Hierarchy or Heuristic? Examining hierarchical structure and the poverty of the stimulus in recurrent neural networks

Isabel V. Papadimitriou
Department of Computer Science
Stanford University
isabelvp@stanford.edu

Abstract

The impressive successes of recurrent neural networks (RNNs) in natural language tasks has led to a new field of research, examining the ways in which RNN models exhibit aspects of the human grammar system as we understand it. Much of this work looks for hierarchical syntactic awareness in RNNs by examining a model's performance in the face of long-range dependencies over subordinate clauses. Successfully recognizing subordinate clauses however, is not equivalent to having a tree-structured hierarchical understanding of language. We show that it is possible for a model to perform well on recognizing long-range dependencies and yet fail to exhibit more global hierarchical awareness. We propose more thorough criteria for defining hierarchical structural awareness: models should not only be able to recognize which clauses are separate, but also to embed relationships between them. We present metrics and analytical methods that begin to disambiguate between successful chunking of clauses, and a structured hierarchical representation of them. We replicate the experiments in McCoy et al (2018), and find that the outputs of the language model indicate awareness of long range dependencies as they report, but do not pass our other metrics of structural sensitivity. However, we look past its outputs and also probe the model's internal state. In doing this, we observe that the model is significantly more likely to produce grammatical than ungrammatical sentences, ultimately supporting McCoy et al's claim that RNNs some hierarchically-informed representation of language.

1 Introduction

LSTMs with attention have been massively successful in a variety of natural language processing tasks. These successes, along with the opacity of end-to-end systems, lead to the question of what makes LSTMs so capable with linguistic data. In recent years, there has been a focus on examining if LSTMs have an understanding of language that is similar to the human language system. Much of this research has focused on the notion of hierarchical structure. While there is a general linguistic consensus that humans create some form of tree-structured parse of utterances, it is not clear whether neural NLP models have this view of language. Techniques that probe LSTMs for hierarchical encodings of language have been the topic of research in the last few years, starting most notably with Linzen et al [7]. McCoy et al [9] build on Linzen's work, by testing if LSTMs have a bias towards developing hierarchical representations even when trained on data that is explainable without a hierarchical grammar.

The way in which these works have generally probed for hierarchical structure is through testing whether models successfully ignore subordinate clauses. If a model knows to fill in the singular "is" in the sentence "The boy petting the cats __ happy", this means that the model somehow encodes

"petting the cats" as separate from the overall sentence "the boy is happy". This, they argue, is indicative of a tree-structured grammar: instead of looking linearly back for the first noun (which would be "cats") to find the clause's number, the model is aware that there is some sort of subtree that it must skip over.

Through replicating and examining McCoy et al's results, we observed that models can perform well on identifying a subordinate clause as a separate unit while in fact failing in higher-level grammatical awareness. For example, upon closer inspection, models that successfully ignored a subordinate clause did not seem to encode which phrase the clause was subordinate to or indeed if it was subordinate to any clause. Showing that subordinate clauses lie on some other level than the main arguments, as the probing literature has focused on, is certainly indicative of some type of structural awareness. However, performance in this domain is not equivalent to having a tree-structured grammar

We propose that to probe for hierarchy, we should expect models to show a more global understanding of the sentence structure than previous work has searched for. We conducted further experiments, expanding the data used in McCoy et al and evaluating performance on a metric that tests awareness of the top-level hierarchical branch: knowing if a clause is subordinate to the main NP, or the main VP in a sentence. We found that none of the models were biased towards keeping constituents subordinate to the correct side, suggesting that they lacked a full top-down hierarchical representation of the sentence. We also probed the models for more global understandings of syntax by looking at the probabilities that they assigned to different types of ungrammatical sentences. We report that models have a bias for for grammatical sentences over ungrammatical ones, but do not have a bias to distinguish the correct grammatical sentence that maintains the same hierarchy as the input.

