Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies

Tal Linzen[1,2] **Emmanuel Dupoux**[1]
LSCP[1] & IJN[2], CNRS,
EHESS and ENS, PSL Research University
{tal.linzen,
emmanuel.dupoux}@ens.fr

Yoav Goldberg
Computer Science Department
Bar Ilan University
yoav.goldberg@gmail.com

Abstract

The success of long short-term memory (LSTM) neural networks in language processing is typically attributed to their ability to capture long-distance statistical regularities. Linguistic regularities are often sensitive to syntactic structure; can such dependencies be captured by LSTMs, which do not have explicit structural representations? We begin addressing this question using number agreement in English subject-verb dependencies. We probe the architecture's grammatical competence both using training objectives with an explicit grammatical target (number prediction, grammaticality judgments) and using language models. In the strongly supervised settings, the LSTM achieved very high overall accuracy (less than 1% errors), but errors increased when sequential and structural information conflicted. The frequency of such errors rose sharply in the language-modeling setting. We conclude that LSTMs can capture a non-trivial amount of grammatical structure given targeted supervision, but stronger architectures may be required to further reduce errors; furthermore, the language modeling signal is insufficient for capturing syntax-sensitive dependencies, and should be supplemented with more direct supervision if such dependencies need to be captured.

1 Introduction

Recurrent neural networks (RNNs) are highly effective models of sequential data (Elman, 1990). The rapid adoption of RNNs in NLP systems in recent years, in particular of RNNs with gating mechanisms such as long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997) or gated recurrent units (GRU) (Cho et al., 2014), has led to significant gains in language modeling (Mikolov et al., 2010; Sundermeyer et al., 2012), parsing (Vinyals et al., 2015; Kiperwasser and Goldberg, 2016; Dyer et al., 2016), machine translation (Bahdanau et al., 2015) and other tasks.

The effectiveness of RNNs[1] is attributed to their ability to capture statistical contingencies that may span an arbitrary number of words. The word *France*, for example, is more likely to occur somewhere in a sentence that begins with *Paris* than in a sentence that begins with *Penguins*. The fact that an arbitrary number of words can intervene between the mutually predictive words implies that they cannot be captured by models with a fixed window such as n-gram models, but can in principle be captured by RNNs, which do not have an architecturally fixed limit on dependency length.

RNNs are sequence models: they do not explicitly incorporate syntactic structure. Indeed, many word co-occurrence statistics can be captured by treating the sentence as an unstructured list of words (*Paris-France*); it is therefore unsurprising that RNNs can learn them well. Other dependencies, however, are sensitive to the syntactic structure of the sentence (Chomsky, 1965; Everaert et al., 2015). To what extent can RNNs learn to model such phenomena based only on sequential cues?

Previous research has shown that RNNs (in particular LSTMs) can learn artificial context-free languages (Gers and Schmidhuber, 2001) as well as nesting and

[1] In this work we use the term RNN to refer to the entire class of sequential recurrent neural networks. Instances of the class include long short-term memory networks (LSTM) and the Simple Recurrent Network (SRN) due to Elman (1990).

Transactions of the Association for Computational Linguistics, vol. 4, pp. 521–535, 2016. Action Editor: Hinrich Schütze.
Submission batch: 1/2016; Revision batch: 9/2016; Published 12/2016.

indentation in a programming language (Karpathy et al., 2016). The goal of the present work is to probe their ability to learn *natural language* hierarchical (syntactic) structures from a corpus without syntactic annotations. As a first step, we focus on a particular dependency that is commonly regarded as evidence for hierarchical structure in human language: English subject-verb agreement, the phenomenon in which the form of a verb depends on whether the subject is singular or plural (*the kids play* but *the kid plays*; see additional details in Section 2). If an RNN-based model succeeded in learning this dependency, that would indicate that it can learn to approximate or even faithfully implement syntactic structure.

Our main interest is in whether LSTMs have the *capacity* to learn structural dependencies from a natural corpus. We therefore begin by addressing this question under the most favorable conditions: training with explicit supervision. In the setting with the strongest supervision, which we refer to as the number prediction task, we train it directly on the task of guessing the number of a verb based on the words that preceded it (Sections 3 and 4). We further experiment with a grammaticality judgment training objective, in which we provide the model with full sentences annotated as to whether or not they violate subject-verb number agreement, without an indication of the locus of the violation (Section 5). Finally, we trained the model without any grammatical supervision, using a language modeling objective (predicting the next word).

Our quantitative results (Section 4) and qualitative analysis (Section 7) indicate that most naturally occurring agreement cases in the Wikipedia corpus are easy: they can be resolved without syntactic information, based only on the sequence of nouns preceding the verb. This leads to high overall accuracy in all models. Most of our experiments focus on the supervised number prediction model. The accuracy of this model was lower on harder cases, which require the model to encode or approximate structural information; nevertheless, it succeeded in recovering the majority of agreement cases even when four nouns of the opposite number intervened between the subject and the verb (17% errors). Baseline models failed spectacularly on these hard cases, performing far below chance levels. Fine-grained analysis revealed that mistakes are much more common when no overt cues to syntactic structure (in particular function words) are available, as is the case in noun-noun compounds and reduced relative clauses. This indicates that the number prediction model indeed managed to capture a decent amount of syntactic knowledge, but was overly reliant on function words.

Error rates increased only mildly when we switched to more indirect supervision consisting only of sentence-level grammaticality annotations without an indication of the crucial verb. By contrast, the language model trained without explicit grammatical supervision performed worse than chance on the harder agreement prediction cases. Even a state-of-the-art large-scale language model (Jozefowicz et al., 2016) was highly sensitive to recent but structurally irrelevant nouns, making more than five times as many mistakes as the number prediction model on these harder cases. These results suggest that explicit supervision is necessary for learning the agreement dependency using this architecture, limiting its plausibility as a model of child language acquisition (Elman, 1990). From a more applied perspective, this result suggests that for tasks in which it is desirable to capture syntactic dependencies (e.g., machine translation or language generation), language modeling objectives should be supplemented by supervision signals that directly capture the desired behavior.

2 Background: Subject-Verb Agreement as Evidence for Syntactic Structure

The form of an English third-person present tense verb depends on whether the head of the *syntactic subject* is plural or singular:[2]

(1) a. The **key is** on the table.
b. *The **key are** on the table.
c. *The **keys is** on the table.
d. The **keys are** on the table.

While in these examples the subject's head is adjacent to the verb, in general the two can be separated by some sentential material:[3]

[2] Identifying the head of the subject is typically straightforward. In what follows we will use the shorthand "the subject" to refer to the head of the subject.

[3] In the examples, the subject and the corresponding verb are marked in boldface, agreement attractors are underlined and intervening nouns of the same number as the subject are marked in italics. Asterisks mark unacceptable sentences.

(2) The **keys** to the cabinet **are** on the table.

Given a syntactic parse of the sentence and a verb, it is straightforward to identify the head of the subject that corresponds to that verb, and use that information to determine the number of the verb (Figure 1).

