows smaller wh-licensing interactions across the board; these are still substantial in the *No Modifier* and *Short Modifier* conditions ($d = 0.88, d = 0.98$, respectively), but are smaller in the *Medium Modifier* and *Long Modifier* conditions ($d = 0.45, d = 0.37$ respectively), suggesting less robustness to intervening material. The

[2]Results for the Larger Data LSTM models for the Hierarchy and Unboundedness experiments presented here can be found in the appendix.

ActionLSTM shows strong interactions in the *No Modifier* condition ($d = 1.02$), but weak interaction once any modifying material is introduced ($d < 0.4$ in all other conditions). This result is significant, as it indicates that RNNG is able to leverage the structural locality afforded by the neural stack to maintain robust gap expectancy.

4.4 Unboundedness

For humans, filler–gap dependencies are not only robust to linearly intervening material that does not change their tree-structural relationship, they can be STRUCTURALLY NON-LOCAL as well, propagating through intervening syntactic structures (subject to constraints examined in Section 5). For example, a filler can be extracted from multiply-nested complement clauses as in (8-b):

(8) a. I know who your aunt insulted __ at the party.
b. I know who the chauffeur said [S the hostess believed [S the butler reported [S her friend thinks your aunt insulted __ at the party.]]]

Humans show sensitivity to a single layer of sentential embedding when processing filler—gap dependencies in an offline 'complexity rating' task (Phillips et al., 2005). This may due to the relative frequency of single versus doubly-embedded filler—gap dependencies. In our training data there were 13,907 examples of filler—gap dependencies, however only 758 examples that spanned two layers of sentential embedding and 19 that spanned three layers. There were no instances of filler—gap dependencies spanning over more than three sentential embeddings, as in (8-b).

The unboundedness of filler–gap dependencies has not previously been tested for contemporary language models. To do this, we constructed 22 test items like (8), varying embedding depth within-item between zero, one, two, three, and four levels, and measured the resulting licensing interactions. The results are in Figure 3, bottom-right panel. No model's filler–gap dependency is perfectly robust to clausal embedding. The LSTM's wh-licensing interaction starts out small and diminishes with embedding depth. The RNNG and ActionLSTM show strong wh-licensing interaction in the unembedded condition but no significant wh-licensing interaction after even a single layer of embedding. Since these experimental materials are new, we also tested the large-data LSTMs on them, which exhibited much larger and more robust filler–dependency effects (Appendix B). Hence the syntactic supervision explored here is not sufficient to guarantee that learned filler–gap dependencies can be structurally unbounded.

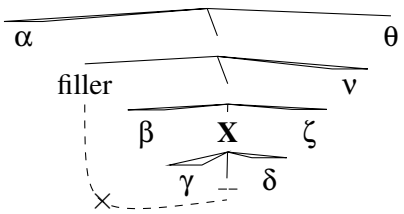

Figure 4: Anatomy of an island constraint. If node **X** is an island, then a filler outside **X** cannot associate with a gap inside **X**. For our analyses, successful learning of an island constraint implies that we should *not* see a wh-licensing interaction at the first part of the material δ immediately following the potential gap site.

5 Island Constraints

A crucial exception to the flexibility and unboundedness of filler–gap dependencies is that ISLAND CONSTRAINTS prevent association of a filler and a gap through certain types of syntactic nodes, illustrated in Figure 4 (Ross, 1967). Contemporary theories variously attribute island effects to grammatical rules, incremental processing considerations, or discourse-structural factors (Ambridge and Goldberg, 2008; Hofmeister and Sag, 2010; Sprouse and Hornstein, 2013). In our setting, a language model is sensitive to an island constraint if it *fails* to show a wh-licensing interaction between a filler and a gap that cross an island. Wilcox et al. (2018) found evidence that large-data LSTMs are sensitive to some island constraints (although see Chowdhury and Zamparelli (2018) for a contrasting view), but not to others. Here we investigate whether LSTMs would learn these from smaller training datasets, and if an RNNG's syntactic supervision provides a learning advantage for island constraints. In this section we measure the wh-licensing interaction in the material immediately following the potential gap site, which is guaranteed to implicate the model's (lack of) expectation for a gap inside the island, rather than throughout the entire embedded clause, which also implicates filler-driven expectations after the end of the island.

5.1 Adjunct Islands

Adjunct clauses block the filler–gap dependency. Wilcox et al. (2018) found evidence that large-

data LSTMs are sensitive to adjunct islands, as evidenced by attenuated and often fully eliminated wh-licensing interactions for materials like (9-b)–(9-c) relative to (9-a) below. (In this and the subsequent subsections, the post-gap material used for wh-interaction computation is in **bold**.)

