Proceedings of the Society for Computation in Linguistics

Volume 4 Article 38

2021

Can RNNs trained on harder subject-verb agreement instances still perform well on easier ones?

Hritik Bansal
Indian Institute of Technology Delhi, hbansal10n@gmail.com

Gantavya Bhatt
Indian Institute of Technology Delhi, gbhatt2@u.washington.edu

Sumeet Agarwal
Indian Institute of Technology Delhi, sumeet@ee.iitd.ac.in

Recommended Citation

Bansal, Hritik; Bhatt, Gantavya; and Agarwal, Sumeet (2021) "Can RNNs trained on harder subject-verb agreement instances still perform well on easier ones?," *Proceedings of the Society for Computation in Linguistics*: Vol. 4 , Article 38.
DOI: https://doi.org/10.7275/5bnr-wc78
Available at: https://scholarworks.umass.edu/scil/vol4/iss1/38

Can RNNs trained on harder subject-verb agreement instances still perform well on easier ones?

Hritik Bansal*[1], Gantavya Bhatt*[1, 2], and Sumeet Agarwal[1]
[1] Indian Institute of Technology Delhi
[2] University of Washington, Seattle
hbansal10n@gmail.com
gbhatt2@u.washington.edu
sumeet@iitd.ac.in

Introduction: Humans show great ability to generalize linguistic knowledge to sentences that they have never been exposed to, with their limited linguistic experience. An example of this is our ability to perform Subject-Verb Agreement (SVA), which is a feature of syntactic structure in language.[1]

Subject-verb agreement is a phenomenon where the *main subject* agrees in grammatical number with its *associated verb*, oblivious to the presence of any other noun phrase in the sentence. An example is:

1. *The **keys** to the cabinet **is** on the table. [2]
2. The **keys** to the cabinet **are** on the table.

In the above example, the number of the main verb *are* (plural) has to agree with the number of the main subject *keys* (plural). Here, the intervening noun *cabinet* has the opposite number (singular) to that of the main subject. Such intervening nouns are referred to as *agreement attractors* (Bock and Miller, 1991). In natural language sentences, there can be any number of intervening nouns behaving as agreement attractors or non-attractors (nouns with the same number as the main noun).

Previous work (Linzen et al., 2016; Marvin and Linzen, 2018; McCoy et al., 2018; Kuncoro et al., 2019; Noji and Takamura, 2020; Hao, 2020) assessed the ability of RNN Language Models (LMs) to capture syntax-sensitive dependencies. However, it is still not clear if good performance on SVA tasks is necessarily a result of the RNN's ability to capture the underlying syntax, and this is the question we seek to further investigate here. McCoy et al. (2020, 2018) showed that hierarchical bias in the models, as well as the inputs, helps to generalize to unseen sentences. On the other hand, Chaves (2020) and Sennhauser and Berwick (2018) provide evidence that LSTM models are more likely to learn surface-level heuristics, such as agreeing with the most recent noun, than the underlying grammar.

Following McCoy et al. (2018) who show that training on sentences with agreement information increases the probability of good syntactic generalization, we experiment with training RNN models on sentences with *at least one attractor* (Figure 1), to impart additional hierarchical cues compared to a natural data set. We test the hypothesis that if the models under consideration were to capture the correct grammatical structure from syntactically rich input, then they would be able to generalize out-of-distribution (OOD), *i.e.*, when tested on sentences without attractors having been trained solely on sentences with at least one attractor. In our experiments, we compare this setting to the more natural one of models trained on a dataset without any restriction on the number of attractors.

The kind of hypothesis learned by a model is guided by inductive biases. To account for the varying inductive biases that different RNN models might encode, we look at multiple architectures – LSTM, GRU, ONLSTM, and Decay RNN.

Our major contributions are the following:

– We show that despite providing strong hierarchical cues via a selectively sampled training set (Figure 1), RNNs do not generalize to an unseen combination of intervening nouns.

– Our findings further suggest that a soft hierarchical inductive bias, as imparted by the ONLSTM, in addition to a syntactically rich training set, is also insufficient to capture the underlying grammar of natural language.