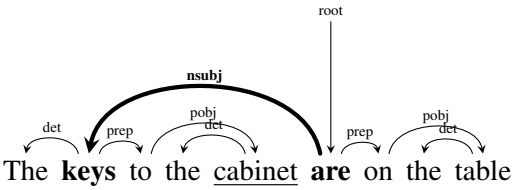

Figure 1: The form of the verb is determined by the head of the subject, which is directly connected to it via an *nsubj* edge. Other nouns that intervene between the head of the subject and the verb (here *cabinet* is such a noun) are irrelevant for determining the form of the verb and need to be ignored.

By contrast, models that are insensitive to structure may run into substantial difficulties capturing this dependency. One potential issue is that there is no limit to the complexity of the subject NP, and any number of sentence-level modifiers and parentheticals—and therefore an arbitrary number of words—can appear between the subject and the verb:

(3) The **building** on the far right that's quite old and run down **is** the Kilgore Bank Building.

This property of the dependency entails that it cannot be captured by an n-gram model with a fixed n. RNNs are in principle able to capture dependencies of an unbounded length; however, it is an empirical question whether or not they will learn to do so in practice when trained on a natural corpus.

A more fundamental challenge that the dependency poses for structure-insensitive models is the possibility of *agreement attraction errors* (Bock and Miller, 1991). The correct form in (3) could be selected using simple heuristics such as "agree with the most recent noun", which are readily available to sequence models. In general, however, such heuristics are unreliable, since other nouns can intervene between the subject and the verb in the linear sequence of the sentence. Those intervening nouns can have the same number as the subject, as in (4), or the opposite number as in (5)-(7):

(4) Alluvial **soils** carried in the *floodwaters* **add** nutrients to the floodplains.

(5) The only championship **banners** that are currently displayed within the building **are** for national or NCAA Championships.

(6) The **length** of the forewings **is** 12-13.

(7) Yet the **ratio** of men who survive to the women and children who survive **is** not clear in this story.

Intervening nouns with the opposite number from the subject are called *agreement attractors*. The potential presence of agreement attractors entails that the model must identify the head of the syntactic subject that corresponds to a given verb in order to choose the correct inflected form of that verb.

Given the difficulty in identifying the subject from the linear sequence of the sentence, dependencies such as subject-verb agreement serve as an argument for structured syntactic representations in humans (Everaert et al., 2015); they may challenge models such as RNNs that do not have pre-wired syntactic representations. We note that subject-verb number agreement is only one of a number of structure-sensitive dependencies; other examples include negative polarity items (e.g., *any*) and reflexive pronouns (*herself*). Nonetheless, a model's success in learning subject-verb agreement would be highly suggestive of its ability to master hierarchical structure.

3 The Number Prediction Task

To what extent can a sequence model learn to be sensitive to the hierarchical structure of natural language? To study this question, we propose the *number prediction* task. In this task, the model sees the sentence up to but not including a present-tense verb, e.g.:

(8) The keys to the cabinet ____

It then needs to guess the number of the following verb (a binary choice, either PLURAL or SINGULAR). We examine variations on this task in Section 5.

In order to perform well on this task, the model needs to encode the concepts of *syntactic number* and *syntactic subjecthood*: it needs to learn that some words are singular and others are plural, and to be able to identify the correct subject. As we have illus-

trated in Section 2, correctly identifying the subject that corresponds to a particular verb often requires sensitivity to hierarchical syntax.

Data: An appealing property of the number prediction task is that we can generate practically unlimited training and testing examples for this task by querying a corpus for sentences with present-tense verbs, and noting the number of the verb. Importantly, we do not need to correctly identify the subject in order to create a training or test example. We generated a corpus of ∼1.35 million number prediction problems based on Wikipedia, of which ∼121,500 (9%) were used for training, ∼13,500 (1%) for validation, and the remaining ∼1.21 million (90%) were reserved for testing.[4] The large number of test sentences was necessary to ensure that we had a good variety of test sentences representing less common constructions (see Section 4).[5]

Model and baselines: We encode words as one-hot vectors: the model does not have access to the characters that make up the word. Those vectors are then embedded into a 50-dimensional vector space. An LSTM with 50 hidden units reads those embedding vectors in sequence; the state of the LSTM at the end of the sequence is then fed into a logistic regression classifier. The network is trained[6] in an end-to-end fashion, including the word embeddings.[7]

To isolate the effect of syntactic structure, we also consider a baseline which is exposed only to the nouns in the sentence, in the order in which they appeared originally, and is then asked to predict the number of the following verb. The goal of this baseline is to withhold the syntactic information carried by function words, verbs and other parts of speech. We explore two variations on this baseline: one that only receives common nouns (*dogs*, *pipe*), and another that also receives pronouns (*he*) and proper nouns (*France*). We refer to these as the *noun-only baselines*.

4 Number Prediction Results

Overall accuracy: Accuracy was very high overall: the system made an incorrect number prediction only in 0.83% of the dependencies. The noun-only baselines performed significantly worse: 4.2% errors for the common-nouns case and 4.5% errors for the all-nouns case. This suggests that function words, verbs and other syntactically informative elements play an important role in the model's ability to correctly predict the verb's number. However, while the noun-only baselines made more than four times as many mistakes as the number prediction system, their still-low absolute error rate indicates that around 95% of agreement dependencies can be captured based solely on the sequence of nouns preceding the verb. This is perhaps unsurprising: sentences are often short and the verb is often directly adjacent to the subject, making the identification of the subject simple. To gain deeper insight into the syntactic capabilities of the model, then, the rest of this section investigates its performance on more challenging dependencies.[8]

Distance: We first examine whether the network shows evidence of generalizing to dependencies where the subject and the verb are far apart. We focus in this analysis on simpler cases where no nouns intervened between the subject and the verb. As Figure 2a shows, performance did not degrade considerably when the distance between the subject and the verb grew up to 15 words (there were very few longer dependencies). This indicates that the network generalized the dependency from the common distances of 0 and 1 to rare distances of 10 and more.

Agreement attractors: We next examine how the model's error rate was affected by nouns that intervened between the subject and the verb in the linear

[4] We limited our search to sentences that were shorter than 50 words. Whenever a sentence had more than one subject-verb dependency, we selected one of the dependencies at random.

[5] Code and data are available at `http://tallinzen.net/projects/lstm_agreement`.

[6] The network was optimized using Adam (Kingma and Ba, 2015) and early stopping based on validation set error. We trained the number prediction model 20 times with different random initializations, and report accuracy averaged across all runs. The models described in Sections 5 and 6 are based on 10 runs, with the exception of the language model, which is slower to train and was trained once.

[7] The size of the vocabulary was capped at 10000 (after lowercasing). Infrequent words were replaced with their part of speech (Penn Treebank tagset, which explicitly encodes number distinctions); this was the case for 9.6% of all tokens and 7.1% of the subjects.

