How much syntax do neural networks learn?

Stephen Pulman

Dec 2018

Some history

- The Chomsky hierarchy of formal languages (from least to most powerful):
  - regular (finite state) e.g. $a^n b^m$
  - context-free e.g. $a^n b^n$
  - context-sensitive e.g. $a^n b^n c^n$
  - unrestricted rewriting systems: Turing machine equivalent
- Chomsky's 1957 Syntactic Structures argued that English and other languages were (at least) context free.
- A language consisting of all strings of the form $a^n b^n$ cannot be defined by a regular grammar.
- But it can be captured by a simple CFG:

  $\mathbf{S} \rightarrow \mathbf{aSb}$
  $\mathbf{S} \rightarrow \mathbf{ab}$

Competence vs. performance

English has some constructions that display this kind of $a^n b^n$ pattern:

$S \rightarrow$ *Either S or S*
$S \rightarrow$ *If S then S*
$S \rightarrow$ *Both S and S*
The X that the Y that ....

If either John loses or Bill resigns, then Mary will take the prize.
If either both John loses and Bill resigns or Mary takes the prize, then Susan will be happy.

The mouse that the cat chased ran away.
The mouse that the cat that the dog chased caught escaped unhurt.
The mouse that the cat that the dog that John shouted at chased caught escaped unhurt.

Structure dependence

If you know a language you know lots of things about the relation between sentences, for example:

Declarative: The man is tall
Yes-no question: Is the man tall?

Hypothesis 1 (how to question): find first verb from the start of the sentence, and prepose it:

The man is tall ⇒ Is the man tall?

This will work most of the time, but not always:

The man who was here is tall ⇒ Was the man who here is tall?

Hypothesis 2: find the first verb after the subject Noun Phrase and prepose that.

[The man who was here] is tall ⇒ Is [the man who was here] tall?

This "structure dependent" hypothesis requires that at some level a speaker is analysing the sentence hierarchically into abstract phrases, i.e. parsing using something like CFG.

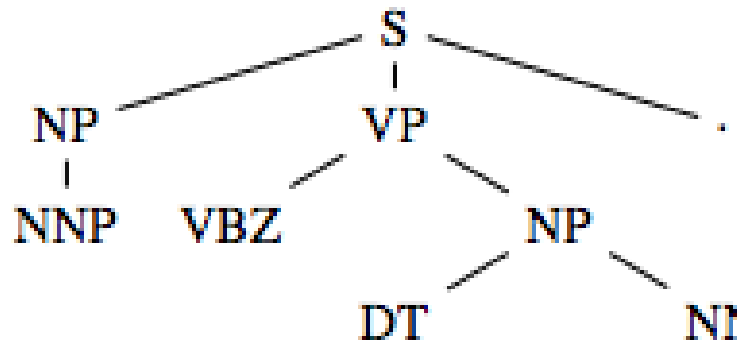
What is the representational capacity of neural networks?

- Recurrent NNs have an architecture that lends itself to the characterisation of regular languages, and they can indeed learn these languages (Casey 1996).
- A number of experiments have been made to see whether RNNs can learn context-free languages, either simple fragments of English, or artificial examples of the $a^n b^n$ type (Elman 1991, Tonkes & Wiles 1997)
- But the results have been debateable, to say the least, and the general view is that such systems are not able to generalise successfully to large values of n (Gers & Schmidhuber 2001)).
- LSTMs, on the other hand, (Gers & Schmidhuber 2001), are claimed to be able to make the right generalisations for these artificial languages, including more complex ones of the form $a^n b^m c^m d^n$, and even to learn some examples of artificial context-sensitive languages.

Structure Dependence: sequence models

Given that structure dependence is a central part of language, how successful have DL methods been at learning this property?

Vinyals et al. (2015) show how an LSTM encoder-decoder sequence-to-sequence model with attention can assign linearised parse trees to input:

John has a dog . → 

John has a dog . → (S (NP NNP)$_{NP}$ (VP VBZ (NP DT NN)$_{NP}$)$_{VP}$.)$_{S}$

Structure Dependence: sequence models

```
(S (NP  NNP  NP) (VP  VBZ  (NP  DT  NN  NP)  VP) S)
        |              |        |   |
      John           has       a  dog
```

- The system was trained on over 11m parsed sentences and reached 92.1% F1 score on PTB23: a very impressive result.
- But has this system really learned the notion of hierarchical structure? Note that the labelled brackets constitute a CF language of the $a^n b^n$ type.
- I would say not: it is transducing between two *strings* of symbols, and has no *general* idea that a left bracket should always be matched with a right bracket.
- The errors nearly all involve missing right brackets, suggesting it is memorising particular patterns rather than learning a general rule.