2 Related Work

This paper is most closely related to the methodological stream exemplified by Linzen et al 2016 [7]. Linzen tested language models' ability to perform the task of subject-verb number agreement. The models were given inputs that signified the number of the subject and were asked to complete the verb. In English, many verbs are inflected in the third-person according to the number of the subject, as in "the cat runs" and "the cats run". The inputs to Linzen's language models were phrases of the type "The cats in the garden of my house (runs/run)", where there are nouns that do not agree with the subject in number that come after the subject. The language model was tested on whether it successfully ignored these nouns to produce a verb in the correct number.

There have since been many papers in this experimental framework, expanding on different aspects of Linzen's study. Bernardy and Lappin [1] widen the scope of Linzen's experiments, conducting similar experiments on different architectures and seeing the effect of changing hyperparameters such as vocabulary size, hidden size, or dropout rate. Gulordava et al [4] note that Linzen's results may be due to semantic cues rather than syntactic awareness. They give examples of sentences like "the dogs in the neighborhood (barks/bark)", where it is clear that dogs bark and neighborhoods do not, and this alone could determine the verb number agreement. To take out this confounding factor, they switch out words with other words of the same part of speech to create nonce sentences like "The braveries in the spaceship (barks/bark)" and train an LSTM on these. Linzen and Leonard [8] expand on Linzen's work by more closely examining the types of mistakes LSTMs make in long-distance agreement, compared to what it is known confuses humans. They find that LSTMs exhibit some types of human confusion, but are also thrown off by a family of sentences that humans do very well on.

Apart from methods that are in the experimental family of Linzen's 2016 paper, there is another probing methodology that has been followed widely in recent years. Exemplified by Conneau et al [3], these methods rely on taking the hidden states of LSTMs in different input contexts and examining if LSTMs encode the syntactic information of these different inputs by looking at how they project them into the space of hidden states.

3 Approach

3.1 Replicating McCoy's Experiments

Our main methodological backbone was replicating McCoy et al's paper [9] and then expanding the experiment scope and the evaluation methods to more critically asses the reported findings. McCoy's work is based around exploring RNNs in the context of the argument about poverty of the stimulus, introduced by Chomsky and revisited by Berwick, Chomsky et al in more contemporary terms in [2]. In general terms, argument is that children are not exposed to enough data in their first years that explicitly shows that language is tree-structured, and yet the grammar that humans deduce is. Therefore, the argument states that humans have an innate bias towards constructing a hierarchical grammar even from limited data. McCoy et al take this approach to the critical study of RNNs. They train models on linearly-explainable data, and then test those models on a "generalization set", which requires a hierarchical rule. This environment is a rudimentary controlled version of the process Chomsky described in children, and sets an experimental framework for extending human language acquisition studies to neural networks.

The task that McCoy et al trained the models on was the sequence-to-sequence task of auxiliary fronting, which has been used often in human language studies around the poverty of the stimulus argument (see [10], [5] and [6]). The task is to produce a question by moving the correct auxiliary verb to the front of the sentence, for example in the following transformation:

(1) The girl with the red shirt will love these cats $\rightarrow$ Will the girl with the red shirt love these cats?

The training set consisted of examples where the main auxiliary was also the one closest to the subject noun, such as (1) above. The generalization set had cases where there was an intervening relative clause between the subject and the main auxiliary, as in "The girl who might visit will love these cats" $\rightarrow$ "Will the girl who might visit love these cats?". In those cases, one has to disambiguate which verb is part of the main phrase ("will"), and which one is part of a lower-level subordinate clause ("might").

McCoy et al use data that is generated from a Phrase Structure Grammar (PSG) that they specify in the supplementary materials of [9]. We generated data from this grammar by writing a generator program. The generator probabilistically generates samples by sampling from syntactic choices (eg, should I put an adjective by this noun?) and vocabulary choices (eg, should this noun be realized as the word "walrus" or the word "otter"?) given distributions over both.

To preserve experimental similarity, we used the encoder-decoder models described in McCoy et al [9]. They tested four types of models: the encoder and decoder could either both be LSTMs or GRUs, and for each of these options they tested with and without attention. After some preliminary results where models without attention performed poorly on many iterations of the fronting task (as McCoy et al also report), we are only considering models with attention for the expanded experimentation, and report results only for the LSTM model. We reconstructed the models with the same hyperparameters described in the paper: The encoder and the decoder each have their own embedding layer which maps between words and 256-dimensional vectors [1]. The hidden and cell size of the recurrent modules is also 256 dimensions.