[8] These properties of the dependencies were identified by parsing the test sentences using the parser described in Goldberg and Nivre (2012).

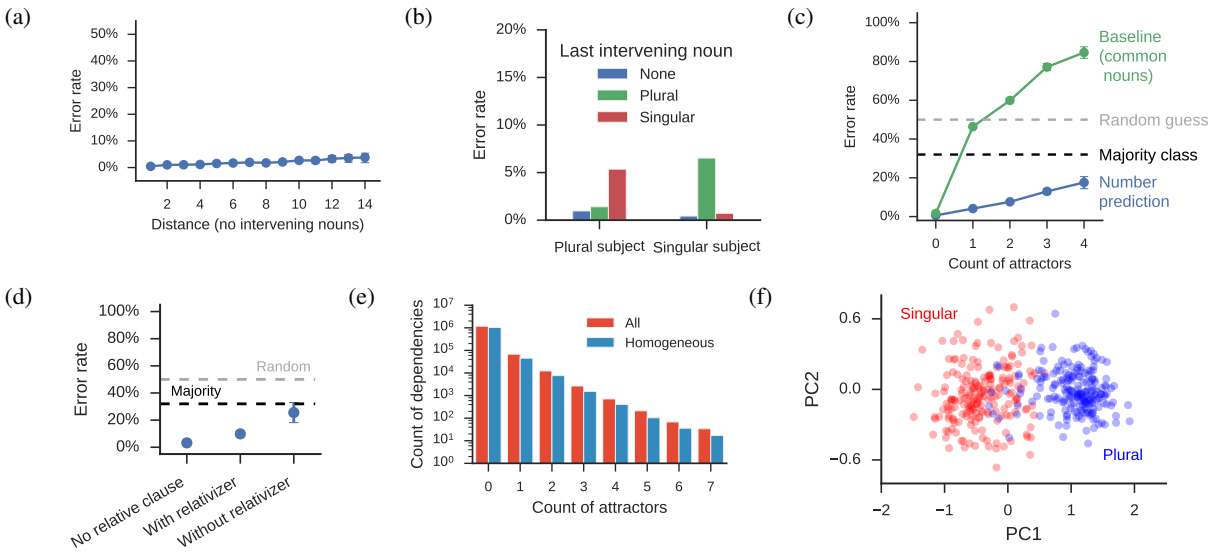

Figure 2: **(a-d)** Error rates of the LSTM number prediction model as a function of: (a) distance between the subject and the verb, in dependencies that have no intervening nouns; (b) presence and number of last intervening noun; (c) count of attractors in dependencies with homogeneous intervention; (d) presence of a relative clause with and without an overt relativizer in dependencies with homogeneous intervention and exactly one attractor. All error bars represent 95% binomial confidence intervals.

(e-f) Additional plots: (e) count of attractors per dependency in the corpus (note that the y-axis is on a log scale); (f) embeddings of singular and plural nouns, projected onto their first two principal components.

order of the sentence. We first focus on whether or not there were any intervening nouns, and if there were, whether the number of the subject differed from the number of the last intervening noun—the type of noun that would trip up the simple heuristic of agreeing with the most recent noun.

As Figure 2b shows, a last intervening noun of the same number as the subject increased error rates only moderately, from 0.4% to 0.7% in singular subjects and from 1% to 1.4% in plural subjects. On the other hand, when the last intervening noun was an agreement attractor, error rates increased by almost an order of magnitude (to 6.5% and 5.4% respectively). Note, however, that even an error rate of 6.5% is quite impressive considering uninformed strategies such as random guessing (50% error rate), always assigning the more common class label (32% error rate, since 32% of the subjects in our corpus are plural) and the number-of-most-recent-noun heuristic (100% error rate). The noun-only LSTM baselines performed much worse in agreement attraction cases, with error rates of 46.4% (common nouns) and 40% (all nouns).

We next tested whether the effect of attractors is cumulative, by focusing on dependencies with multiple attractors. To avoid cases in which the effect of an attractor is offset by an intervening noun with the same number as the subject, we restricted our search to dependencies in which all of the intervening nouns had the same number, which we term *dependencies with homogeneous intervention*. For example, (9) has homogeneous intervention whereas (10) does not:

(9) The **roses** in the vase by the door **are** red.

(10) The **roses** in the vase by the *chairs* **are** red.

Figure 2c shows that error rates increased gradually as more attractors intervened between the subject and the verb. Performance degraded quite slowly, however: even with four attractors the error rate was only 17.6%. As expected, the noun-only baselines performed significantly worse in this setting, reaching an error rate of up to 84% (worse than chance) in the case of four attractors. This confirms that syntactic cues are critical for solving the harder cases.

Relative clauses: We now look in greater detail into the network's performance when the words that intervened between the subject and verb contained a relative clause. Relative clauses with attractors are likely to be fairly challenging, for several reasons. They typically contain a verb that agrees with the attractor, reinforcing the misleading cue to noun number. The attractor is often itself a subject of an irrelevant verb, making a potential "agree with the most recent subject" strategy unreliable. Finally, the existence of a relative clause is sometimes not overtly indicated by a function word (relativizer), as in (11) (for comparison, see the minimally different (12)):

(11) The **landmarks** this article lists here **are** also run-of-the-mill and not notable.

(12) The **landmarks** *that* this article lists here **are** also run-of-the-mill and not notable.

For data sparsity reasons we restricted our attention to dependencies with a single attractor and no other intervening nouns. As Figure 2d shows, attraction errors were more frequent in dependencies with an overt relative clause (9.9% errors) than in dependencies without a relative clause (3.2%), and considerably more frequent when the relative clause was not introduced by an overt relativizer (25%). As in the case of multiple attractors, however, while the model struggled with the more difficult dependencies, its performance was much better than random guessing, and slightly better than a majority-class strategy.

Word representations: We explored the 50-dimensional word representations acquired by the model by performing a principal component analysis. We assigned a part-of-speech (POS) to each word based on the word's most common POS in the corpus. We only considered relatively unambiguous words, in which a single POS accounted for more than 90% of the word's occurrences in the corpus. Figure 2f shows that the first principal component corresponded almost perfectly to the expected number of the noun, suggesting that the model learned the number of specific words very well; recall that the model did not have access during training to noun number annotations or to morphological suffixes such as *-s* that could be used to identify plurals.

Visualizing the network's activations: We start investigating the inner workings of the number prediction network by analyzing its activation in response to particular syntactic constructions. To simplify the analysis, we deviate from our practice in the rest of this paper and use constructed sentences.

We first constructed sets of sentence prefixes based on the following patterns:

(13) **PP:** The toy(s) of the boy(s)...

(14) **RC:** The toy(s) that the boy(s)...

These patterns differ by exactly one function word, which determines the type of the modifier of the main clause subject: a prepositional phrase (PP) in the first sentence and a relative clause (RC) in the second. In PP sentences the correct number of the upcoming verb is determined by the main clause subject *toy(s)*; in RC sentences it is determined by the embedded subject *boy(s)*.