– We verify that our findings are consistent across multiple learning paradigms, self-supervised

*Equal contribution

[1] A longer version of this paper is available at https://arxiv.org/abs/2010.04976.

[2] The main noun and the associated verb are in bold. Intervening nouns are underlined, and * denotes a grammatically incorrect sentence.

Proceedings of the Society for Computation in Linguistics (SCiL) 2021, pages 366-373.
Held on-line February 14-19, 2021

language modeling and supervised grammaticality judgment, as well as varied test sets, natural and constructed (Tables 1, 2).

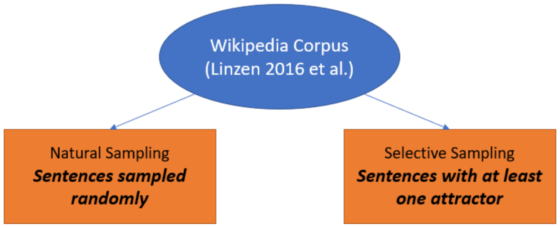


Figure 1: Dataset description. As structure-insensitive RNN models suffer from agreement attraction errors, our *selectively sampled* dataset is syntactically challenging for such sequential models, for whom surface-level heuristics such as 'agree with recent noun' are efficiently available. Hence, training RNNs on such datasets might induce them to capture hierarchical relations rather than learning shallower heuristics.

Architectures: In this work, we conduct our experiments on four recurrent schemes – LSTM (Hochreiter and Schmidhuber, 1997), GRU (Cho et al., 2014), Decay RNN (DRNN) (Bhatt et al., 2020), and ONLSTM (Shen et al., 2019). The governing equations of these architectures are mentioned in §A.1. ONLSTM is a recurrent network with soft hierarchical inductive bias.

Datasets: We use sentences from the Wikipedia corpus made available by Linzen et al. (2016). For training, we further choose two subsets from the main dataset, based on the number of attractors in each sentence (Figure 1). The sentences without any attractor are grammatically simple and allow for out-of-distribution testing as they are not seen while training on the selectively sampled dataset.

For the binary classifier, we augment each sentence with its corresponding counterfactual example.[3] Apart from testing on the sentences from the corpus (157k), we also test our models on synthetically generated sentences for targeted syntactic evaluation (Marvin and Linzen, 2018).

Experiments: In this work, we focus on evaluating the models' ability to make grammaticality judgments when trained for classification (supervised) and language modeling (self-supervised). For each task, we train models (with 5 different random seeds) on both training subsets from the corpus.

Consider the sentences from the introduction. A classifier is expected to label sentence 1 as ungrammatical and sentence 2 as grammatical. For grammaticality judgment via an LM, we train on a standard LM objective and during inference, check if our model gives a higher probability to the grammatically correct verb form conditioned on previous tokens in the sentence.

Training set	Natural Sampling				Selective Sampling			
Test attractors	0	1	2	3	0	1	2	3
	LANGUAGE MODEL							
LSTM	**0.98**	0.91	0.84	0.78	0.89	**0.98**	**0.98**	0.95
ONLSTM	**0.98**	0.92	0.86	0.82	0.90	**0.98**	**0.98**	0.95
GRU	**0.97**	0.88	0.78	0.73	0.87	**0.98**	0.97	0.94
DRNN	**0.96**	0.69	0.47	0.36	0.83	**0.97**	0.94	0.91
	BINARY CLASSIFIER							
LSTM	**0.97**	0.93	0.87	0.82	0.60	**0.98**	0.96	0.97
ONLSTM	**0.97**	0.91	0.84	0.81	0.64	**0.98**	0.97	**0.98**
GRU	**0.97**	0.88	0.76	0.69	0.62	0.95	0.94	**0.96**
DRNN	**0.97**	0.90	0.81	0.77	0.70	**0.97**	0.96	0.96

Table 1: Accuracy of RNN architectures trained as LMs and classifiers, for test instances with an increasing number of attractors between main subject and verb. The maximum accuracy for each model and training setup across attractor counts is in bold; standard deviations are in the Appendix, Table 5. Note that the models trained on the selectively sampled dataset are not able to generalize well OOD (sentences without attractors).