The results that McCoy et al present show that the GRU with attention is moderately biased to be able to correctly identify the main auxiliary in the generalization set. They evaluated the models by checking if the model's output had as its first word the correct auxiliary [2]. On this metric, the GRU with attention got around 75% of the generalization sentences correct while all the other models performed very poorly, getting an average of very close to 0%.

[1] We are aware that this dimensionality is large for our problem. 256 dimensions are often used in language models trained on real data but in our case, as in McCoy et al, the data is constructed from a context-free grammar with a vocabulary of under 100 words. However, we preserve this aspect of McCoy's architecture for a more controlled experimental setup in the same vein.

[2] So, for example, the sentence "I will go to the beach" could be transformed into "Will I beach beach beach beach" and be correct. They argued, quite reasonably, that it is not necessary to evaluate with exact match as language models have various dimensions of failure (such as getting stuck in a one-word loop as above) and they could still be extracting relevant hierarchical information even if things like that happen.

3.2 Analyzing the Replication Results

We could not reproduce McCoy's results. In replicating the experiments, we found that the LSTM with attention performed consistently better than the GRU with attention. Depending on the probability distribution of the training set grammar (which we varied as McCoy et al did not state the probabilities), the LSTM could perform from around 50% to almost perfectly on the generalization set if evaluated by their metric. However, when we looked at the outputs of the models, we saw clear cues that even the best-performing models did not have a hierarchical understanding of grammar as we would define it. Sentences in the generalization set all had a relative clause on the subject, and the model consistently performed a transformation of the following type:

(2) The girl **who might visit** will love these cats → Will the girl love these cats **who might visit**

This transformation is correct by McCoy et al's metric, and the fact that the model can do it shows that it had enough structural awareness enough to know that "who might visit" is a separate clause, and was not distracted by the other auxiliary "might". However, we can see from example (3) above that such structural awareness does not translate to a higher-level grammatical awareness. The relative clause "who might visit" is subordinate to the subject NP about the girl. The model does not preserve this, instead taking it to the end of the sentence.

Hierarchical awareness is a structured understanding of data into multiple levels. The model seems to have a structural awareness of which clauses are separate, and as such it can pick out the clause "who might visit". However, it does not exhibit an awareness of hierarchy. By taking the relative clause from the subject and putting it on the object, the model violates the top-level branch of a parse tree. In fact, even though in English we parse the output as if the relative clause is subordinate to the NP "the cats", it is not clear that the model recognizes the relative clause as subordinate to anything or just moves it to the end. Awareness of these subtree relations are vital to the notion of hierarchy, and such awareness is consistently not exhibited in the outputs of this experiment.

3.3 Extending Hierarchy Evaluation

We extended McCoy's approach in two ways. Firstly, we expanded the PSG that they used in order to add structural diversity and see its effects on structural awareness. Secondly, we extended McCoy's method of evaluation for structural awareness. We evaluated the results based on more high-level structural awareness. Based on the above observations, that the model often moves the relative clause to different phrases, we developed the VP-integrity metric to check for structural awareness.

The VP-integrity metric leniently checks if a model keeps the main NP and the main VP separate. It checks that the output doesn't have any constituents that should be subordinate to the NP moved to the VP or vice versa. This metric is more globally grammatically aware than McCoy's, as the subject-predicate separation is a basic factor of language and represents the topmost branch in a parse tree. The metric works by taking an output of the model, and comparing it to the gold output in three ways: 1) The sentences have the same root verb 2) the sentences have the same subject and 3) For every word that is in the output, if it is in the gold, it must match the gold sentence in whether that word is in the main NP or the main VP.

We also evaluate the model by examining discrepancies in the probabilities it assigns different structures of grammatical and ungrammatical outputs. We created different types of outputs for the model to score, trying to narrow down on what types of changes would cause a model to drastically change its score for an output.