Structure Dependence: modelling subject-verb agreement

Linzen et al. (2016) experimented with trying to predict **subject**-*verb* agreement using LSTMs on word embeddings, with different numbers of "attractors":

1. The **students** *submit* a final project to complete the course.
2. The **students** enrolled in the program *submit* a final project to complete the course.
3. The **students** enrolled in the program in the Department *submit* a final project to complete the course.
4. The **students** enrolled in the program in the Department where my colleague teaches *submit* a final project to complete the course.

Results seem reasonable: with 4 intervening distractors, error rate is only 17.6%, although note that "most naturally occurring agreement cases in the Wikipedia corpus are easy".

> *"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."*

Learning syntax, or lexical relations?

Bernardy & Lappin (2017) repeated these experiments, with broadly similar results, although accuracy improved with more data and higher dimension embeddings. They also experimented with a reduced vocabulary version intended to encourage learning of abstract syntactic structure (100 most frequent words vocab, with all other words represented by their POS tags.) This didn't work well:

> *"DNNs learn better from data populated by richer lexical sequences. This suggests that DNNs are not efficient at picking up abstract syntactic patterns when they are explicitly marked in the data. Instead they extract them incrementally from lexical embeddings through recognition of their distributional regularities. It is also possible that they use the lexical semantic cues that larger vocabularies introduce to determine agreement preferences for a verb."*

Colorless green recurrent networks dream hierarchically

This raises the question of what exactly is being learned here: structure dependence, or lexical correlations? Is it the correlation of "dogs" with "bark" that is learned rather than the subject-verb grammatical relationship?

The dogs in the field owned by the farmer bark frequently.

If so, this would suggest that DNNs would do less well on sentences that are grammatical but nonsensical, like Chomsky's "colorless green ideas sleep furiously", an idea put to the test by Gulordava et al. (2018).
In this paper, the Facebook AI group repeated the Linzen et al. experiments using both grammatical and nonsensical sentences, and a range of 12 different agreement constructions in four different languages, English, Italian, Hebrew and Russian.
NB Only two of these constructions - subject-verb agreement and conjoined verb agreement ("He sings songs and dances" vs "he sings songs and dance") - occur in English.

Sense, nonsense, and syntax

The nonsensical sentences were generated from the original test sentences by randomly replacing content words by others with the same morphology and part of speech. Note that this method does not preserve argument structure requirements, as can be seen from *presents* vs. *stays* in the following example:

- It *presents* the case for marriage equality and *states* ...
- It *stays* the shuttle for honesty insurance and *finds* ...

They also conducted an experiment to compare LSTMs with human accuracy. The Italian test set (119 original and 1071 nonce sentences, balanced out with fillers) was presented to native speakers via Amazon Mechanical Turk, up to the point of the target. Then subjects were asked to choose which was the more plausible form for the target. (Average 9 judgements per item)

All languages LSTM accuracy

		N V V	V NP conj V
Italian	Original	$93.3_{\pm 4.1}$	$83.3_{\pm 10.4}$
	Nonce	$92.5_{\pm 2.1}$	$78.5_{\pm 1.7}$
English	Original	$89.6_{\pm 3.6}$	$67.5_{\pm 5.2}$
	Nonce	$68.7_{\pm 0.9}$	$82.5_{\pm 4.8}$
Hebrew	Original	$86.7_{\pm 9.3}$	$83.3_{\pm 5.9}$
	Nonce	$65.7_{\pm 4.1}$	$83.1_{\pm 2.8}$
Russian	Original	-	$95.2_{\pm 1.9}$
	Nonce	-	$86.7_{\pm 1.6}$

Table 2: LSTM accuracy in the constructions N V V (subject-verb agreement with an intervening embedded clause) and V NP conj V (agreement between conjoined verbs separated by a complement of the first verb).