Performance on Natural Sentences: Table 1 shows the main results for the described experiments. For the models trained on a naturally sampled dataset, the performance degrades quite quickly with an increasing number of attractors between the subject and the corresponding verb, for both the LM and the classifier versions. However, the reduction in the accuracy with increasing attractor count for the models trained on the selectively sampled dataset is much less than with the natural sampling training.

For the selectively sampled dataset, the sentences without attractors serve as OOD sentences, and the performance boost on *in-distribution* complex sentences comes at the cost of a reduction in the accuracy on the OOD yet relatively *simple* sentences. The error rate for the ONLSTM, a model with inherent tree bias, also increases when tested on the OOD sentences, and when trained for a classification objective it performs worse than the architecturally simpler Decay RNN.

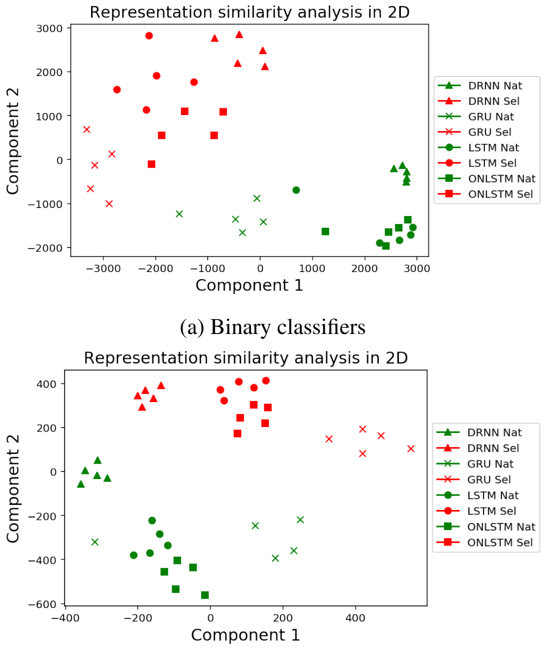
This fall-off on grammatically simpler OOD

[3] Augmenting with counterfactual examples is effective in reducing the spurious correlation in sentiment analysis (Kaushik et al., 2020). More information provided in §A.2 & §A.3.

samples seems counter-intuitive. We note that the increase in error rates is much greater when training the models as classifiers rather than LMs. This shows that models with supervised training for grammaticality on syntactically rich and counterfactually augmented data are still unable to capture the actual syntactic rules, and appear to be learning shallower heuristics, but ones which capture more nuanced patterns than simply going by linear distance. We can infer this because while our selectively sampled subset contains sentences with at least one attractor, many (over 30%) of the intervening nouns in these sentences are non-attractors. Hence there are sentences in which a non-attractor noun (same number as the main subject) immediately precedes the verb rather than an attractor noun. Therefore, the agreement performance (on sentences with attractors) of the models trained on this dataset cannot arise from an overly simple heuristic like disagreeing with the most recent noun, and the observed decline in OOD performance implies that less trivial heuristics are being learned which nevertheless fail to capture the actual syntax.

Analysis of representations: To analyze the differences in the learned internal representations among the models trained on the two subsets of the data, we perform a representation similarity analysis (RSA) (Laakso and Cottrell, 2000). We take 2000 sentences selected randomly from the test set. Our major observation from Figure 2 is that the representations of models trained on different subsets are easily linearly separable in this space, for both the LM and the classifier objectives. This implies that the representation clustering is not so much based on model architecture or inductive bias, but is driven more by the training data.

Targeted Syntactic Evaluation (TSE): We test how training the language models on the strategically chosen inputs impacts generalisation to different syntactic constructions. Testing on such examples lets us evaluate if our models are capturing what we intend them to capture (Marvin and Linzen, 2018). Table 2 presents the performance of the LMs on the synthetic data, for sentences with 0 or 1 agreement attractors. These findings corroborate our observations on natural language sentences – the models trained on the selectively sampled dataset performed worse on sentences without attractors which are syntactically simpler.