We generated all four versions of each pattern, and repeated the process ten times with different lexical items (*the house(s) of/that the girl(s)*, *the computer(s) of/that the student(s)*, etc.), for a total of 80 sentences. The network made correct number predictions for all 40 PP sentences, but made three errors in RC sentences. We averaged the word-by-word activations across all sets of ten sentences that had the same combination of modifier (PP or RC), first noun number and second noun number. Plots of the activation of all 50 units are provided in the Appendix (Figure 5). Figure 3a highlights a unit (Unit 1) that shows a particularly clear pattern: it tracks the number of the main clause subject throughout the PP modifier; by contrast, it resets when it reaches the relativizer *that* which introduces the RC modifier, and then switches to tracking the number of the embedded subject.

To explore how the network deals with dependencies spanning a larger number of words, we tracked its activation during the processing of the following two sentences:[9]

(15) The houses of/that the man from the office across the street...

The network made the correct prediction for the PP

[9] We simplified this experiment in light of the relative robustness of the first experiment to lexical items and to whether each of the nouns was singular or plural.

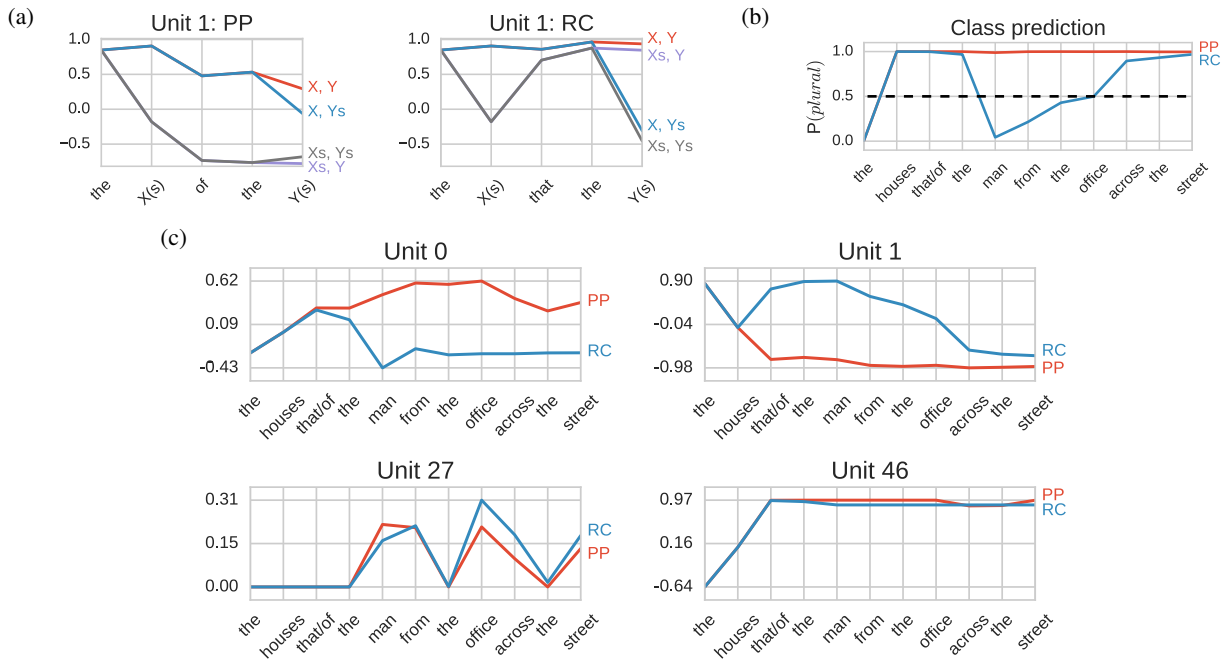

Figure 3: Word-by-word visualization of LSTM activation: (a) a unit that correctly predicts the number of an upcoming verb. This number is determined by the first noun (X) when the modifier is a prepositional phrase (PP) and by the second noun (Y) when it is an object relative clause (RC); (b) the evolution of the predictions in the case of a longer modifier: the predictions correctly diverge at the embedded noun, but then incorrectly converge again; (c) the activation of four representative units over the course of the same sentences.

but not the RC sentence (as before, the correct predictions are PLURAL for PP and SINGULAR for RC). Figure 3b shows that the network begins by making the correct prediction for RC immediately after *that*, but then falters: as the sentence goes on, the resetting effect of *that* diminishes. The activation time courses shown in Figure 3c illustrate that Unit 1, which identified the subject correctly when the prefix was short, gradually forgets that it is in an embedded clause as the prefix grows longer. By contrast, Unit 0 shows a stable capacity to remember the current embedding status. Additional representative units shown in Figure 3c are Unit 46, which consistently stores the number of the main clause subject, and Unit 27, which tracks the number of the most recent noun, resetting at noun phrase boundaries.

While the interpretability of these patterns is encouraging, our analysis only scratches the surface of the rich possibilities of a linguistically-informed analysis of a neural network trained to perform a syntax-sensitive task; we leave a more extensive investigation for future work.

5 Alternative Training Objectives

The number prediction task followed a fully supervised objective, in which the network identifies the number of an upcoming verb based only on the words preceding the verb. This section proposes three objectives that modify some of the goals and assumptions of the number prediction objective (see Table 1 for an overview).

Verb inflection: This objective is similar to number prediction, with one difference: the network receives not only the words leading up to the verb, but also the singular form of the upcoming verb (e.g., *writes*). In practice, then, the network needs to decide between the singular and plural forms of a particular verb (*writes* or *write*). Having access to the semantics of the verb can help the network identify the noun that serves as its subject without using the syntactic subjecthood criteria. For example, in the following sentence:

(16) People from the capital often eat pizza.

Training objective	Sample input	Training signal	Prediction task	Correct answer
Number prediction	*The keys to the cabinet*	PLURAL	SINGULAR/PLURAL?	PLURAL
Verb inflection	*The keys to the cabinet [is/are]*	PLURAL	SINGULAR/PLURAL?	PLURAL
Grammaticality	*The keys to the cabinet are here.*	GRAMMATICAL	GRAMMATICAL/UNGRAMMATICAL?	GRAMMATICAL
Language model	*The keys to the cabinet*	are	$P(are) > P(is)$?	True

Table 1: Examples of the four training objectives and corresponding prediction tasks.

only *people* is a plausible subject for *eat*; the network can use this information to infer that the correct form of the verb is *eat* is rather than *eats*.

This objective is similar to the task that humans face during language production: after the speaker has decided to use a particular verb (e.g., *write*), he or she needs to decide whether its form will be *write* or *writes* (Levelt et al., 1999; Staub, 2009).

Grammaticality judgments: The previous objectives explicitly indicate the location in the sentence in which a verb can appear, giving the network a cue to syntactic clause boundaries. They also explicitly direct the network's attention to the number of the verb. As a form of weaker supervision, we experimented with a grammaticality judgment objective. In this scenario, the network is given a complete sentence, and is asked to judge whether or not it is grammatical.

