Systematicity in a Recurrent Neural Network by Factorizing Syntax and Semantics

Jacob Russin (jlrussin@ucdavis.edu)
Department of Psychology
UC Davis

Jason Jo
MILA
Université de Montréal

Randall C. O'Reilly
Department of Psychology and Computer Science
Center for Neuroscience
UC Davis

Yoshua Bengio
MILA
Université de Montréal
CIFAR Senior Fellow

Abstract

Standard methods in deep learning fail to capture compositional or systematic structure in their training data, as shown by their inability to generalize outside of the training distribution. However, human learners readily generalize in this way, e.g. by applying known grammatical rules to novel words. The inductive biases that might underlie this powerful cognitive capacity remain unclear. Inspired by work in cognitive science suggesting a functional distinction between systems for syntactic and semantic processing, we implement a modification to an existing deep learning architecture, imposing an analogous separation. The resulting architecture substantially outperforms standard recurrent networks on the SCAN dataset, a compositional generalization task, without any additional supervision. Our work suggests that separating syntactic from semantic learning may be a useful heuristic for capturing compositional structure, and highlights the potential of using cognitive principles to inform inductive biases in deep learning.

Keywords: compositional generalization; systematicity; deep learning; inductive bias; SCAN dataset

Introduction

A crucial property underlying the power of human cognition is its systematicity (Lake, Ullman, Tenenbaum, & Gershman, 2017; Fodor & Pylyshyn, 1988): known concepts can be combined in novel ways according to systematic rules, allowing the number of expressible combinations to grow exponentially in the number of concepts that are learned. Recent work has shown that standard algorithms in deep learning fail to capture this important property: when tested on unseen combinations of known elements, standard models fail to generalize (Lake & Baroni, 2018; Loula, Baroni, & Lake, 2018; Bastings, Baroni, Weston, Cho, & Kiela, 2018). It has been suggested that this failure represents a major deficiency of current deep learning models, especially when they are compared to human learners (Marcus, 2018; Lake et al., 2017; Lake, Linzen, & Baroni, 2019).

A recently published dataset called SCAN (Lake & Baroni, 2018) tests compositional generalization in a sequence-to-sequence (seq2seq) setting by systematically holding out of the training set all inputs containing a basic primitive verb ("jump"), and testing on sequences containing that verb. Success on this difficult problem requires models to generalize knowledge gained about the other primitive verbs ("walk", "run" and "look") to the novel verb "jump," without having seen "jump" in any but the most basic context ("jump" → JUMP). It is trivial for human learners to generalize in this way (e.g., if I tell you that "dax" is a verb, you can generalize its usage to all kinds of constructions, like "dax twice and then dax again", without even knowing what the word means) (Lake & Baroni, 2018; Lake et al., 2019). However, powerful recurrent seq2seq models perform surprisingly poorly on this task (Lake & Baroni, 2018; Bastings et al., 2018).

From a statistical-learning perspective, this failure is quite natural. The neural networks trained on the SCAN task fail to generalize because they have memorized biases that do indeed exist in the training set. Because "jump" has never been seen with any adverb, it would not be irrational for a learner to assume that "jump twice" is an invalid sentence in this language. The SCAN task requires networks to make an inferential leap about the entire structure of part of the distribution that they have not seen — that is, it requires them to make an out-of-domain (o.o.d.) *extrapolation* (Marcus, 2018), rather than merely *interpolate* according to the assumption that train and test data are independent and identically distributed (i.i.d.) (see left part of Figure 3). Seen another way, the SCAN task and its analogues in human learning (e.g., "dax"), require models *not* to learn some of the correlations that are actually present in the training data (Kriete, Noelle, Cohen, & O'Reilly, 2013). To the extent that humans can perform well on certain kinds of o.o.d. tests, they must be utilizing inductive biases that are lacking in current deep learning models (Battaglia et al., 2018).