LSTM vs. human

Construction	#original	Original		Nonce	
		Subjects	LSTM	Subjects	LSTM
DET [AdjP] NOUN	14	98.7	$98.6_{\pm 3.2}$	98.1	$91.7_{\pm 0.4}$
NOUN [RelC / PartP] clitic VERB	6	93.1	$100_{\pm 0.0}$	95.4	$97.8_{\pm 0.8}$
NOUN [RelC / PartP] VERB	27	97.0	$93.3_{\pm 4.1}$	92.3	$92.5_{\pm 2.1}$
ADJ [conjoined ADJs] ADJ	13	98.5	$100_{\pm 0.0}$	98.0	$98.1_{\pm 1.1}$
NOUN [AdjP] relpron VERB	10	95.9	$98.0_{\pm 4.5}$	89.5	$84.0_{\pm 3.3}$
NOUN [PP] ADVERB ADJ	13	91.5	$98.5_{\pm 3.4}$	79.4	$76.9_{\pm 1.4}$
NOUN [PP] VERB (participial)	18	87.1	$77.8_{\pm 3.9}$	73.4	$71.1_{\pm 3.3}$
VERB [NP] CONJ VERB	18	94.0	$83.3_{\pm 10.4}$	86.8	$78.5_{\pm 1.7}$
(Micro) average		94.5	$92.1_{\pm 1.6}$	88.4	$85.5_{\pm 0.7}$

Table 3: Subject and LSTM accuracy on the Italian test set, by construction and averaged.

Although there was a drop in accuracy between grammatical and nonsense sentences, it was about the same for their LSTM system and for people. For Italian at least, the LSTM almost reached human performance. They conclude, tentatively, that the LSTM *is* learning grammatical representations rather than lexical dependencies.

LSTM vs RNNG

Recursive Neural Network Grammars (Dyer et al. 2016) are stack-based shift-reduce parsers and sentence generators which achieve state of the art performance on standard benchmarks. Parsing example:

Input: *The hungry cat meows .*

	Stack	**Buffer**	**Action**
0		*The* | *hungry* | *cat* | *meows* | .	NT(S)
1	(S	*The* | *hungry* | *cat* | *meows* | .	NT(NP)
2	(S | (NP	*The* | *hungry* | *cat* | *meows* | .	SHIFT
3	(S | (NP | *The*	*hungry* | *cat* | *meows* | .	SHIFT
4	(S | (NP | *The* | *hungry*	*cat* | *meows* | .	SHIFT
5	(S | (NP | *The* | *hungry* | *cat*	*meows* | .	REDUCE
6	(S | (NP *The hungry cat*)	*meows* | .	NT(VP)
7	(S | (NP *The hungry cat*) | (VP	*meows* | .	SHIFT
8	(S | (NP *The hungry cat*) | (VP *meows*	.	REDUCE
9	(S | (NP *The hungry cat*) | (VP *meows*)	.	SHIFT
10	(S | (NP *The hungry cat*) | (VP *meows*) | .		REDUCE
11	(S (NP *The hungry cat*) (VP *meows*) .)		

RNNG

- Parsing decisions are conditioned on the state of the input buffer, the sequence of parsing actions, the structure built so far, and the state of the stack.
- The stack is encoded as a stack LSTM, and the other components as RNNs.
- Separately, Kuncoro et al. (2017) showed, using an attention mechanism, that RNNGs seem to partly learn the notion of the "head" of a constituent (the most "important" word in a phrase). In complex phrases this is important for agreement: "[NP the dogs in the kennel] bark all night"
- RNNGs outperform both sequential LSTMs, and LSTMs trained on linearised parse trees (Choe & Charniak 2016).
- Note that LSTMs with a much bigger hidden state did a lot better than the original Linzen et al systems.