To train the network, we made half of the examples in our training corpus ungrammatical by flipping the number of the verb.[10] The network read the entire sentence and received a supervision signal at the end. This task is modeled after a common human data collection technique in linguistics (Schütze, 1996), although our training regime is of course very different to the training that humans are exposed to: humans rarely receive ungrammatical sentences labeled as such (Bowerman, 1988).

Language modeling (LM): Finally, we experimented with a word prediction objective, in which the model did not receive any grammatically relevant supervision (Elman, 1990; Elman, 1991). In this scenario, the goal of the network is to predict the next word at each point in every sentence. It receives unlabeled sentences and is not specifically instructed to attend to the number of the verb. In the network that implements this training scenario, RNN activation after each word is fed into a fully connected dense layer followed by a softmax layer over the entire vocabulary.

We evaluate the knowledge that the network has acquired about subject-verb noun agreement using a task similar to the verb inflection task. To perform the task, we compare the probabilities that the model assigns to the two forms of the verb that in fact occurred in the corpus (e.g., *write* and *writes*), and select the form with the higher probability.[11] As this task is not part of the network's training objective, and the model needs to allocate considerable resources to predicting each word in the sentence, we expect the LM to perform worse than the explicitly supervised objectives.

Results: When considering all agreement dependencies, all models achieved error rates below 7% (Figure 4a); as mentioned above, even the noun-only number prediction baselines achieved error rates below 5% on this task. At the same time, there were large differences in accuracy across training objectives. The verb inflection network performed slightly but significantly better than the number prediction one (0.8% compared to 0.83% errors), suggesting that the semantic information carried by the verb is moderately helpful. The grammaticality judgment objective performed somewhat worse, at 2.5% errors, but still outperformed the noun-only baselines by a large margin, showing the capacity of the LSTM architecture to learn syntactic dependencies even given fairly indirect evidence.

[10] In some sentences this will not in fact result in an ungrammatical sentence, e.g. with collective nouns such as *group*, which are compatible with both singular and plural verbs in some dialects of English (Huddleston and Pullum, 2002); those cases appear to be rare.

[11] One could also imagine performing the equivalent of the number prediction task by aggregating LM probability mass over all plural verbs and all singular verbs. This approach may be more severely affected by part-of-speech ambiguous words than the one we adopted; we leave the exploration of this approach to future work.

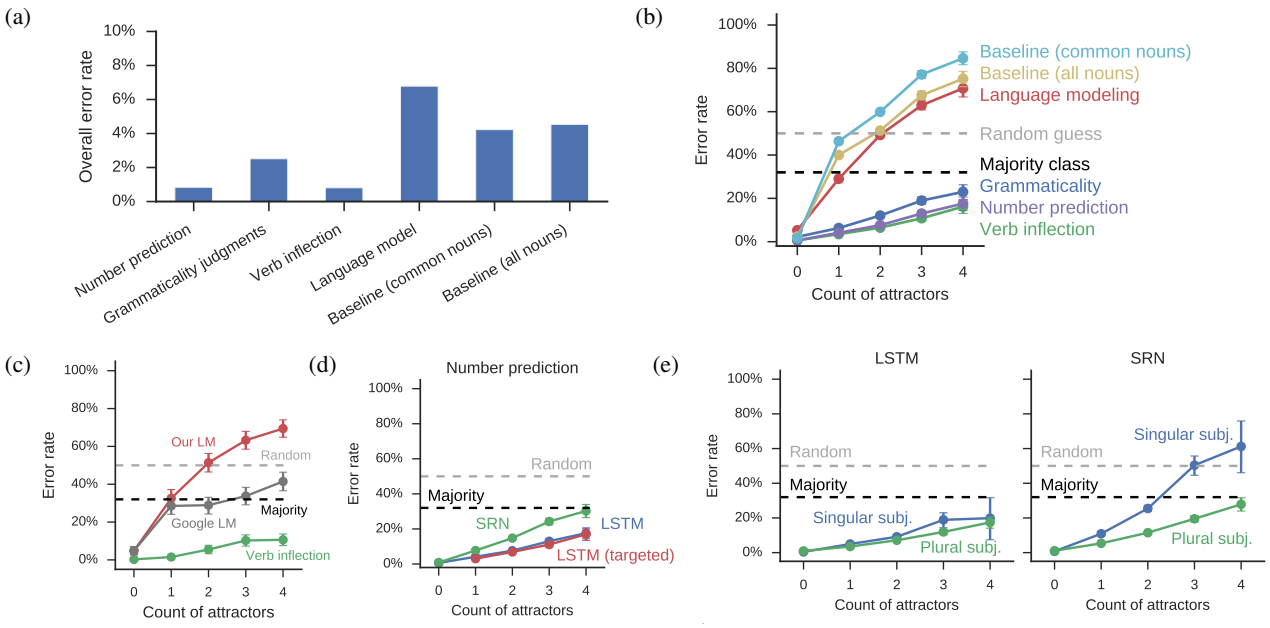

Figure 4: Alternative tasks and additional experiments: (a) overall error rate across tasks (note that the y-axis ends in 10%); (b) effect of count of attractors in homogeneous dependencies across training objectives; (c) comparison of the Google LM (Jozefowicz et al., 2016) to our LM and one of our supervised verb inflection systems, on a sample of sentences; (d) number prediction: effect of count of attractors using SRNs with standard training or LSTM with targeted training; (e) number prediction: difference in error rate between singular and plural subjects across RNN cell types. Error bars represent binomial 95% confidence intervals.

The worst performer was the language model. It made eight times as many errors as the original number prediction network (6.78% compared to 0.83%), and did substantially worse than the noun-only baselines (though recall that the noun-only baselines were still explicitly trained to predict verb number).

The differences across the networks are more striking when we focus on dependencies with agreement attractors (Figure 4b). Here, the language model does worse than chance in the most difficult cases, and only slightly better than the noun-only baselines. The worse-than-chance performance suggests that attractors actively confuse the networks rather than cause them to make a random decision. The other models degrade more gracefully with the number of agreement attractors; overall, the grammaticality judgment objective is somewhat more difficult than the number prediction and verb inflection ones. In summary, we conclude that while the LSTM is capable of learning syntax-sensitive agreement dependencies under various objectives, the language-modeling objective alone is not sufficient for learning such dependencies, and a more direct form of training signal is required.

Comparison to a large-scale language model: One objection to our language modeling result is that our LM faced a much harder objective than our other models—predicting a distribution over 10,000 vocabulary items is certainly harder than binary classification—but was equipped with the same capacity (50-dimensional hidden state and word vectors). Would the performance gap between the LM and the explicitly supervised models close if we increased the capacity of the LM?