It has long been suggested that the human capacity for systematic generalization is linked to mechanisms for processing syntax, and their functional separation from the meanings of individual words (Chomsky, 1957; Fodor & Pylyshyn, 1988). Furthermore, recent work in cognitive and computational neuroscience suggests that human learners may factorize knowledge about structure and content, and that this may be important for their ability to generalize to novel combinations (Behrens et al., 2018; Ranganath & Ritchey, 2012). In this work, we take inspiration from these ideas and explore operationalizing a separation between structure and content as an inductive bias within a deep-learning attention mechanism (Bahdanau, Cho, & Bengio, 2015). The resulting architecture, which we call *Syntactic Attention*, separates structural learning about the alignment of input words to target actions (which can be seen as a rough analogue of syntax in the seq2seq setting) from learning about the meanings of individual words (in terms of their corresponding actions).

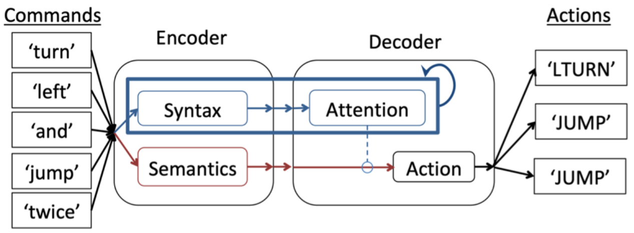

Figure 1: Syntactic Attention architecture. Syntactic and semantic information are maintained in separate streams. The semantic stream is used to directly produce actions, and processes words with a simple linear transformation, so that sequential information is not maintained. The syntactic stream processes inputs with a recurrent neural network, allowing it to capture temporal dependencies between words. This stream determines the attention over semantic representations at each time step during decoding.

The modified attention mechanism achieves substantially improved compositional generalization performance over standard recurrent networks on the SCAN task.

An important contribution of this work is in showing how changes in the connectivity of a network can shape its learning in order to develop a separation between structure and content, without any direct manual imposition of this separation per se. These changes act as an inductive bias or soft constraint that only manifests itself through learning. Furthermore, our model shows that attentional modulation can provide a mechanism for structural representations to control processing in the content pathway, similar to how spatial attention in the dorsal visual pathway can generically modulate object-recognition processing in the ventral visual stream (O'Reilly, Wyatte, & Rohrlich, 2017). Thus, attentional modulation may be critical for enabling structure-sensitive processing — a natural property of symbolic models — to be realized in neural hardware. This provides a more purely neural framework for achieving systematicity, compared to hybrid approaches that combine symbolic and neural network mechanisms (e.g., Yi et al., 2018).

Model

The Syntactic Attention model improves the compositional generalization capability of an existing attention mechanism (Bahdanau et al., 2015) by implementing two separate streams of information processing for syntax and semantics (see Figure 1). In the seq2seq setting, we operationalize 'semantics' to mean the information in each word in the input that determines its *meaning* in terms of target outputs, and we operationalize 'syntax' to mean the information contained in the input sequence that should determine the structure of the *alignment* of input to target words. We describe the mechanisms of this separation and the other details of the model below, following the notation of (Bahdanau et al., 2015), where possible.

Factorizing Syntax and Semantics in Seq2seq

In the seq2seq setting, models must learn a mapping from arbitrary-length sequences of inputs $\mathbf{x} = \{x_1, x_2, ..., x_{T_x}\}$ to arbitrary-length sequences of outputs $\mathbf{y} = \{y_1, y_2, ..., y_{T_y}\}$: $p(\mathbf{y}|\mathbf{x})$. In the SCAN task, the inputs are a sequence of instructions, and the outputs are a sequence of actions. The attention mechanism of Bahdanau et al. (2015) models the conditional probability of each target action given the input sequence and previous targets: $p(y_i|y_1, y_2, ..., y_{i-1}, \mathbf{x})$. This is accomplished by processing the instructions with a recurrent neural network (RNN) in an encoder. The outputs of this RNN are used both for encoding individual words for subsequent translation, and for determining their alignment to actions during decoding.