4 Experiments

Experiment 1: Replication

Our first experiments used data from the phrase structure grammar described in the supplementary materials of McCoy et al [9]. McCoy et al do not state probabilities for their grammar, so if there are two rules for a constituent it is not clear how often one is followed versus the other. They state two rules for making an RC, one with just an intransitive VP, and one with an NP and a transitive VP: 1) RC → Rel Aux V_intrans 2) RC → Rel NP Aux V_trans. These two rules create the two

different styles of RC in "The cat **who can swim**" and "The cat **who the dog will chase**". Given that the grammar does not include any adverbs, and all of the verb phrases in the grammar are realized with auxiliaries, note that rule 1 can only produce three-word relative clauses. We ran the experiment for 11 different probability settings in the grammar, ranging the probability of the RC having an NP from 0 to 1 in intervals of 0.1.

Generalization Set 1	Subject has intransitive RC	The girl who might visit will love those cats
Generalization Set 2	Subject and object have intransitive RC	The girl who might visit will love those cats that are grey
Generalization Set 3	Subject has transitive RC	The girl who I might invite will love those cats
Generalization Set 4	Sentence is intransitive, subject has intransitive RC	The cats that can swim will play

Figure 1: The four different generalization sets

We trained every network on 120,000 sentences generated on the grammar described above, which is the same size of training data used in McCoy et al. Half of the training points ended with a flag that asked the model to do auxiliary fronting, and half with a flag that asked the model to reproduce the same sentence again. The models trained until there was no improvement on the dev set (distributed identically to the training set) for 20,000 iterations. We split the generalization set into four distinct categories as described in Figure 1. For evaluation we used the accuracy metric that McCoy et al use: if the model's output has the same first word as the gold output, then that output has an accuracy of 1. Otherwise, it has an accuracy of 0.

Figure 2 gives the results of the different grammars on the generalization sets. All of the models achieved near 100% dev accuracy, which is quite logical given the fact that the problem and language are quite simple, and models of this size can model even real language with some accuracy. While McCoy et al report that initialization had a large effect on the results, we do not see this as the standard deviation across random initializations is quite small in most cases.

Models trained on data with mostly intransitive RCs performed very well on generalization sets 1 and 2 which had RCs of this type. However, we see that models trained with higher percentages of more complicated RCs do not do well on these sets, suggesting that the former models depend on some heuristic to skip over the RCs rather than relying on a syntactic parse. The grammar that performs best across all generalization sets is the grammar with 20% complex RCs. The discrepancy of performance between the different generalization sets, with models consistently performing worse than random on Set 4, suggest that the model had not acquired a representation of the grammar, but instead had acquired tools to help with specific forms.

Experiment 2: Grammar Expansion

The phrase structure grammar used in McCoy et al was quite limited. We expanded the phrase structure grammar, to make it incrementally more like English. We tried to make the grammar more complex, so as to take away some of the heuristics that the model may have latched on to in the original grammar. For example, we made it so that the relative clause can be in positions other than the end by allowing PPs to come after it, or allowing fronted prepositions that contain RCs as in "In the fort that I will build my cat can relax". We also added adjectives on the NPs and adverbs on the VPs, making the length of clauses less deterministic. We also tried another approach: expanding the task. We trained the model on three different types of question formation: Auxiliary fronting ("Will the cat eat the mouse?"), object wh-fronting ("Who will the cat eat?") and subject wh-fronting ("Who will eat the mouse?"), signaling with a flag at the end of each sentence which one to perform.

For all of these grammars, we trained a model from scratch and we also tried pretraining the decoder as a language model that generated complicated NPs. We thought the pretraining might give the model a bias to treat the NP as a constituent that could appear in many places, and therefore be biased towards a more hierarchical and recursive view. Pretraining did not yield any significant improvements on any front, though it did as expected cause a faster convergence when training the whole model. We do not present the results of the models that were pretrained and fine-tuned, as they are very similar to the results of the ones trained from scratch.