(9) a. The director discovered what the robbers stole __ **last night**. [OBJECT]
b. *The director discovered what the security guard slept while the robbers stole __ **last night**. [ADJ-BACK]
c. *The director discovered what, while the robbers stole __ **last night**, the security guard slept. [ADJ-FRONT]

We adapted these materials; results are in Figure 5, upper-left panel. The RNNG shows a strong licensing interaction in the baseline main-clause object extraction position, but no licensing interaction for a gap in an adjunct either at the back or front of the main clause. Because RNNGs failed our test for unboundedness of filler–gap dependency, however (Section 4.4), this result is inconclusive as to whether anything corresponding to an island constraint is learned. The LSTM and the ActionLSTM show no sign of filler–gap dependency attenuation from adjunct islands, in contrast to previous findings using the LSTM architecture on much larger training datasets.

5.2 Wh Islands

Embedded sentences introduced by *wh-* words are also islands; hence, (10-c) is anomalous but (10-a) and (10-b) are not.

(10)a. I know what the guide said the lion devoured __ **yesterday**. [NULL COMP]
b. I know what the guide said that the lion devoured __ **yesterday**. [THAT COMP]
c. *I know what the guide said whether the lion devoured __ **yesterday**. [WH- COMP]

Wilcox et al. (2018) found that the large-data LSTMs learned this island constraint: the wh-licensing interaction was eliminated or severely attenuated for the WH-COMPlementizer variant but not for the other variants. Results for our three models are in Figure 5, top-right panel. These materials paint a slightly more optimistic picture than the results of Section 4.4 for the RNNG's ability to propagate a gap expectation from a filler down one level of clausal embedding. However, no models show an appreciable attenuation in the WH- COMP condition that would suggest an island constraint-like generalization.

5.3 Complex Noun-Phrase Islands

Extractions from within clauses dominated by a lexical head noun are unacceptable; this is the Complex Noun Phrase Constraint. For example, (11-b) and (11-c) are unacceptable object extractions compared with (11-a); the same acceptability pattern holds for subject extractions.

(11)a. I know what the collector bought __ **last week**. [ARGUMENT extraction]
b. *I know what the collector bought the painting which depicted __ **last week**. [WH- COMPLEX NP]
c. *I know what the collector bought the painting that depicted __ **last week**. [THAT- COMPLEX NP]

Wilcox et al. (2018) found that large-data LSTM behavior reflected this island constraint, with attenuated wh-licensing interactions for complex NPs like (11-b)–(11-c) and for analogous complex NPs involving subject extractions. Our results for adaptations of their materials are shown in Figure 5, bottom-left panel. All three models show attenuated wh-licensing interactions inside complex NPs in subject position, with the licensing interaction in the grammatical ARGUMENT STRUCTURE position greatest for the RNNG and ActionLSTM. This may be taken as an indication of Complex NP Constraint-like learning, but is inconclusive due to the models' general failure to propagate gap expectations into embedded clauses (Section 4.4).

5.4 Subject Islands

Prepositional phrases attaching to subjects are islands: this is the Subject Constraint, and accounts for the unacceptability of (12-d) compared to (12-c) (Huang, 1998).

(12)a. I know what the collector bought __ **yesterday**. [OBJ VERBAL-ARG]
b. I know what the collector bought a painting of __ **yesterday**. [OBJ PREP-ARG]
c. I know what __ **sold** for a high price at auction. [SUBJ VERBAL-ARG]
d. *I know what a painting of __ **sold** for a high price at auction. [SUBJ PREP-ARG]

Wilcox et al. (2018) found that the wh-licensing interactions of large-data LSTMs fail to distinguish between subject-modifying PPs, which cannot be extracted from, and object-modifying PPs, which can. Our results for adaptations of their materials can be seen in Figure 5, bottom right panel. The syntactically supervised models show a significant decrease between the verbal argument and prepositional argument conditions in subject position ($p < 0.001$ for RNNG; $p < 0.01$ for ActionLSTM), and no significant difference between the two conditions in object position (however, note that the licensing in object position is significantly less than the licensing in the grammatical, *Verbal Argument Subject* position, following the pattern in 4.1). LSTMs fare worse, showing a clear wh-