The underlying assumption made by the Syntactic Attention architecture is that the dependence of target actions on the input sequence can be separated into two independent factors. One factor, $p(y_i|x_j)$, which we refer to as "semantics," models the conditional distribution from individual words in the input to individual actions in the target. Note that, unlike in the model of Bahdanau et al. (2015), these x_j do not contain any information about the other words in the input sequence because they are not processed with an RNN. They are "semantic" in the sense that they contain the information relevant to translating the instruction words into corresponding actions. The other factor, $p(j \rightarrow i|\mathbf{x}, y_{1:i-1})$, which we refer to as "syntax," models the conditional probability that word j in the input is relevant to word i in the action sequence, given the entire set of instructions. This is the alignment of words in the instructions to particular steps in the action sequence, and is accomplished by computing the attention weights over the instruction words at each step in the action sequence using encodings from an RNN. This factor is "syntactic" in the sense that it must capture all of the temporal dependencies in the instructions that are relevant to determining the serial order of outputs (e.g., what should be done "twice", etc.). The crucial architectural assumption, then, is that any temporal dependency between individual words in the instructions that can be captured by an RNN should largely be relevant to their alignment to words in the target sequence, and less relevant to the meanings of individual words. We argue that this can be seen as a factorization of syntax and semantics, because the grammatical rules governing the composition of instruction words' meanings (e.g., how adverbs modify verbs) must be learned in a module that does not have access to those meanings. This assumption will be made clearer in the model description below.

Encoder

The encoder produces two separate vector representations for each word in the input sequence. Unlike the previous attention model (Bahdanau et al., 2015), we separately extract the semantic information from each word with a linear transformation:

$$m_j = W_m x_j \tag{1}$$

where W_m is a learned weight matrix that multiplies the one-hot encodings $\{x_1, ..., x_{T_x}\}$. This weight matrix W_m can be thought of as extracting the information from the inputs that will be relevant to translating individual words into their corresponding actions (e.g. ”jump” → JUMP).

As in the previous attention mechanism (Bahdanau et al., 2015), we use a bidirectional RNN (biRNN) to extract what we now interpret as the syntactic information from each word in the input sequence. The biRNN processes the (one-hot) input vectors $\{x_1, ..., x_{T_x}\}$ and produces a hidden-state vector for each word on the forward pass, $(\overrightarrow{h_1}, ..., \overrightarrow{h_{T_x}})$, and a hidden-state vector for each word on the backward pass, $(\overleftarrow{h_1}, ..., \overleftarrow{h_{T_x}})$. The syntactic information (or “annotations” (Bahdanau et al., 2015)) of each word x_j is determined by the two vectors $\overrightarrow{h_{j-1}}$, $\overleftarrow{h_{j+1}}$ corresponding to the words surrounding it:

$$h_j = [\overrightarrow{h_{j-1}}; \overleftarrow{h_{j+1}}] \tag{2}$$

In all experiments, we used a bidirectional Long Short-Term Memory (LSTM) for this purpose. These representations h_j differ from the previous model in that only the surrounding words are used to infer the relevant syntactic information about each input. Our motivation for doing this was to encourage the encoder to rely on the role each word plays in the input sentence. Note that because there is no sequence information in the semantic representations, all of the information required to parse (i.e., align) the input sequence correctly (e.g., phrase structure, modifying relationships, etc.) must be encoded by the biRNN.

Decoder

The decoder models the conditional probability of each target word given the input and the previous targets: $p(y_i|y_1, y_2, ..., y_{i-1}, \mathbf{x})$, where y_i is the target action and $\mathbf{x}$ is the whole instruction sequence. As in the previous model, we use an RNN to determine an attention distribution over the inputs at each time step (i.e., to align words in the input to the current action). However, our decoder diverges from this model in that the mapping from inputs to outputs is performed from a weighted average of the *semantic* representations of the input words:

$$d_i = \sum_{j=1}^{T_x} \alpha_{ij} m_j \qquad p(y_i|y_1, y_2, ..., y_{i-1}, \mathbf{x}) = f(d_i) \tag{3}$$