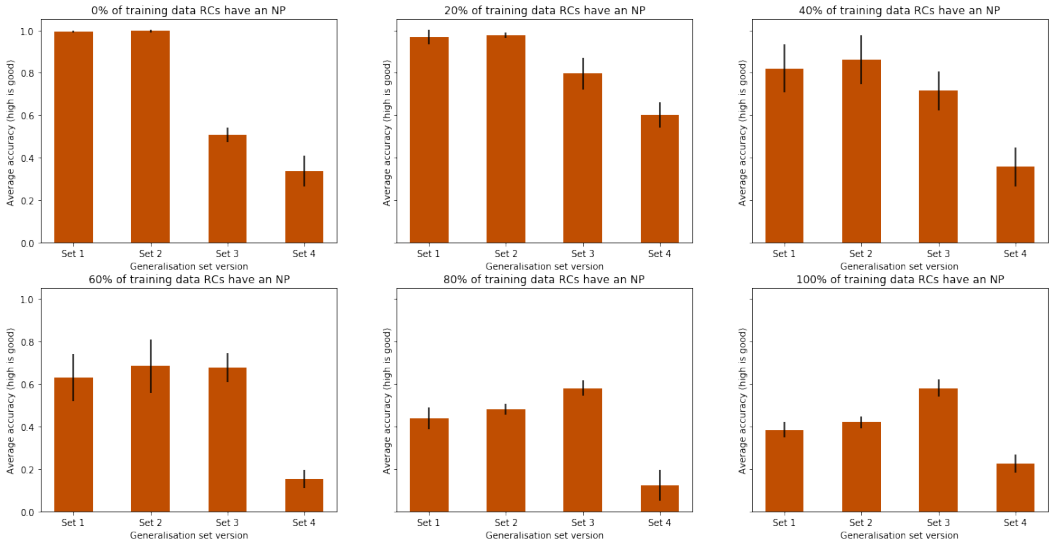

Figure 2: The accuracies of LSTMs trained on data from six different phrase structure grammar probabilities. Each model is tested on the four generalization sets described in Figure 1. Each model was trained five times with different random initialization, and error bars show std deviation across initializations

Figure 3 shows the performance of the models evaluated on McCoy's first-word metric. The worst results are when trained on Grammar 4, where the RC moved around the most in the training data, and Grammar 6, where the task were more complicated. This suggests again that the performance on the other grammars is rooted in heuristics about RC placement, rather than a structural understanding about the RC's role in a sentence.

Number	Description	Example Sentence	Accuracy on first word metric (%)
1	No RCs and RC on object	The cat can eat the mouse who might like cheese.	99.98
2	PP could appear after RC on object	The cat can eat the mouse who might like cheese in my house.	99.89
3	Sentences could have fronted PP with an RC	By the cat that is friendly, the mouse can roam.	99.52
4	RC on fronted PP could have subject	In the house where the cat will live the mouse can't roam	71.45
5	Like 4, but the preposition is kept fronted even in the question	In the house where the cat will live can't the mouse roam?	90.91
6	Included the task of wh-fronting in the training data.	Who will the cat eat?	82.52

Figure 3: The results on expanded grammars

Prompted by the observation in section 3.2 that the models often erroneously moved the RC clause, we also evaluated the outputs using our VP-integrity metric described in section 3.3. All of the above experiments got under 1% accuracy on our metric with the differences being minute and probably random, so it would be spurious to report them.

Experiment 3: Analysis of Hierarchy Likelihood

In order to more directly probe our models' internal distributions for structural awareness, we tested how they scored outputs that were grammatically plausible versus ones that were not. So far, we have

only accessed models' internal distribution through the decoder's generation algorithm [3]. However, it may be the case that the model is in fact encoding structural information in some ways, and those encodings are overridden by linear heuristics that are very strong because of the regular patterns in the data. For example, there may be a linear bias to putting the relative clause at the end since it appears near the end in most of the training data. This does not mean that there is no structural information about what the relative clause is subordinate to, only that there is bias to listen to the linear heuristic instead.