We address this question using a very large publicly available LM (Jozefowicz et al., 2016), which we refer to as the Google LM.[12] The Google LM represents the current state-of-the-art in language modeling: it is trained on a billion-word corpus (Chelba et al., 2013), with a vocabulary of 800,000 words. It is based on a two-layer LSTM with 8192 units in each layer, or more than 300 times as many units

[12] https://github.com/tensorflow/models/tree/master/lm_1b

as our LM; at 1.04 billion parameters it has almost 2000 times as many parameters. It is a fine-tuned language model that achieves impressive perplexity scores on common benchmarks, requires a massive infrastructure for training, and pushes the boundaries of what's feasible with current hardware.

We tested the Google LM with the methodology we used to test ours.[13] Due to computational resource limitations, we did not evaluate it on the entire test set, but sampled a random selection of 500 sentences for each count of attractors (testing a single sentence under the Google LM takes around 5 seconds on average). The results are presented in Figure 4c, where they are compared to the performance of the supervised verb inflection system. Despite having an order of magnitude more parameters and significantly larger training data, the Google LM performed poorly compared to the supervised models; even a single attractor led to a sharp increase in error rate to 28.5%, almost as high as our small-scale LM (32.6% on the same sentences). While additional attractors caused milder degradation than in our LM, the performance of the Google LM on sentences with four attractors was still worse than always guessing the majority class (SINGULAR).

In summary, our experiments with the Google LM do not change our conclusions: the contrast between the poor performance of the LMs and the strong performance of the explicitly supervised objectives suggests that direct supervision has a dramatic effect on the model's ability to learn syntax-sensitive dependencies. Given that the Google LM was already trained on several hundred times more data than the number prediction system, it appears unlikely that its relatively poor performance was due to lack of training data.

6 Additional Experiments

Comparison to simple recurrent networks: How much of the success of the network is due to the LSTM cells? We repeated the number prediction experiment with a simple recurrent network (SRN) (Elman, 1990), with the same number of hidden units. The SRN's performance was inferior to the LSTM's, but the average performance for a given number of agreement attractors does not suggest a qualitative difference between the cell types: the SRN makes about twice as many errors as the LSTM across the board (Figure 4d).

Training only on difficult dependencies: Only a small proportion of the dependencies in the corpus had agreement attractors (Figure 2e). Would the network generalize better if dependencies with intervening nouns were emphasized during training? We repeated our number prediction experiment, this time training the model only on dependencies with at least one intervening noun (of any number). We doubled the proportion of training sentences to 20%, since the total size of the corpus was smaller (226K dependencies).

This training regime resulted in a 27% decrease in error rate on dependencies with exactly one attractor (from 4.1% to 3.0%). This decrease is statistically significant, and encouraging given that the total number of dependencies in training was much lower, which complicates the learning of word embeddings. Error rates mildly decreased in dependencies with more attractors as well, suggesting some generalization (Figure 4d). Surprisingly, a similar experiment using the grammaticality judgment task led to a slight *increase* in error rate. While tentative at this point, these results suggest that oversampling difficult training cases may be beneficial; a curriculum progressing from easier to harder dependencies (Elman, 1993) may provide additional gains.

7 Error Analysis

Singular vs. plural subjects: Most of the nouns in English are singular: in our corpus, the fraction of singular subjects is 68%. Agreement attraction errors in humans are much more common when the attractor is plural than when it is singular (Bock and Miller, 1991; Eberhard et al., 2005). Do our models' error rates depend on the number of the subject?

As Figure 2b shows, our LSTM number prediction model makes somewhat more agreement attraction errors with plural than with singular attractors; the difference is statistically significant, but the asymmetry is much less pronounced than in humans. Interestingly, the SRN version of the model does show a large asymmetry, especially as the count of attractors

[13] One technical exception was that we did not replace low-frequency words with their part-of-speech, since the Google LM is a large-vocabulary language model, and does not have parts-of-speech as part of its vocabulary.

increases; with four plural attractors the error rate reaches 60% (Figure 4e).

Qualitative analysis: We manually examined a sample of 200 cases in which the majority of the 20 runs of the number prediction network made the wrong prediction. There were only 8890 such dependencies (about 0.6%). Many of those were straightforward agreement attraction errors; others were difficult to interpret. We mention here three classes of errors that can motivate future experiments.

The networks often misidentified the heads of noun-noun compounds. In (17), for example, the models predict a singular verb even though the number of the subject *conservation refugees* should be determined by its head *refugees*. This suggests that the networks didn't master the structure of English noun-noun compounds.[14]

(17) Conservation **refugees live** in a world colored in shades of gray; limbo.

(18) Information technology (IT) **assets** commonly **hold** large volumes of confidential data.

Some verbs that are ambiguous with plural nouns seem to have been misanalyzed as plural nouns and consequently act as attractors. The models predicted a plural verb in the following two sentences even though neither of them has any plural nouns, possibly because of the ambiguous verbs *drives* and *lands*:

(19) The **ship** that the player drives **has** a very high speed.

(20) It was also to be used to learn if the **area** where the lander lands **is** typical of the surrounding terrain.

Other errors appear to be due to difficulty not in identifying the subject but in determining whether it is plural or singular. In Example (22), in particular, there is very little information in the left context of the subject *5 paragraphs* suggesting that the writer considers it to be singular:

(21) Rabaul-based Japanese **aircraft make** three dive-bombing attacks.

(22) The lead is also rather long; 5 **paragraphs is** pretty lengthy for a 62 kilobyte article.

The last errors point to a limitation of the number prediction task, which jointly evaluates the model's ability to identify the subject and its ability to assign the correct number to noun phrases.

8 Related Work

The majority of NLP work on neural networks evaluates them on their performance in a task such as language modeling or machine translation (Sundermeyer et al., 2012; Bahdanau et al., 2015). These evaluation setups average over many different syntactic constructions, making it difficult to isolate the network's syntactic capabilities.

Other studies have tested the capabilities of RNNs to learn simple artificial languages. Gers and Schmidhuber (2001) showed that LSTMs can learn the context-free language $a^n b^n$, generalizing to ns as high as 1000 even when trained only on $n \in \{1, \ldots, 10\}$. Simple recurrent networks struggled with this language (Rodriguez et al., 1999; Rodriguez, 2001). These results have been recently replicated and extended by Joulin and Mikolov (2015).

Elman (1991) tested an SRN on a miniature language that simulated English relative clauses, and found that the network was only able to learn the language under highly specific circumstances (Elman, 1993), though later work has called some of his conclusions into question (Rohde and Plaut, 1999; Cartling, 2008). Frank et al. (2013) studied the acquisition of anaphora coreference by SRNs, again in a miniature language. Recently, Bowman et al. (2015) tested the ability of LSTMs to learn an artificial language based on propositional logic. As in our study, the performance of the network degraded as the complexity of the test sentences increased.

Karpathy et al. (2016) present analyses and visualization methods for character-level RNNs. Kádár et al. (2016) and Li et al. (2016) suggest visualization techniques for word-level RNNs trained to perform tasks that aren't explicitly syntactic (image captioning and sentiment analysis).