where f is parameterized by a linear function with a softmax nonlinearity, and the α_{ij} are the weights determined by the attention model. The softmax in f produces a distribution over the possible actions. We note again that the m_j are produced directly from corresponding x_j, and do not depend on the other inputs. The attention weights are computed by a function measuring how well the syntactic information of a given word in the input sequence aligns with the current hidden state of the decoder RNN, s_i:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \qquad e_{ij} = a(s_i, h_j) \tag{4}$$

where e_{ij} can be thought of as measuring the importance of a given input word x_j to the current action y_i, and s_i is the current hidden state of the decoder RNN. Bahdanau et al. (2015) model the function a with a feedforward network, but we choose to use a simple dot product:

$$a(s_i, h_j) = s_i \cdot h_j, \tag{5}$$

relying on the end-to-end backpropagation during training to allow the model to learn to make appropriate use of this function. Finally, the hidden state of the RNN is updated with the same weighted combination of the *syntactic* representations of the inputs:

$$s_i = g(s_{i-1}, c_i) \qquad c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{6}$$

where g is the decoder RNN, s_i is the current hidden state, and c_i can be thought of as the information in the attended words that can be used to determine what to attend to on the next time step. Again, in all experiments an LSTM was used.

Simulations

SCAN task

The SCAN[1] task was specifically designed to test compositional generalization (see figure 2). In the task, sequences of commands (e.g., “jump twice”) must be mapped to sequences of actions (e.g., JUMP JUMP), and is generated from a simple finite phrase-structure grammar that includes things like adverbs and conjunctions (Lake & Baroni, 2018). The splits of the dataset include: **1) Simple split**, where training and testing data are split randomly, **2) Length split**, where training includes only shorter sequences, and **3) Add primitive split**, where a primitive command (e.g., “turn left” or “jump”) is held out of the training set, except in its most basic form (e.g., “jump” → JUMP).

Here we focus on the most difficult problem in the SCAN dataset, the add-jump split, where “jump” is held out of the training set. The best test accuracy reported in the original paper (Lake & Baroni, 2018), using basic seq2seq models, was 1.2%. More recent work has tested other kinds of seq2seq models, including Gated Recurrent Units (GRU) augmented with attention (Bastings et al., 2018), convolutional neural networks (CNNs) (Dessì & Baroni, 2019), meta-seq2seq (Lake, 2019), and a novel architecture (Li, Zhao, Wang, & Hestness, 2019). Here, we compare the Syntactic Attention model to the best previously reported results.

Implementation details

Train and test sets were kept as they were in the original dataset, but following Bastings et al. (2018), we used early stopping by validating on a 20% held out sample of the training set. All reported results are from runs of 200,000 iterations with a batch size of 1. Unless stated otherwise, each

[1] The SCAN task can be downloaded at `https://github.com/brendenlake/SCAN`

jump	⇒	JUMP
jump left	⇒	LTURN JUMP
jump around right	⇒	RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP
turn left twice	⇒	LTURN LTURN
jump thrice	⇒	JUMP JUMP JUMP
jump opposite left and walk thrice	⇒	LTURN LTURN JUMP WALK WALK WALK
jump opposite left after walk around left	⇒	LTURN WALK LTURN WALK LTURN WALK LTURN WALK LTURN LTURN JUMP

Figure 2: Examples from the SCAN dataset. Details about the detaset can be found in (Lake & Baroni, 2018). Figure reproduced from (Lake & Baroni, 2018).

architecture was trained 5 times with different random seeds for initialization, to measure variability in results. All experiments were implemented in PyTorch. Our best model used LSTMs, with 2 layers and 200 hidden units in the encoder, and 1 layer and 400 hidden units in the decoder, and 120-dimensional vectors for the semantic representations, m_j. The model included a dropout rate of 0.5, and was optimized using an Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.001.