(a) Binary classifiers

(b) Language models

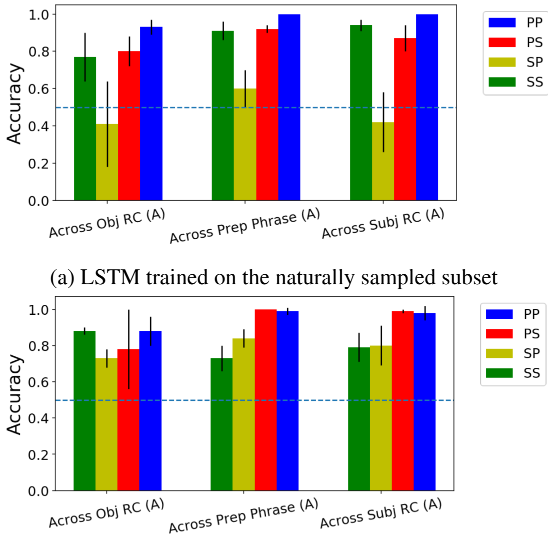
Figure 2: Representation similarity analysis of the hidden units of different RNN models (5 different seeds for each model). We observe that for both the learning objectives, one can partition the 2D space using a line which separates models trained on the two subsets of the data, natural and selective sampling.

To assess the performance of the models trained on the selectively sampled dataset, we take a closer look at constructed sentences that are structurally similar to in-distribution sentences but contain non-attractor intervening nouns rather than agreement attractors. Figure 3 depicts the performance of the LSTM LM on three agreement conditions – across Object RC, Preposition Phrase, and Subject RC, each with animate main noun. We observe that with our selective training, the performance on sentences with non-attractor intervening nouns (the SS/PP configurations, which are unobserved in the selectively sampled dataset) worsens substantively for 2 out of 3 syntactic constructions – across Preposition Phrase and Subject RC.

Discussion and Conclusion: In this work, we analyzed the effects of a strategically chosen training set with exclusively 'hard' agreement instances, on neural language models and binary classifiers for grammaticality judgment. We observed that the models' inability to perform well on out of distribution (OOD) sentences, even those which would

Training set	Natural		Selective	
Test attractors	0	1	0	1
LSTM	**0.77** (± 0.05)	0.66 (± 0.04)	0.63 (± 0.04)	**0.83** (± 0.06)
ONLSTM	**0.76** (± 0.07)	0.70 (± 0.06)	0.60 (± 0.05)	**0.85** (± 0.01)
GRU	**0.74** (± 0.02)	0.64 (± 0.02)	0.51 (± 0.02)	**0.81** (± 0.04)
DRNN	**0.67** (± 0.04)	0.44 (± 0.04)	0.48 (± 0.04)	**0.79** (± 0.03)

Table 2: Accuracy of LMs on test instances with 0 or 1 attractors from the artificial corpus. Models trained on the selectively sampled subset do not generalize well on OOD sentences without attractors. Performance across different syntactic constructions is shown in Table 6 in the Appendix.



(a) LSTM trained on the naturally sampled subset

(b) LSTM trained on the selectively sampled subset

Figure 3: Fine-grained analysis of the LSTM LM on Obj/Subj Relative Clauses and Preposition Phrases, demarcated by the inflections of the main subject and the embedded subject. P: Plural, S: Singular; thus SS denotes sentences with a singular main noun and a singular embedded subject, and likewise for the other cases.

seem to be ‘easy’ agreement instances, is consistent across variation in learning mechanism (supervised or self-supervised), innate architectural bias, and testing set – natural or artificial sentences.