Early work that used neural networks to model

[14] The dependencies are presented as they appeared in the corpus; the predicted number was the opposite of the correct one (e.g., singular in (17), where the original is plural).

grammaticality judgments includes Allen and Seidenberg (1999) and Lawrence et al. (1996). More recently, the connection between grammaticality judgments and the probabilities assigned by a language model was explored by Clark et al. (2013) and Lau et al. (2015). Finally, arguments for evaluating NLP models on a strategically sampled set of dependency types rather than a random sample of sentences have been made in the parsing literature (Rimell et al., 2009; Nivre et al., 2010; Bender et al., 2011).

9 Discussion and Future Work

Neural network architectures are typically evaluated on random samples of naturally occurring sentences, e.g., using perplexity on held-out data in language modeling. Since the majority of natural language sentences are grammatically simple, models can achieve high overall accuracy using flawed heuristics that fail on harder cases. This makes it difficult to distinguish simple but robust sequence models from more expressive architectures (Socher, 2014; Grefenstette et al., 2015; Joulin and Mikolov, 2015). Our work suggests an alternative strategy—evaluation on naturally occurring sentences that are sampled based on their grammatical complexity—which can provide more nuanced tests of language models (Rimell et al., 2009; Bender et al., 2011).

This approach can be extended to the training stage: neural networks can be encouraged to develop more sophisticated generalizations by oversampling grammatically challenging training sentences. We took a first step in this direction when we trained the network only on dependencies with intervening nouns (Section 6). This training regime indeed improved the performance of the network; however, the improvement was quantitative rather than qualitative: there was limited generalization to dependencies that were even more difficult than those encountered in training. Further experiments are needed to establish the efficacy of this method.

A network that has acquired syntactic representations sophisticated enough to handle subject-verb agreement is likely to show improved performance on other structure-sensitive dependencies, including pronoun coreference, quantifier scope and negative polarity items. As such, neural models used in NLP applications may benefit from grammatically sophisticated sentence representations developed in a multi-task learning setup (Caruana, 1998), where the model is trained concurrently on the task of interest and on one of the tasks we proposed in this paper. Of course, grammatical phenomena differ from each other in many ways. The distribution of negative polarity items is highly sensitive to semantic factors (Giannakidou, 2011). Restrictions on unbounded dependencies (Ross, 1967) may require richer syntactic representations than those required for subject-verb dependencies. The extent to which the results of our study will generalize to other constructions and other languages, then, is a matter for empirical research.

Humans occasionally make agreement attraction mistakes during language production (Bock and Miller, 1991) and comprehension (Nicol et al., 1997). These errors persist in human acceptability judgments (Tanner et al., 2014), which parallel our grammaticality judgment task. Cases of grammatical agreement with the nearest rather than structurally relevant constituent have been documented in languages such as Slovenian (Marušič et al., 2007), and have even been argued to be occasionally grammatical in English (Zwicky, 2005). In future work, exploring the relationship between these cases and neural network predictions can shed light on the cognitive plausibility of those networks.

10 Conclusion

LSTMs are sequence models; they do not have built-in hierarchical representations. We have investigated how well they can learn subject-verb agreement, a phenomenon that crucially depends on hierarchical syntactic structure. When provided explicit supervision, LSTMs were able to learn to perform the verb-number agreement task in most cases, although their error rate increased on particularly difficult sentences. We conclude that LSTMs can learn to approximate structure-sensitive dependencies fairly well given explicit supervision, but more expressive architectures may be necessary to eliminate errors altogether. Finally, our results provide evidence that the language modeling objective is not by itself sufficient for learning structure-sensitive dependencies, and suggest that a joint training objective can be used to supplement language models on tasks for which syntax-sensitive dependencies are important.

Acknowledgments

We thank Marco Baroni, Grzegorz Chrupała, Alexander Clark, Sol Lago, Paul Smolensky, Benjamin Spector and Roberto Zamparelli for comments and discussion. This research was supported by the European Research Council (grant ERC-2011-AdG 295810 BOOTPHON), the Agence Nationale pour la Recherche (grants ANR-10-IDEX-0001-02 PSL and ANR-10-LABX-0087 IEC) and the Israeli Science Foundation (grant number 1555/15).

References

Joseph Allen and Mark S. Seidenberg. 1999. The emergence of grammaticality in connectionist networks. In Brian MacWhinney, editor, *Emergentist approaches to language: Proceedings of the 28th Carnegie symposium on cognition*, pages 115–151. Mahwah, NJ: Erlbaum.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference for Learning Representations*.

Emily M. Bender, Dan Flickinger, Stephan Oepen, and Yi Zhang. 2011. Parser evaluation over local and non-local deep dependencies in a large corpus. In *Proceedings of EMNLP*, pages 397–408.

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

Melissa Bowerman. 1988. The "no negative evidence" problem: How do children avoid constructing an overly general grammar? In John A. Hawkins, editor, *Explaining language universals*, pages 73–101. Oxford: Basil Blackwell.

Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. 2015. Tree-structured composition in neural networks without tree-structured architectures. In *Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

Bo Cartling. 2008. On the implicit acquisition of a context-free grammar by a simple recurrent neural network. *Neurocomputing*, 71(7):1527–1537.

Rich Caruana. 1998. Multitask learning. In Sebastian Thrun and Lorien Pratt, editors, *Learning to learn*, pages 95–133. Boston: Kluwer.

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*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT press.

Alexander Clark, Gianluca Giorgolo, and Shalom Lappin. 2013. Statistical representation of grammaticality judgements: The limits of n-gram models. In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 28–36.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and A. Noah Smith. 2016. Recurrent neural network grammars. In *Proceedings of NAACL/HLT*, pages 199–209.

Kathleen M. Eberhard, J. Cooper Cutting, and Kathryn Bock. 2005. Making syntax of sense: Number agreement in sentence production. *Psychological Review*, 112(3):531–559.

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

Jeffrey L. Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7(2-3):195–225.

Jeffrey L. Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Martin B. H. Everaert, Marinus A. C. Huybregts, Noam Chomsky, Robert C. Berwick, and Johan J. Bolhuis. 2015. Structures, not strings: Linguistics as part of the cognitive sciences. *Trends in Cognitive Sciences*, 19(12):729–743.

Robert Frank, Donald Mathis, and William Badecker. 2013. The acquisition of anaphora by simple recurrent networks. *Language Acquisition*, 20(3):181–227.

Felix Gers and Jürgen Schmidhuber. 2001. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340.

Anastasia Giannakidou. 2011. Negative and positive polarity items: Variation, licensing, and compositionality. In Claudia Maienborn, Klaus von Heusinger, and Paul Portner, editors, *Semantics: An international handbook of natural language meaning*. Berlin: Mouton de Gruyter.

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *Proceedings of COLING 2012*, pages 959–976.

Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. 2015. Learning to transduce with unbounded memory. In *Advances in Neural Information Processing Systems*, pages 1828–1836.