Results

The Syntactic Attention model improves compositional generalization performance on the standard seq2seq SCAN dataset (see table 1). The table shows results (mean test accuracy (%) ± standard deviation) on the test splits of the dataset. Syntactic Attention is compared to the previous best models, which were a CNN (Dessì & Baroni, 2019), and GRUs augmented with an attention mechanism (+ attn), which either included or did not include a dependency (- dep) in the decoder on the previous action (Bastings et al., 2018). Transformers (Vaswani et al., 2017) were not included in our experiments, but have been shown to suffer similar problems with compositional generalization on the SCAN dataset (Keysers et al., 2020).

The best model from the hyperparameter search showed strong compositional generalization performance, attaining a mean accuracy of 91.1% (median = 98.5%) on the test set of the add-jump split. However, as in Dessì and Baroni (2019), we found that our model showed variance across initialization seeds. For this reason, we ran the best model 25 times on the add-jump split to get a more accurate assessment of performance. These results were highly skewed, with a mean accuracy of 78.4 % but a median of 91.0 %. Overall, this represents an improvement over the best previously reported results from standard seq2seq models in this task (Bastings et al., 2018; Dessì & Baroni, 2019).

Recently, Lake (2019) showed that a meta-learning architecture using an external memory achieves 99.95% accuracy on a meta-seq2seq version of the SCAN task. In this version, models are trained to learn how to generalize systematically across a number of variants of a compositional seq2seq problem. Here, we focus on the standard seq2seq version, which limits the model to one training set.

We also report the newer results of Li et al. (2019), which was work done concurrently with ours using a very similar approach. These results are very consistent with our own, and, taken together, lend support to the idea that separating mechanisms for learning syntactic information from mechanisms for learning the meanings of individual words can encourage systematicity in neural networks.

Additional experiments

To test our hypothesis that compositional generalization requires a separation between syntax (i.e., sequential information used for alignment), and semantics (i.e., the mapping from individual instruction words to individual actions), we conducted two more experiments:

- *Sequential semantics*. An additional biLSTM was used to process the semantics of the sentence: $m_j = [\overrightarrow{m_j};\overleftarrow{m_j}]$, where $\overrightarrow{m_j}$ and $\overleftarrow{m_j}$ are the vectors produced for the input word x_j by a biLSTM on the forward and backward passes, respectively. These m_j replace those generated by the simple linear layer in the Syntactic Attention model (in equation (1)).
- *Syntax-action*. Syntactic information was allowed to directly influence the output at each time step in the decoder: $p(y_i|y_1, y_2, ..., y_{i-1}, \mathbf{x}) = f([d_i; c_i])$, where again f is parameterized with a linear function and a softmax output nonlinearity.

The results of the additional experiments (mean test accuracy (%) ± standard deviations) are shown in table 2. These results partially confirmed our hypothesis: performance on the add-jump test set was worse when the strict separation between syntax and semantics was violated by allowing sequential information to be processed in the semantic stream. In the *sequential semantics* experiment, the model performed comparably on the simple split (99.3 %) but performed worse on the compositional split even though we augmented its learning capacity by replacing a simple linear transformation with an RNN. This result suggests that this increase in capacity, which corresponded to a violation of the factorization assumption, allowed the model to memorize regularities in the dataset that prohibited systematic generalization during testing.

Table 1: Compositional generalization results. The Syntactic Attention model achieves an improvement on the compositional generalization tasks of the SCAN dataset in the standard seq2seq setting, compared to the standard models (Bastings et al., 2018; Dessì & Baroni, 2019). Recent results from another novel architecture (Li et al., 2019), developed concurrently using very similar principles, are also reported. Star* indicates average of 25 runs with random initializations. Others are averages of 5 runs.