RNNGs vs LSTM on the number agreement dataset

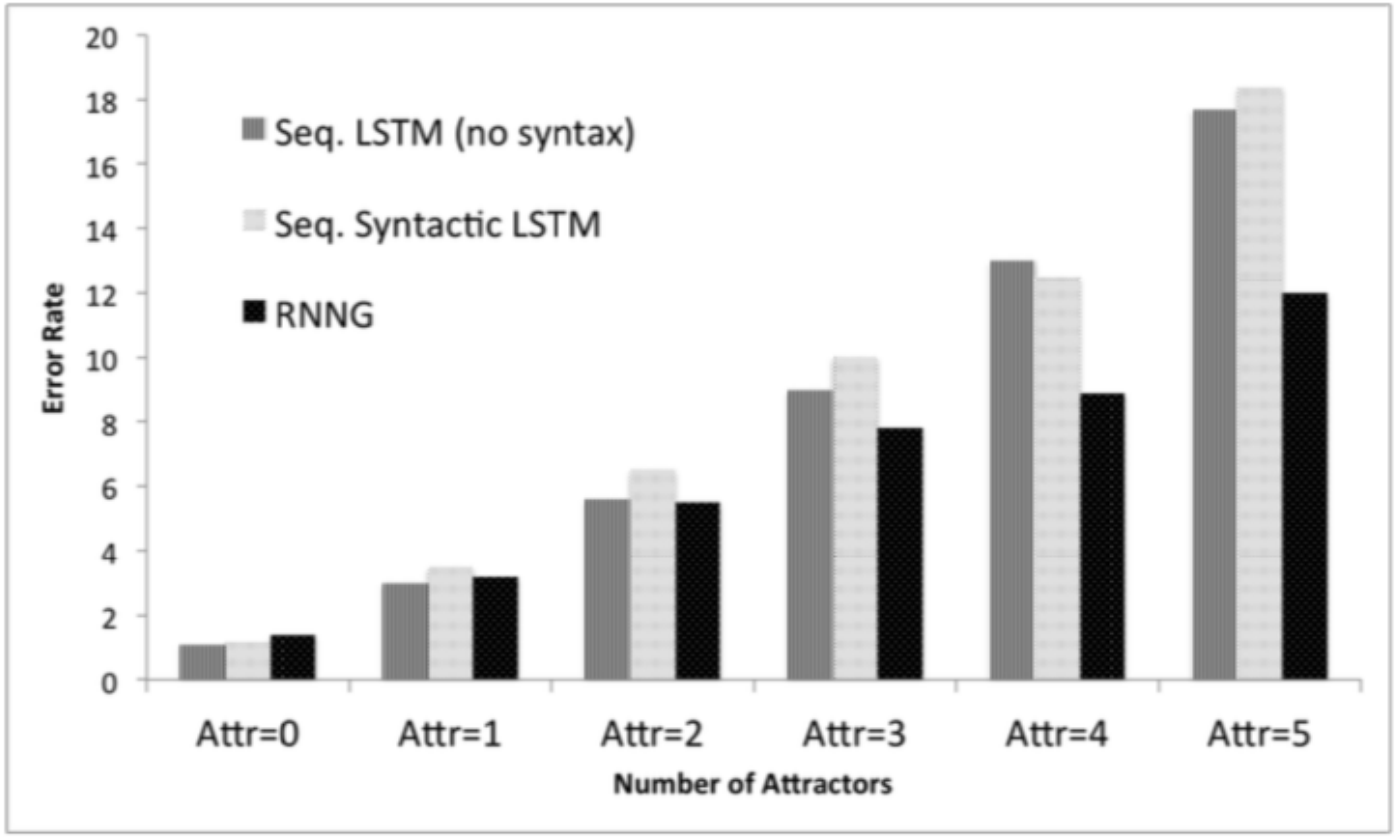


Figure 2: Number agreement error rates for sequential LSTM language models (left), sequential syntactic LSTM language models (Choe and Charniak, 2016, center), and RNNGs (right).

RNNGs and stacks

It is the hierarchical nature of constituent assembly made possible by use of a stack that enables RNNGs to outperform sequential LSTMs. This suggests that if you augment an LSTM with a stack, a similar improvement should be possible.
The same DeepMind group did exactly this (Yogatama et al. 2018).

Model	**Number of attractors**						**Acc.**	**Ppx.**
	0	**1**	**2**	**3**	**4**	**5**		
Best LSTM	99.3	97.2	95.0	92.2	90.0	84.2	99.11	23.8
Best attention	**99.4**	97.7	95.9	92.9	90.7	84.2	99.18	22.7
Best stack	**99.4**	**97.9**	**96.5**	**93.5**	**91.6**	**88.0**	**99.23**	**22.2**

Table 2: Accuracies on the Linzen number prediction dataset. 0, 1, 2, 3, 4, and 5 refer to the number of attractors between the subject and the predicted verb (see text for details).