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

Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge.

Armand Joulin and Tomas Mikolov. 2015. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Advances in Neural Information Processing Systems*, pages 190–198.

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

Ákos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2016. Representation of linguistic form and function in recurrent neural networks. *arXiv preprint arXiv:1602.08952*.

Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2016. Visualizing and understanding recurrent networks. In *ICLR Workshop*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association of Computational Linguistics*, 4:313–327.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2015. Unsupervised prediction of acceptability judgements. In *Proceedings of ACL/IJCNLP*, pages 1618–1628.

Steve Lawrence, Lee C. Giles, and Santliway Fong. 1996. Can recurrent neural networks learn natural language grammars? In *IEEE International Conference on Neural Networks*, volume 4, pages 1853–1858.

Willem J. M. Levelt, Ardi Roelofs, and Antje S. Meyer. 1999. A theory of lexical access in speech production. *Behavioral and Brain Sciences*, 22(1):1–75.

Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of NAACL-HLT 2016*, pages 681–691.

Franc Marušič, Andrew Nevins, and Amanda Saksida. 2007. Last-conjunct agreement in Slovenian. In *Annual Workshop on Formal Approaches to Slavic Linguistics*, pages 210–227.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048.

Janet L. Nicol, Kenneth I. Forster, and Csaba Veres. 1997. Subject–verb agreement processes in comprehension. *Journal of Memory and Language*, 36(4):569–587.

Joakim Nivre, Laura Rimell, Ryan McDonald, and Carlos Gomez-Rodriguez. 2010. Evaluation of dependency parsers on unbounded dependencies. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 833–841. Association for Computational Linguistics.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of EMNLP*, pages 813–821.

Paul Rodriguez, Janet Wiles, and Jeffrey L. Elman. 1999. A recurrent neural network that learns to count. *Connection Science*, 11(1):5–40.

Paul Rodriguez. 2001. Simple recurrent networks learn context-free and context-sensitive languages by counting. *Neural Computation*, 13(9):2093–2118.

Douglas L. T. Rohde and David C. Plaut. 1999. Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 72(1):67–109.

John Robert Ross. 1967. *Constraints on variables in syntax*. Ph.D. thesis, MIT.

Carson T. Schütze. 1996. *The empirical base of linguistics: Grammaticality judgments and linguistic methodology*. Chicago, IL: University of Chicago Press.

Richard Socher. 2014. *Recursive Deep Learning for Natural Language Processing and Computer Vision*. Ph.D. thesis, Stanford University.

Adrian Staub. 2009. On the interpretation of the number attraction effect: Response time evidence. *Journal of Memory and Language*, 60(2):308–327.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *INTERSPEECH*.

Darren Tanner, Janet Nicol, and Laurel Brehm. 2014. The time-course of feature interference in agreement comprehension: Multiple mechanisms and asymmetrical attraction. *Journal of Memory and Language*, 76:195–215.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2755–2763.

Arnold Zwicky. 2005. Agreement with nearest always bad? `http://itre.cis.upenn.edu/~myl/languagelog/archives/001846.html`.

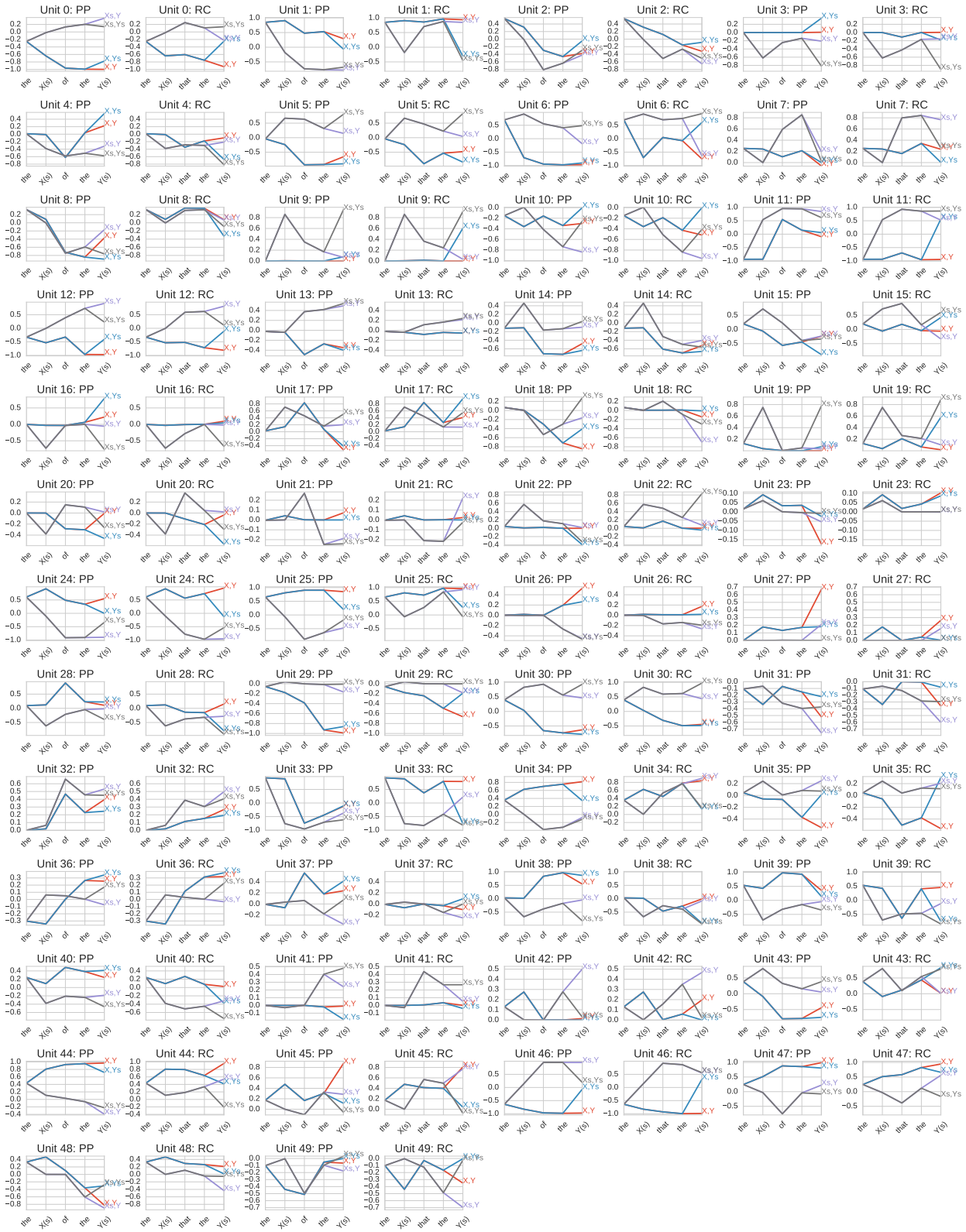

Figure 5: Activation plots for all units (see Figure 3a).