Model	**Simple**	**Length**	**Add turn left**	**Add jump**
GRU + attn (Bastings et al., 2018)	100.0 ± 0.0	18.1 ± 1.1	59.1 ± 16.8	12.5 ± 6.6
GRU + attn - dep (Bastings et al., 2018)	100.0 ± 0.0	17.8 ± 1.7	90.8 ± 3.6	0.7 ± 0.4
CNN (Dessì & Baroni, 2019)	100.0 ± 0.0	-	-	69.2 ± 8.2
Li et al. (2019)	99.9 ± 0.0	20.3 ± 1.1	99.7 ± 0.4	98.8 ± 1.4
Syntactic Attention	100.0 ± 0.0	15.2 ± 0.7	99.9 ± 0.16	78.4* ± 27.4

However, *syntax-action*, which included sequential information produced by a biLSTM (in the syntactic stream) in the final production of actions, maintained good compositional generalization performance. We hypothesize that this was because in this setup, it was easier for the model to learn to use the semantic information to directly translate actions, so it largely ignored the syntactic information. This experiment suggests that the separation between syntax and semantics does not have to be perfectly strict, as long as non-sequential semantic representations are available for direct translation.

Table 2: Results of additional experiments. Again star* indicates average of 25 runs with random initializations.

Model	**Add turn left**	**Add jump**
Sequential semantics	99.4 ± 1.1	42.3 ± 32.7
Syntax-action	98.2 ± 2.2	88.7 ± 14.2
Syntactic Attention	99.9 ± 0.16	78.4* ± 27.4

Related work

The principles of systematicity and compositionality have recently regained the attention of deep learning researchers (Bahdanau et al., 2019; Lake et al., 2017; Lake & Baroni, 2018; Battaglia et al., 2018). In particular, these issues have been explored in the visual-question answering (VQA) setting (Andreas, Rohrbach, Darrell, & Klein, 2016; Hudson & Manning, 2018; Yi et al., 2018). Many of the successful models in this setting learn hand-coded operations (Andreas et al., 2016), use highly specialized components (Hudson & Manning, 2018), or use additional supervision (Yi et al., 2018). In contrast, our model uses standard recurrent networks and simply imposes the additional constraint that mechanisms for syntax and semantics are separated.

Some of the recent research on compositionality in machine learning has had a special focus on the use of attention. For example, in the Compositional Attention Network, built for VQA, a strict separation is maintained between the representations used to encode images and the representations used to encode questions (Hudson & Manning, 2018). This separation is enforced by restricting them to interact only through attention distributions. Our model utilizes a similar restriction, reinforcing the idea that compositionality is enhanced when information from different modules are only allowed to interact through discrete probability distributions.

The results from the meta-seq2seq version of the SCAN task (Lake, 2019) suggest that meta-learning may be a viable approach to inducing compositionality in neural networks. Humans have ample opportunity through a long developmental trajectory to meta-learn the inductive biases that could facilitate compositional generalization, so this is a promising alternative to the work discussed here. However, a key difference in the particular implementation used in that study is that the additional training episodes explicitly demarcate the primitive verbs by permuting their meanings across episodes. In our work, the training is restricted to a single episode in which no such permutation occurs.

The work of Li et al. (2019) was done concurrently with ours; although their presentation is framed slightly differently, we believe very similar principles have motivated their model. There are few differences with our architecture, but their improved results on the SCAN task may be due to their use of additive noise during training. Future work will explore the exact differences with their model and analyze the important factors contributing to differences in results.

Finally, we note that the experiments presented here are limited to the SCAN dataset, which may not completely capture the kinds of compositional generalization that humans regularly manifest. This may be important, as recent work has shown that the extent to which models can generalize outside of their training distribution can depend heavily on the kind of environments in which they are trained (Hill et al., 2020). Recent work has experimented with other compositional generalization problems that may be more realistic (Lake, 2019; Keysers et al., 2020). Future work will identify whether the principles developed in this paper can aid generalization performance in these other settings.

Discussion

The Syntactic Attention model was designed to incorporate principles from cognitive science and neuroscience as inductive biases into a neural network architecture: the mechanisms for learning rule-like or syntactic information are separated (or factorized (Behrens et al., 2018)) from mechanisms for learning semantic information. Our experiments confirm that this simple organizational principle encourages systematicity in recurrent neural networks in the seq2seq setting, as shown by the substantial improvement in the model's performance on the compositional generalization tasks in the SCAN dataset.