To test for syntactic sensitivity, we created a data set for the trained models to score. The inputs in the scoring set all had an RC on the subject like the generalization inputs, and each input was matched with four possible outputs. One was the correct output, the other had the relative clause moved to the end, and the other two had the relative clause inserted somewhere where it would not make sense grammatically to insert it. The scoring set consisted of 200 such sentences. Figure 4 shows an example entry.

Input	The cat that can swim will catch the fish
Correct output	Will the cat that can swim catch the fish?
Grammatical output, wrong RC placement	Will the cat catch the fish that can swim?
Ungrammatical output 1	*Will the cat catch that can swim the fish?
Ungrammatical output 2	*Will the cat catch the that can swim fish?

Figure 4: An example entry in the scoring set

For each model described in Experiment 2, we collected the scores that it assigned to each of the categories of output (correct, movedRC, ungrammatical1, ungrammatical2), getting the model's likelihood of those outputs given 200 different inputs. This aligned data allowed us to run paired significance tests [4]. All of the models assigned higher probability to the RC at the end than to the correct one, which would be expected given the fact that they consistently outputted this. More importantly, **all of the models assigned a significantly higher probability to the correct output than to the two ungrammatical outputs** ($p < 0.01$), even though they had never seen anything of the form of the correct output during training.

The fact that the models consider the correct RC placement significantly more likely than the ungrammatical RC placement suggests that they have encoded higher-level grammatical information. There were no instances in any of the training examples where the subject NP took a subordinate RC. Nevertheless, the models predict that subjects can take an RC in a similar place where the training data shows the object NP take RCs. This notion of recursivity, that the same constituent can appear in many places in a grammar, is a key element of human grammatical structure that previous probes fail to test for.

The fact that the model has preferences about where an RC is allowed to be inserted suggests more generally that the model has a global structured representation of the sentence. We do not have enough information from the current probes to know the nature of this representation. However, the model certainly predicts that there are spots where the link between adjacent words is weak enough that an RC can be inserted, and others (such as between a determiner and a noun as in the ungrammatical 2 example) where this is much less likely to happen. This reflects the fact that there are places in a parse tree where a subordinate clause can be introduced and places where there is a direct link between two items and no rule could insert something between them.

There were no significant differences between models. That is, no model was more likely to discard ungrammatical clauses than any other. This shows that the grammatical expansions did not affect the structural sensitivity that this probe exposes.

[3] The models we train do not use BEAM search or any complex decoding algorithm. Since this is a probing paper, and not an attempt to create a good model, we considered it more productive to directly analyze the decoder likelihoods as we are doing here than to try to use them to optimize the output.

[4] The most common example used to describe the paired significance test is the case of "two midterms". If we have the same set of students take the two midterms in a class, we look at their scores and ask whether one of the midterms was significantly harder than the other. This is a different question than the more common independent sample significance test, which would be the equivalent of looking at the scores of males and females who took one test and determining if one group had a higher score

5 Analysis

We provide sample outputs from the above experiments, to discuss possible reasons that the results were as presented. For Experiment 1, the outputs support the hypothesis that the model relies on heuristic rather than hierarchical information to perform the fronting task. Firstly, as mentioned in section 3.2, we noticed that all of the models on all of the generalization sets consistently moved the whole relative clause to the end of the sentence. This shows that the models most likely encoded the relative clause as placed somewhere (at the end), rather than as subordinate to another clause.

We more closely examine the outputs of generalization set 4 in Experiment 1, as set 4 with the intransitive verb data caused very bad performance on all of the models. We take the input "Her peacock that your quails will entertain can smile". The model which was trained on only intransitive RCs transformed this input to "Will her peacock entertain her quails that can smile?". The model fashions an intransitive RC using the intransitive main verb, and moves it to the end. This supports the idea that the model had more of a pattern-matching approach to what an RC could be, rather than a hierarchical understanding of which words make up which clauses in what structure from its input.

Conversely, the model trained with 100% transitive RCs outputted the question "Will her quails entertain her peacock who your yaks can entertain?". Like the other model, it fashioned an RC that did not exist, even introducing a new noun to make the RC transitive, and ignoring the intransitive verb while repeating the transitive verb. Similarly to the previous example, the output shows more of a pattern-matching approach than a hierarchical one.