Our analysis showed that the error rates of models trained on sentences with at least one agreement attractor are higher on sentences with no attractors than on sentences with attractors, for both corpus sentences (Table 1) and artificial sentences (Table 2). This observation is counter-intuitive because the models were trained on syntactically rich sentences and yet failed to perform well on simpler sentences. Had our RNN models picked up the correct grammatical rules, we would not expect this behavior. We obtained a similar counter-intuitive result for targeted syntactic evaluation (Appendix, Table 6), where models trained on the selectively sampled dataset performed much better on difficult constructed sentences involving agreement across nested dependencies, than on simpler sentences involving agreement within nested dependencies.

Our analysis of representations suggested that training set bias dominates over the model’s architectural features or inductive bias in shaping representation learning; *e.g.*, there was no discernible difference between the learned representations of ONLSTM and LSTM models. The reasons for this merit further exploration. Moreover, for the binary classifiers (Figure 2a), although we observe little variance in test accuracy across different training seeds, the variance in the projected representation space is substantially greater than for LMs. Thus, we posit that an LM objective is more reliable when comparing the ability of different RNN models to capture syntax-sensitive dependencies.

We observed that the hierarchical inductive bias in the ONLSTM is not sufficient to perform well on OOD sentences. McCoy et al. (2020) argued that an architecture with explicit tree bias, plus syntactically annotated inputs, are needed to capture syntax for sequence-to-sequence tasks. Here we show that the ONLSTM (soft tree bias) trained on a syntactically rich dataset (soft structural information) turns out to be insufficient to generalize well to OOD sentences and capture the underlying grammar. Our targeted syntactic evaluation pinpoints the cases which our models fail to capture, and improving performance on such cases is a key future direction.

Our observations suggest that RNNs being fundamentally statistical models can efficiently capture the correlation of the output variable with the input as observed during training, even for relatively ‘hard’ or non-linear linguistic dependencies, without necessarily learning the underlying hierarchical structure. This is consistent with the conclusions of Sennhauser and Berwick (2018) and Chaves (2020). Thus, we need to be cautious in inferring the ability of such models to capture syntax-sensitive dependencies. Performance on any particular kind of construction might always reflect some overfitting to it, even if it is syntactically rich or complex. Broad-based testing on instances of diverse types and complexity levels is essential to the development of models which better capture the structure of human language in all its richness and variety.

References

Gantavya Bhatt, Hritik Bansal, Rishubh Singh, and Sumeet Agarwal. 2020. How much complexity does an RNN architecture need to learn syntax-sensitive dependencies? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 244–254, Online. Association for Computational Linguistics.

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

Rui Chaves. 2020. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 1–11, New York, New York. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, 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 the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Yiding Hao. 2020. Attribution analysis of grammatical dependencies in lstms. *arXiv preprint arXiv:2005.00062*.

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

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Adhiguna Kuncoro, Chris Dyer, Laura Rimell, Stephen Clark, and Phil Blunsom. 2019. Scalable syntax-aware language models using knowledge distillation. *arXiv preprint arXiv:1906.06438*.

Aarre Laakso and Garrison Cottrell. 2000. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical psychology*, 13(1):47–76.

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.

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

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

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? Sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

Hiroshi Noji and Hiroya Takamura. 2020. An analysis of the utility of explicit negative examples to improve the syntactic abilities of neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3375–3385, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Luzi Sennhauser and Robert C Berwick. 2018. Evaluating the ability of lstms to learn context-free grammars. *arXiv preprint arXiv:1811.02611*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*. OpenReview.net.

A Appendix

A.1 Model Architectures

Following are the equations of the models used in this papers. '$\circ$' denotes the Hadamard product.