The model makes the assumption that the meanings of individual words should be independent of their alignment to actions in the target sequence (i.e., the attention weight applied to each word at each step in the action sequence). To this end, two separate encodings are produced for the words in the input: semantic representations in which each word is not influenced by other words in the sentence, and syntactic representations which are produced by an RNN that could capture temporal dependencies in the input sequence (e.g., modifying relationships, grammatical roles). The syntactic system alone has access to the sequential information in the inputs, but is constrained to influence actions through an attention mechanism only (see Figure 1). These constraints ensure that learning about the meanings of individual words happens independently of learning about the structured relationships *between* words. This encourages systematic generalization because, even if a word has only been encountered in a single context (e.g., "jump" in the add-jump split), as long as its syntactic role is known (e.g., that it is a verb that can be modified by adverbs such as "twice"), it can be used in many other constructions that follow the rules for that syntactic role. Additional experiments confirmed this intuition, showing that when sequential information is allowed to be processed by the semantic system (*sequential semantics*), systematic generalization performance is substantially reduced.

The paradigmatic example of systematicity is a symbolic system in which representational content (e.g., the value of a variable stored in memory) is maintained separately from the computations that are performed on that content. This separation ensures that the *manipulation* of the content stored in variables can be completely independent of the content itself, and will therefore generalize to arbitrary elements. Our model implements an analogous separation, but in a purely neural architecture that does not rely on hand-coded rules or additional supervision. In this way, it can be seen as transforming a difficult out-of-domain (o.o.d.) generalization problem into two separate i.i.d. generalization problems — one where the individual meanings of words are learned, and one where *how words are used* (e.g., how adverbs modify verbs) is learned (see Figure 3). This may be a useful approach to encouraging systematicity in neural networks, which are very good at i.i.d. generalization but generally fail when presented with o.o.d. problems.

Our work shows that a strict separation between syntax and semantics can be useful for encouraging systematicity and allowing for compositional generalization. It is unlikely that the human brain has such a strict separation, but our work builds on related ideas in neuroscience (Behrens et al., 2018) and suggests a useful framework for investigating whether a similar principle may be at work in the human brain. Future work will explore this principle in other settings, e.g. with transformer models (Vaswani et al., 2017), and investigate other ways in which such a separation can be softened while maintaining good compositional generalization performance.

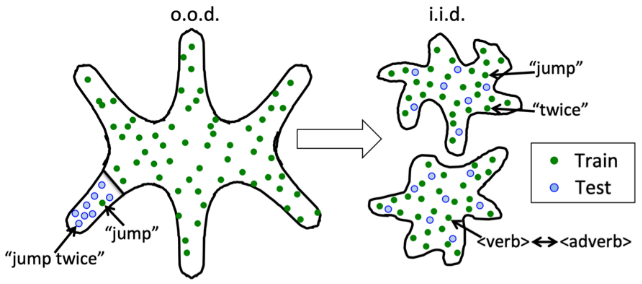

Figure 3: Illustration of the transformation of an out-of-domain (o.o.d.) generalization problem into two independent, identically distributed (i.i.d.) generalization problems. This transformation is accomplished by the Syntactic Attention model without hand-coding grammatical rules or supervising with additional information such as parts-of-speech tags.

Acknowledgments

We would like to thank reviewers for their thorough comments and useful suggestions and references. We would also like to thank all of the members of the Computational Cognitive Neuroscience Lab at UC Davis and the Analogy Group for helpful ongoing discussion on these topics. This work was supported by ONR N00014-19- 1-2684 / N00014-18-1-2116, ONR N00014-14-1-0670 / N00014-16-1-2128, and ONR N00014-18-C-2067.

References

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016, June). Neural Module Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 39–48). Las Vegas, NV, USA: IEEE. doi: 10.1109/CVPR.2016.12

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, CA, USA, may 7-9, 2015, conference track proceedings.*

Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T. H., de Vries, H., & Courville, A. C. (2019). Systematic Generalization: What Is Required and Can It Be Learned? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net.