To analyze the results of Experiment 3, we took an input and broke down the probabilities that a model assigned to each word of an output for the four different kind of outputs described in Figure 4. Figure 5 presents a visualization of the results for the input "My walrus who can swim will amuse the raven". We observe that the model was surprised every time for the relativizer "who", except in the moved RC example. There is in fact a huge discrepancy between the probability of "who" in the correct example and the moved RC example. This discrepancy suggests that the model does not see the two NPs as two instances of the same structural unit, again providing some evidence that its understanding of the training language was more template-based than recursive and hierarchical. However, we do see that it is very surprised for the wrong outputs both when the RC starts and when the RC ends, suggesting that it does have a notion of where the RC is allowed to enter the parse tree of the dominating clause.

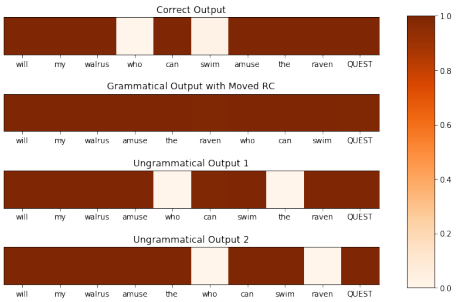

Figure 5: The probabilities the LSTM gave to each word in the output, conditioned on the input string "My walrus who can swim will giggle"

6 Conclusion

We argue that showing long-distance dependency awareness is not equivalent to showing hierarchical grammar awareness in a model. We identify two key aspects of hierarchy: subordination and recursion. Subordination awareness is knowing that there are relationships between clauses that make one a subtree of the other (this VP is part of this RC which is part of this NP etc). Recursion is recognizing that the same structural unit can appear in multiple places in a grammar. The ability to recognize long-distance dependencies does not necessarily imply awareness of these two key components of hierarchy.

Using the experimental framework laid out by McCoy et al, we showed that though the models were often successfully ignoring the RC on the subject, they did not exhibit awareness of the RC's place in the phrase structure as a whole. We expanded the grammar specified by McCoy et al to make the experiments more realistic, and saw that the models still failed to keep the RC subordinate to the correct phrase.

However, we also take a step in the direction of creating new experiments that probe for more global hierarchical knowledge. We see that the models do in fact have a more global structural sensitivity, showing awareness of which RC placements are grammatical between different options that they had never seen during training. This method still has some level of opacity. It is not possible to make definitive claims about what (if any) hierarchical information the model is actually encoding. Nevertheless, we provide a more robust theoretical and methodological framework to understand syntactic awareness in RNNs than only focusing on long-range dependencies.

Mentor: Chris Manning

References

[1] Jean-Philippe Bernardy and Shalom Lappin. Using deep neural networks to learn syntactic agreement. *LiLT (Linguistic Issues in Language Technology)*, 15, 2017.

[2] Robert C Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. Poverty of the stimulus revisited. *Cognitive Science*, 35(7):1207–1242, 2011.

[3] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018.

[4] Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. *CoRR*, abs/1803.11138, 2018.

[5] Julie Anne Legate and Charles D Yang. Empirical re-assessment of stimulus poverty arguments. *The Linguistic Review*, 18(1-2):151–162, 2002.

[6] John D Lewis and Jeffrey L Elman. Learnability and the statistical structure of language: Poverty of stimulus arguments revisited. In *Proceedings of the 26th annual Boston University conference on language development*, volume 1, pages 359–370. Cascadilla Press, 2001.

[7] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

[8] Tal Linzen and Brian Leonard. Distinct patterns of syntactic agreement errors in recurrent networks and humans. *arXiv preprint arXiv:1807.06882*, 2018.

[9] R. Thomas McCoy, Robert Frank, and Tal Linzen. Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks. *CoRR*, abs/1802.09091, 2018.

[10] Amy Perfors, Joshua B Tenenbaum, and Terry Regier. The learnability of abstract syntactic principles. *Cognition*, 118(3):306–338, 2011.