A.1.1 Long Short Term Memory (LSTM)

Following are the equations governing the standard LSTM (Hochreiter and Schmidhuber, 1997) with the standard notations.

$$
\begin{aligned}
i_t &= \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right)\\
f_t &= \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right)\\
g_t &= \tanh\left(W_g\left[h_{t-1}, x_t\right] + b_g\right)\\
o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)\\
c_t &= f_t \circ c_{t-1} + i_t \circ g_t\\
h_t &= o_t \circ \tanh\left(c_t\right)
\end{aligned}
$$

A.1.2 Gated Recurrent Unit (GRU)

Following are the equations governing the standard GRU (Cho et al., 2014) with the standard notations.

$$
\begin{aligned}
r_t &= \sigma\left(W_r\left[h_{t-1}, x_t\right] + b_r\right)\\
z_t &= \sigma\left(W_z\left[h_{t-1}, x_t\right] + b_z\right)\\
\tilde{h} &= \tanh\left(W_x\left[r_t \circ h_{t-1}, x_t\right] + b_x\right)\\
h_t &= z_t \circ h_{t-1} + \left(1 - z_t\right) \circ \tilde{h}
\end{aligned}
$$

A.1.3 Ordered Neurons (ONLSTM)

Ordered Neuron or Ordered Neuron LSTMs (Shen et al., 2019) are recurrent schemes that have been claimed to represent hierarchical information in their representations by their *cumax* or cumulative softmax activation. The following are the equations of Ordered Neurons with the standard notations.

$$
\begin{aligned}
f_t &= \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right)\\
i_t &= \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right)\\
o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)\\
\hat{c}_t &= \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right)\\
\tilde{f}_t &= \operatorname{cumax}\left(W_{\tilde{f}}\left[h_{t-1}, x_t\right] + b_{\tilde{f}}\right)\\
\tilde{i}_t &= 1 - \operatorname{cumax}\left(W_{\tilde{i}}\left[h_{t-1}, x_t\right] + b_{\tilde{i}}\right)\\
\omega_t &= \tilde{f}_t \circ \tilde{i}_t\\
\hat{f}_t &= f_t \circ \omega_t + \left(\tilde{f}_t - \omega_t\right)\\
\hat{i}_t &= i_t \circ \omega_t + \left(\tilde{i}_t - \omega_t\right)\\
c_t &= \hat{f}_t \circ c_{t-1} + \hat{i}_t \circ \hat{c}_t\\
h_t &= o_t \circ \tanh\left(c_t\right)
\end{aligned}
$$

A.1.4 Decay RNN (DRNN)

Decay RNN (DRNN) (Bhatt et al., 2020) is a bio-inspired recurrent baseline without any gating mechanism. Authors also show that DRNN surpasses vanilla RNNs on linguistic tasks.

$$
\begin{aligned}
c^{(t)} &= \left(\text{Re}\,LU(W)W_{\text{dale}}\right)h^{(t-1)} + Ux^{(t)} + b\\
h^{(t)} &= tanh\left(\alpha h^{(t-1)} + (1-\alpha)c^{(t)}\right)
\end{aligned}
$$

Here $\alpha \in$ (0,1) as a learnable parameter and $\mathbf{W}_{dale}$ is a diagonal matrix which provides biological constraints.

Property	Natural	Selective
Training sentences	97842	97842
Ratio of Singular to Plural main nouns	0.67	0.45
Ratio of Singular to Plural nouns (total)	0.79	0.71
Fraction of 0 attractors	0.93	-
Fraction of 1 attractors	0.056	0.79
Fraction of 2 attractors	0.011	0.15
Fraction of 3 attractors	0.003	0.037
Testing Sentences	157k	157k

Table 3: Training data statistics.

A.2 Training Settings

Statistical information about our training data is shown in Table 3. In our experiments, we train a two-layered LM where we keep the hidden size at 650 units and the input size at 200 units. We perform standard dropout with a rate of 0.2 and the batch size 128. Optimization starts with a 0.001 learning rate for all architecture and clips the gradient if necessary.

For Binary classifiers, we use a single-layered recurrent unit, batch size of 64, hidden, and input size of 50 units. For LSTM and ONLSTM, the initial learning rate is 0.005, while for the GRU and DRNN, it is 0.01. No gradient clipping is performed to train the classifier.

All models are optimized with Adam (Kingma and Ba, 2015), and the codes are written in PyTorch (Paszke et al., 2019).