Bastings, J., Baroni, M., Weston, J., Cho, K., & Kiela, D. (2018, November). Jump to better conclusions:

SCAN both left and right. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (pp. 47–55). Brussels, Belgium: Association for Computational Linguistics. doi: 10.18653/v1/W18-5407

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., … Pascanu, R. (2018, June). Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261 [cs, stat]*.

Behrens, T. E. J., Muller, T. H., Whittington, J. C. R., Mark, S., Baram, A. B., Stachenfeld, K. L., & Kurth-Nelson, Z. (2018, October). What Is a Cognitive Map? Organizing Knowledge for Flexible Behavior. *Neuron*, *100*(2), 490–509. doi: 10.1016/j.neuron.2018.10.002

Chomsky, N. (Ed.). (1957). *Syntactic structures*. The Hague: Mouton & Co.

Dessì, R., & Baroni, M. (2019, July). CNNs found to jump around more skillfully than RNNs: Compositional Generalization in Seq2seq Convolutional Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3919–3923). Florence, Italy: Association for Computational Linguistics. doi: 10.18653/v1/P19-1381

Fodor, J. A., & Pylyshyn, Z. W. (1988, March). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*(1-2), 3–71. doi: 10.1016/0010-0277(88)90031-5

Hill, F., Lampinen, A., Schneider, R., Clark, S., Botvinick, M., McClelland, J. L., & Santoro, A. (2020, February). Environmental drivers of systematicity and generalization in a situated agent. *arXiv:1910.00571 [cs]*.

Hudson, D. A., & Manning, C. D. (2018). Compositional Attention Networks for Machine Reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.* OpenReview.net.

Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., … Bousquet, O. (2020). Measuring compositional generalization: A comprehensive method on realistic data. , 38.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

Kriete, T., Noelle, D. C., Cohen, J. D., & O'Reilly, R. C. (2013, October). Indirection and symbol-like processing in the prefrontal cortex and basal ganglia. *Proceedings of the National Academy of Sciences*, *110*(41), 16390–16395. doi: 10.1073/pnas.1303547110

Lake, B. M. (2019). Compositional generalization through meta sequence-to-sequence learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 9788–9798). Curran Associates, Inc.

Lake, B. M., & Baroni, M. (2018). Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In J. G. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (Vol. 80, pp. 2879–2888). PMLR.

Lake, B. M., Linzen, T., & Baroni, M. (2019). Human few-shot learning of compositional instructions. In A. K. Goel, C. M. Seifert, & C. Freksa (Eds.), *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019* (pp. 611–617). cognitivesciencesociety.org.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017, January). Building machines that learn and think like people. *The Behavioral and Brain Sciences*, *40*, e253. doi: 10.1017/S0140525X16001837

Li, Y., Zhao, L., Wang, J., & Hestness, J. (2019, November). Compositional Generalization for Primitive Substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 4293–4302). Hong Kong, China: Association for Computational Linguistics. doi: 10.18653/v1/D19-1438

Loula, J., Baroni, M., & Lake, B. (2018, November). Rearranging the Familiar: Testing Compositional Generalization in Recurrent Networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (pp. 108–114). Brussels, Belgium: Association for Computational Linguistics. doi: 10.18653/v1/W18-5413

Marcus, G. (2018, January). Deep learning: A critical appraisal.

O'Reilly, R. C., Wyatte, D. R., & Rohrlich, J. (2017, September). Deep predictive learning: A comprehensive model of three visual streams. *arXiv:1709.04654 [q-bio]*.

Ranganath, C., & Ritchey, M. (2012, October). Two cortical systems for memory-guided behaviour. *Nature Reviews Neuroscience*, *13*(10), 713–726.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is All you Need. In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA* (pp. 5998–6008).

Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., & Tenenbaum, J. (2018). Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 31* (pp. 1031–1042). Curran Associates, Inc.