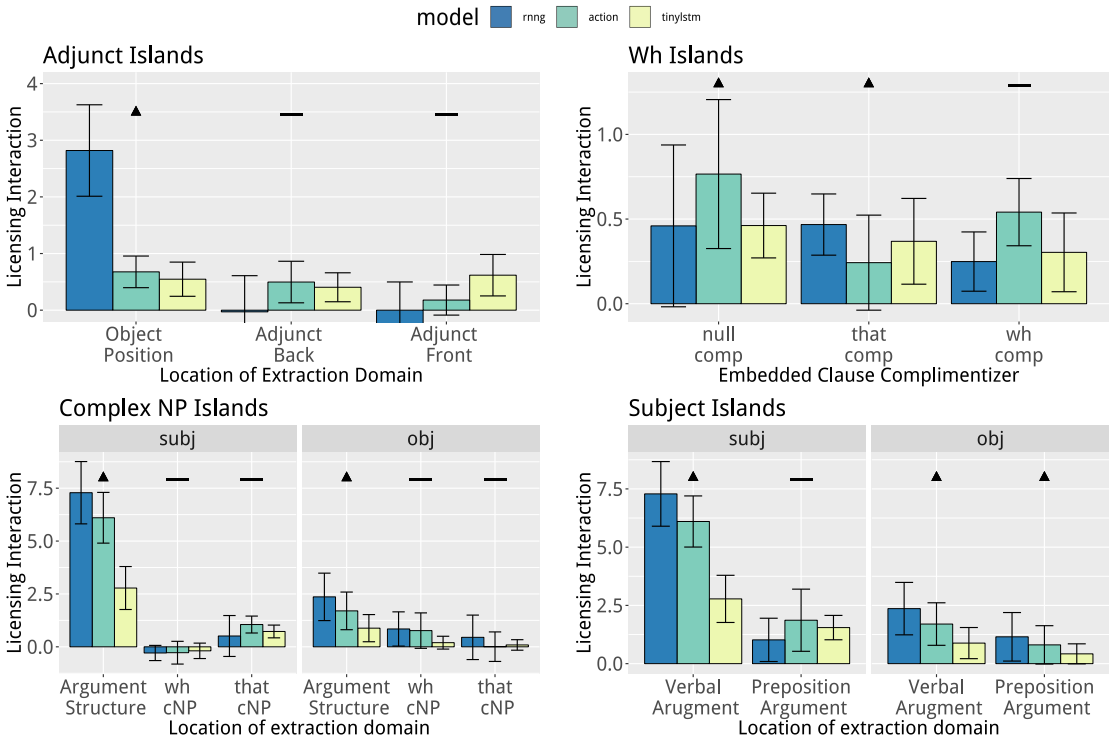

Figure 5: Model results for Syntactic Islands. "▲" indicates grammatical conditions in which models should display strong wh-licensing interaction, "–" indicates ungrammatical conditions in which models should display reduced wh-licensing interaction.

licensing interaction for subject-modifying PPs, which should be islands, and no wh-licensing interaction for object-modifying PPs.

6 Conclusion

In this paper we have argued that structural supervision provides advantages over purely string-based training of neural language models in acquiring more human-like generalizations about non-local grammatical dependencies. We have also demonstrated how the neural compositionality of the RNNG architecture can provide even further advantages, especially at maintaining expectations into structurally-local but linearly distant material. We compared RNNG, ActionLSTM and LSTM models using recently developed controlled experimental materials, and developed additional experimental materials to further test several characteristics of grammatical dependency learning for neural language models (Sections 4.2, 4.4). We found advantages for syntactic supervision in learning conditions for **Negative Polarity Item licensing** and a majority of tests involving **filler–gap dependencies**, showing particularly strong wh-licensing effects in tree-structurally-local contexts. On basic filler—gap dependency properties the RNNG significantly outperformed the LSTM in 8/13 and the ActionLSTM outperformed the LSTM on 5/13 cases where strong licensing interaction was expected. While the RNNG, and to some extent the ActionLSTM, exhibited more humanlike behavior than the LSTM for a number of **Island Constraints**, the tests were inconclusive due to the models' failure to propagate gap expectation into embedded clauses: island-like behavior may merely be sensitivity to general syntactic complexity, not the highly-specific syntactic arrangements that constitute the family of island constructions. Thus, major-category supervision does not provide enough information for the neural component to learn fully robust and human-like filler—gap dependencies from 1-million words alone. However, for some dependencies tested (i.e. NPIs) structural supervision on 1 million words provides better outcomes than even large-data LSTMs. Scaling the gains derived from structural supervision is a challenge for data-scarce NLP and is the basis for future work.

Acknowledgments

This work was supported by the MIT-IBM Watson AI Lab.

References

Ben Ambridge and Adele E Goldberg. 2008. The island status of clausal complements: Evidence in favor of an information structure explanation. *Cognitive Linguistics*, 19(3):357–389.

R Harald Baayen, Douglas J Davidson, and Douglas M Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of memory and language*, 59(4):390–412.

Eugene Charniak et al. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336.

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.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proc. of NAACL*.

Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. Rnns as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. *arXiv preprint arXiv:1803.11138*.

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. Association for Computational Linguistics.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Philip Hofmeister and Ivan A Sag. 2010. Cognitive constraints and island effects. *Language*, 86(2):366.

C-T James Huang. 1998. *Logical relations in Chinese and the theory of grammar*. Taylor & Francis.

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

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A Smith. 2016. What do recurrent neural network grammars learn about syntax? *arXiv preprint arXiv:1611.05774*.