A.3 Binary Classifier and Counterfactual Augmentation

For the binary classifier, we augment each sentence with its corresponding counterfactual example. Augmenting with counterfactual examples is effective in reducing the spurious correlation in sentiment analysis (Kaushik et al., 2020). In our case, the counterfactual example will be constructed by

Binary Classifier	LSTM		GRU		ONLSTM		DRNN	
Configuration	Without Augmentation	With Augmentation	Without Augmentation	With Augmentation	Without Augmentation	With Augmentation	Without Augmentation	With Augmentation
Natural Sampling	0.96	0.97	0.94	0.96	0.95	0.96	0.95	0.96
Selective Sampling	0.64	0.64	0.62	0.67	0.59	0.69	0.71	0.74

Table 4: Performance of Binary classifier without counterfactual augmentation. Counterfactual augmentation effectively doubles the training size.

Architecture	Natural Sampling				Selective Sampling			
	0	1	2	3	0	1	2	3
	LANGUAGE MODEL							
LSTM	**0.98** (±0.00)	0.91 (±0.01)	0.84 (±0.03)	0.78 (±0.06)	0.89 (±0.01)	**0.98** (±0.00)	**0.98** (±0.00)	0.95 (±0.01)
ONLSTM	**0.98** (±0.00)	0.92 (±0.01)	0.86 (±0.01)	0.82 (±0.03)	0.90 (±0.01)	**0.98** (±0.00)	**0.98** (±0.00)	0.95 (±0.01)
GRU	**0.97** (±0.00)	0.88 (±0.01)	0.78 (±0.02)	0.73 (±0.03)	0.87 (±0.01)	**0.98** (±0.00)	0.97 (±0.00)	0.94 (±0.01)
DRNN	**0.96** (±0.00)	0.69 (±0.02)	0.47 (±0.03)	0.36 (±0.03)	0.83 (±0.01)	**0.97** (±0.00)	0.94 (±0.01)	0.91 (±0.01)
	BINARY CLASSIFIER							
LSTM	**0.97** (±0.01)	0.93 (±0.02)	0.87 (±0.03)	0.82 (±0.03)	0.60 (±0.06)	**0.98** (±0.00)	0.96 (±0.00)	0.97 (±0.01)
ONLSTM	**0.97** (±0.01)	0.91 (±0.05)	0.84 (±0.07)	0.81 (±0.07)	0.64 (±0.08)	**0.98** (±0.00)	0.97 (±0.00)	**0.98** (±0.01)
GRU	**0.97** (±0.00)	0.88 (±0.01)	0.76 (±0.02)	0.69 (±0.04)	0.62 (±0.05)	0.95 (±0.01)	0.94 (±0.02)	**0.96** (±0.01)
DRNN	**0.97** (±0.00)	0.90 (±0.01)	0.81 (±0.02)	0.77 (±0.02)	0.70 (±0.02)	**0.97** (±0.00)	0.96 (±0.00)	0.96 (±0.01)

Table 5: Performance of LM and classifier with an increasing number of attractors between the main subject and verb. Bolds mark the maximum accuracy in each configuration across the attractor, for each model; the more the better.

flipping the number of the main verb of a grammatically correct sentence. Thus, we use the correct as well as the incorrect version of the same sentence in training. This results in the training size of 195k sentences for the binary classifier. Table 4 shows the performance with/without counterfactual augmentation. Note that, the accuracy improved substantially for ONLSTM trained on the selectively sampled dataset.

In Table 5 we give a full version of Table 1 including the standard deviations on 5 different runs.

A.4 Targeted Syntactic Evaluation

Table 6 presents the detailed performance of models on the synthetically constructed sentences (TSE).