Errors

Model			Example
LSTM	attention	stack	
✗	✗	✗	*the NN notes and front cover title {is,**are**}*
✗	✗	✓	*other NNS that in the recent past were part of the JJ parish {is,**are**}*
✗	✓	✗	*the class of all VBN sets with JJ functions as NNS {form,**forms**}*
✓	✗	✗	*various brands of JJ compound or NN NN {helps,**help**}*
✗	✓	✓	*score based on penalties for fallen bars , NNS , {**falls**,fall}*
✓	✗	✓	*the loss of basic needs providers VBG from VBN countries {**has**,have}*
✓	✓	✗	*the construction of the JJ walls , floors , and VBG walls {**is**,are}*

Table 3: Examples of mistakes made by competing models on the Linzen number prediction dataset. ✗ indicates an incorrect prediction, whereas ✓ indicates a correct prediction. In general, we observe that the mistakes made by both the LSTM and attention models that are correctly predicted by the stack model (row 2) typically involve longer sentences regardless of the number of attractors.

A quick bit of philosophy

- **Empiricism**: the doctrine that learning starts with a "tabula rasa" or "blank sheet of paper" and "nothing is in the mind that was not first in the senses". Simple mechanisms like similarity and difference suffice to learn language by exposure to data.
- **Rationalism**: the idea that learning requires some *a priori* structure or "innate ideas". We have a strong inductive bias towards some (among many logically possible) solutions to learning problems.
- Chomsky took his observations about structure dependence of language (and other facts about language learning) to support a rationalist position against the (then and now) prevailing empiricism. In this view, notions like structure dependence are hard-wired in us.

Which view does our NN story support?

- NNs with no *a priori* structure struggle to learn to accurately process languages with the simplest characteristics of natural languages, i.e. structure dependence.
- NNs with the right inductive bias (RNNGs, LSTMs with a stack) seem to do a much better job.
- If the first two points above are correct, then it suggests that Chomsky was essentially right.
- If so, it follows that we are unlikely to find "general learning mechanisms", but should rather be looking for the right *a priori* structure to encode to arrive at (relatively) task specific learning systems.

Bernardy, J.-L. & Lappin, S. (2017), 'Using deep neural networks to learn syntactic agreement', *LILT: Linguistics in Language Technology* **15**(2).

Casey, M. (1996), 'The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction', *Neural Comput.* **8**(6), 1135–1178.
URL: *http://dx.doi.org/10.1162/neco.1996.8.6.1135*

Choe, D. K. & Charniak, E. (2016), Parsing as language modeling, *in* 'EMNLP', The Association for Computational Linguistics, pp. 2331–2336.

Dyer, C., Kuncoro, A., Ballesteros, M. & Smith, N. A. (2016), Recurrent neural network grammars, *in* 'Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies', Association for Computational Linguistics, pp. 199–209.
URL: *http://aclweb.org/anthology/N16-1024*

Elman, J. L. (1991), 'Distributed representations, simple recurrent networks, and grammatical structure', *Machine Learning* **7**, 195–224.

Gers, F. A. & Schmidhuber, J. (2001), 'LSTM recurrent networks learn simple context-free and context-sensitive languages', *IEEE Trans. Neural Networks* **12**(6), 1333–1340.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T. & Baroni, M. (2018), Colorless green recurrent networks dream hierarchically, *in* 'NAACL-HLT', Association for Computational Linguistics, pp. 1195–1205.

Kuncoro, A., Ballesteros, M., Kong, L., Dyer, C., Neubig, G. & Smith, N. A. (2017), What do recurrent neural network grammars learn about syntax?, *in* 'Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers', Association for Computational Linguistics, Valencia, Spain, pp. 1249–1258.
URL: *http://www.aclweb.org/anthology/E17-1117*

Linzen, T., Dupoux, E. & Goldberg, Y. (2016), 'Assessing the ability of lstms to learn syntax-sensitive dependencies', *TACL* **4**, 521–535.

Tonkes, B. & Wiles, J. (1997), Learning a context-free task with a recurrent neural network: An analysis of stability, *in* 'Proc of Fourth Biennial Conference of the Australasian Cognitive Science Society'.

Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I. & Hinton, G. E. (2015), Grammar as a foreign language, *in* 'NIPS', pp. 2773–2781.

Yogatama, D., Miao, Y., Melis, G., Ling, W., Kuncoro, A., Dyer, C. & Blunsom, P. (2018), Memory architectures in recurrent neural network language models, *in* 'International Conference on Learning Representations'.
URL: *https://openreview.net/forum?id=SkFqf0lAZ*