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.

William Ladusaw. 1979. *Polarity Sensitivity as Inherent Scope Relations*. Ph.D. thesis, University of Texas at Austin.

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

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

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. *arXiv preprint arXiv:1808.09031*.

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.

Colin Phillips, Nina Kazanina, and Shani H Abada. 2005. Erp effects of the processing of syntactic long-distance dependencies. *Cognitive Brain Research*, 22(3):407–428.

John Robert Ross. 1967. Constraints on variables in syntax.

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

Jon Sprouse and Norbert Hornstein. 2013. *Experimental syntax and island effects*. Cambridge University Press.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. *arXiv preprint arXiv:1707.08976*.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do rnn language models learn about filler-gap dependencies? *arXiv preprint arXiv:1809.00042*.

Appendix

We present results for two large data LSTM models on novel experiments described in the paper. The two models tested here are the 'BIG LSTM + CNN Inputs' from Jozefowicz et al. (2016)

(the 'google' model) and the highest-performing model presented in the supplementary materials of Gulordava et al. (2018) (the 'Gulordava' model). Both models where shown in (Wilcox et al., 2018) to represent filler—gap dependencies and some island constraints.

A Syntactic Hierarchy

We tested the two large LSTM models using our stimuli from the syntactic hierarchy experiment and measured the wh-licensing interaction across the entire embedded clause. The results of this experiment can be seen in Figure 6. Both models show significant licensing interaction in the grammatical *Subject* condition ($p < 0.001$), and a significant reduction in licensing interaction between the *Subject* and *Matrix* conditions ($p < 0.001$ in both models). Additionally, there is a significant licensing interaction in the *Matrix* condition for the Google model, but not so for the Gulordava model.

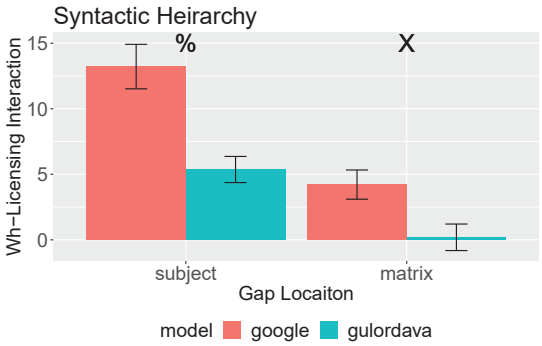

Figure 6: Syntactic Hierarchy. %s indicate conditions where we would expect a strong wh-licensing interaction, Xs where we expect low wh-licensing interaction.

B Unboundedness

We tested the two large LSTM models from Wilcox et al. (2018) following the stimuli from our unboundedness experiment, with two variants, one that included gaps in *Object* position and one that included gaps in indirect object or *Goal* position. The results can be seen in Figure 7. For the Google model in *Object* position, we find a significant reduction of wh-licensing interaction across more than three layers of embedding ($p < 0.001$). For the Gulordava model, we find a significant reduction in wh-licensing interaction after only one layer of embedding ($p < 0.001$). In the *Goal* position: For the Google model, we find a significant reduction in licensing interaction after two layers of embedding ($p < 0.05$ for 2 layers, $p < 0.001$ for 3-4 layers). For the Gulordava model, we find no significant licensing interaction after one layer of embedding. These results indicate the larger LSTMs are able to thread gap expectation through embedded clauses.

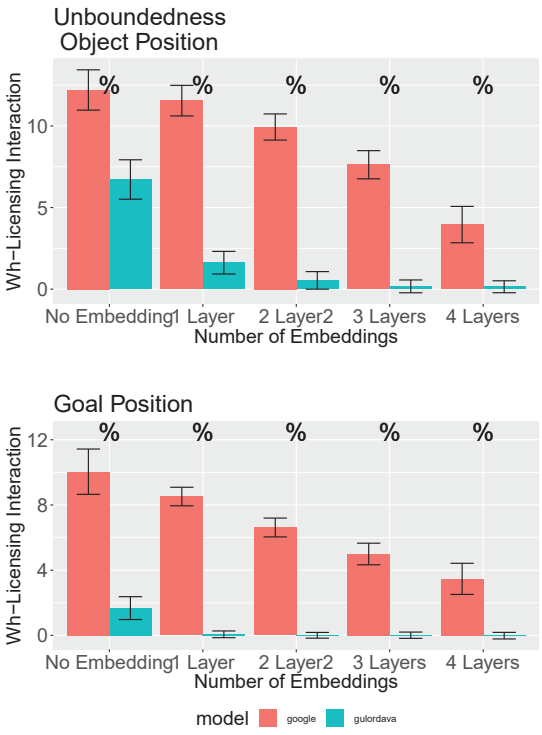

Figure 7: Unboundedness, %s indicate conditions where we expect a strong wh-licensing interaction.