Subject Verb Agreement	#sentences	LSTM		ONLSTM		GRU		DRNN	
Condition		Natural	Selective	Natural	Selective	Natural	Selective	Natural	Selective
Simple	312	**0.99** (±0.01)	0.86 (±0.01)	**0.98** (±0.02)	0.86 (±0.01)	**0.98** (±0.01)	0.84 (±0.04)	**0.97** (±0.02)	0.79 (±0.05)
Short VP	3432	**0.85** (±0.02)	0.71 (±0.06)	**0.88** (±0.02)	0.73 (±0.08)	**0.81** (±0.03)	0.69 (±0.04)	**0.70** (±0.05)	0.66 (±0.04)
Within ORC (A)	9984	**0.79** (±0.06)	0.63 (±0.05)	**0.78** (±0.10)	0.59 (±0.06)	**0.75** (±0.02)	0.50 (±0.02)	**0.7** (±0.08)	0.46 (±0.04)
Within ORC (IA)	4032	**0.77** (±0.06)	0.64 (±0.06)	**0.75** (±0.08)	0.59 (±0.04)	**0.73** (±0.02)	0.50 (±0.03)	**0.69** (±0.06)	0.46 (±0.05)
Within no that ORC (A)	9984	**0.73** (±0.06)	0.61 (±0.05)	**0.72** (±0.08)	0.57 (±0.07)	**0.72** (±0.03)	0.47 (±0.04)	**0.63** (±0.04)	0.45 (±0.06)
Within no that ORC (IA)	4032	**0.66** (±0.04)	0.61 (±0.05)	**0.66** (±0.06)	0.56 (±0.06)	**0.62** (±0.04)	0.47 (±0.04)	**0.68** (±0.06)	0.45 (±0.06)
Long VP	520	0.65 (±0.03)	**0.69** (±0.07)	**0.67** (±0.04)	**0.67** (±0.06)	0.63 (±0.04)	**0.65** (±0.04)	0.56 (±0.05)	**0.65** (±0.03)
Across Prep Phrase (A)	29952	0.86 (±0.04)	**0.89** (±0.03)	**0.88** (±0.03)	**0.88** (±0.01)	0.81 (±0.02)	**0.88** (±0.02)	0.68 (±0.04)	**0.83** (±0.01)
Across Prep Phrase (IA)	4032	0.87 (±0.03)	**0.94** (±0.02)	0.88 (±0.02)	**0.95** (±0.01)	0.86 (±0.02)	**0.94** (±0.01)	0.69 (±0.06)	**0.91** (±0.02)
Across SRC	9984	0.81 (±0.03)	**0.89** (±0.05)	0.81 (±0.05)	**0.87** (±0.02)	0.77 (±0.05)	**0.86** (±0.05)	0.58 (±0.04)	**0.80** (±0.05)
Across ORC (A)	9984	0.73 (±0.10)	**0.82** (±0.07)	0.78 (±0.07)	**0.84** (±0.02)	0.72 (±0.06)	**0.79** (±0.05)	0.63 (±0.04)	**0.78** (±0.05)
Across ORC (IA)	4032	0.74 (±0.09)	**0.84** (±0.10)	0.81 (±0.07)	**0.87** (±0.02)	0.74 (±0.08)	**0.85** (±0.05)	0.65 (±0.07)	**0.86** (±0.02)
Across no that ORC (A)	9984	0.61 (±0.04)	**0.72** (±0.08)	0.62 (±0.05)	**0.78** (±0.02)	0.60 (±0.02)	**0.68** (±0.06)	0.64 (±0.03)	**0.73** (±0.02)
Across no that ORC (IA)	4032	0.66 (±0.04)	**0.77** (±0.11)	0.66 (±0.06)	**0.84** (±0.03)	0.62 (±0.04)	**0.72** (±0.07)	0.68 (±0.06)	**0.83** (±0.02)
Average Performance	104296	0.78 (±0.03)	0.78 (±0.02)	0.79 (±0.03)	0.78 (±0.01)	0.75 (±0.01)	0.73 (±0.02)	0.66 (±0.02)	0.71 (±0.02)

Table 6: Accuracy of models on targeted syntactic evaluation. Quantities in bold marks the maximum accuracy for each model across the configuration. ORC: Objective Relative Clause, SRC: Subject Relative Clause, Prep Phrase: Prepositional Phrase, VP: Verb Phrase. A/IA in the parenthesis represents an animate/inanimate main subject. Models trained on selectively sampled subset perform well on the difficult sentences, but not on the simpler ones.