Systematicity Emerges in Transformers when Abstract Grammatical Roles Guide Attention

Ayush Chakravarthy*
Department of Computer Science
University of California Davis
akchakravarthy@ucdavis.edu

Jacob Russin*
Center for Neuroscience
University of California Davis
jlrussin@ucdavis.edu

Randall O'Reilly
Department of Computer Science
Center for Neuroscience
University of California Davis
One Shields Ave. Davis, CA 95616
oreilly@ucdavis.edu

Abstract

Systematicity is thought to be a key inductive bias possessed by humans that is lacking in standard natural language processing systems such as those utilizing transformers. In this work, we investigate the extent to which the failure of transformers on systematic generalization tests can be attributed to a lack of linguistic abstraction in its attention mechanism. We develop a novel modification to the transformer by implementing two separate input streams: a role stream controls the attention distributions (i.e., queries and keys) at each layer, and a filler stream determines the values. Our results show that when abstract role labels are assigned to input sequences and provided to the role stream, systematic generalization is improved.

1 Introduction

Transformers have achieved state-of-the-art performance on many natural language processing (NLP) tasks (Brown et al., 2020; Devlin et al., 2019; Vaswani et al., 2017), but it has been suggested that they remain inferior to human language learners when it comes to sample efficiency (Linzen, 2020) and more difficult generalization problems (Baroni, 2020; Lake and Baroni, 2018; Lake et al., 2019; Keysers et al., 2020). These architectures have proven to scale remarkably well (Brown et al., 2020), but may lack the strong inductive biases that contribute to these human abilities (Battaglia et al., 2018; Lake et al., 2017).

Systematicity, or the capacity to leverage structural or grammatical knowledge to compose familiar concepts in novel ways (Fodor and Pylyshyn, 1988; Smolensky, 1990), has been highlighted as one potential inductive bias present in humans (Lake et al., 2019; O'Reilly et al., 2021) that deep learning architectures may lack (Lake and Baroni, 2018; Lake et al., 2017). It has been argued that in humans, the ability to understand sentences such as "John loves Mary" necessarily implies the ability to understand certain other sentences, e.g., those that are constructed from the same elements and grammatical relations such as "Mary loves John" (Fodor and Pylyshyn, 1988).

The SCAN dataset (Lake and Baroni, 2018) was introduced to evaluate the systematic generalization capabilities of deep neural networks. In SCAN, instructions generated from an artificial grammar must be translated into action sequences, and train-test splits require models to generalize to novel compositions of familiar words. Although deep learning models achieve good generalization performance when train and test data are split randomly, their performance suffers on these systematic generalization tests (Lake and Baroni, 2018), even though humans perform well on analogous generalization problems (Lake et al., 2019).

The mechanisms underlying human systematicity remain unclear, but a number of candidates have been proposed, including tensor-product representations (Schlag et al., 2019; Smolensky, 1990) and specialized attention mechanisms (Goyal et al., 2019; Bengio, 2017; Russin et al., 2020; Webb et al., 2021). Attention is central to the transformer architecture (Vaswani et al., 2017) and has been leveraged in mechanisms resembling systematic symbolic processing (Graves et al., 2014; Webb et al., 2021), thus making it a key potential target for encouraging systematicity (Russin et al., 2020).

In this work, we explore the connection between attention and systematicity using a novel transformer architecture designed to leverage structural or abstract information in its attention mechanism.

*equal contribution

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, pages 1 - 8

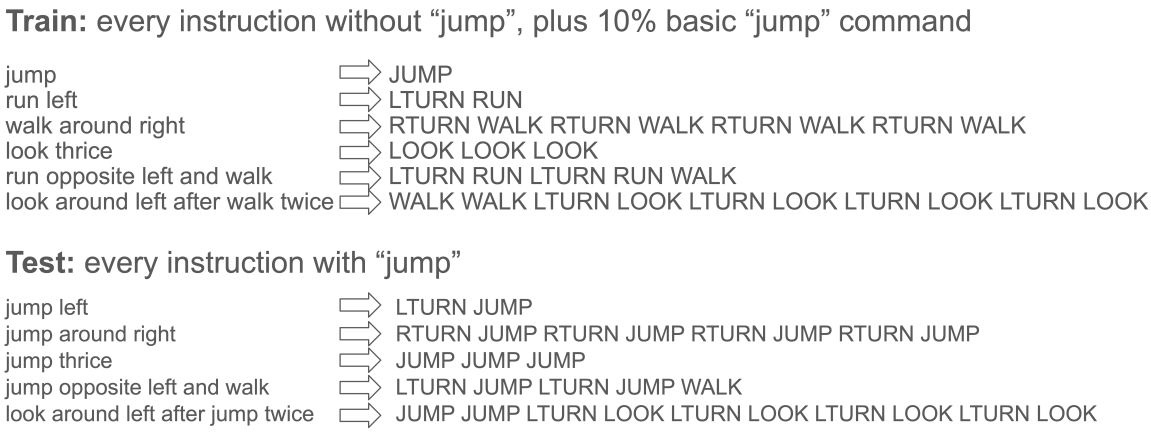

Figure 1: Examples from the add-jump split of SCAN. All except the simplest instructions with the word "jump" are held out of the training set, requiring models to generalize its usage to more complicated constructions.

We hypothesized that systematicity would improve if attention distributions in the transformer were strictly determined from abstract inputs containing minimal token-specific information, as this may prevent memorization of spurious relationships in the training data. Previous work has experimented with incorporating additional linguistic inputs into NLP systems (e.g., Sachan et al., 2021), but here we propose a novel way of utilizing additional linguistic knowledge: a separate "role" input stream is introduced to the transformer, which determines the attention distributions at each layer but is kept separate from the typical ("filler") input stream used to directly generate outputs. Many kinds of information can be passed to the role input stream (including the original tokens themselves), thereby allowing us to explore the kinds of inputs that, when used to determine attention, result in improved systematicity. In our preliminary work, we explore the use of abstract grammatical roles to determine attention in the transformer on the SCAN dataset.

2 Related Work

2.1 SCAN

The SCAN dataset (see Figure 1) uses a simple finite phrase-structure grammar to generate instruction sequences that must be translated into sequences of actions (Lake and Baroni, 2018). In the *simple split*, train and test examples are sampled randomly from the set of all possible instructions. In the systematic generalization test called the *add-jump split*, all instruction sequences containing one of the primitive verbs ("jump") are systematically held out of the training set, except in its simplest form ("jump" → JUMP). The original work showed that recurrent neural networks such as long short-term memory (LSTM) succeed at the simple split but fail on the add-jump split (Lake and Baroni, 2018).

Subsequent work introduced a new framework for generating systematic generalization tests called distribution-based compositionality assessment, and showed that transformers perform poorly on these tests in addition to the original add-jump split (Keysers et al., 2020). Although standard deep learning architectures consistently fail at this task, a number of non-standard approaches have demonstrated some success, including a meta-learning (Lake, 2019), recurrent networks that factorize alignment and translation (Russin et al., 2020) or are designed for primitive substitution (Li et al., 2019), masked language model pretraining (Furrer et al., 2021); iterative back-translation (Guo et al., 2020), use of analytic expressions (Liu et al., 2020), and auxiliary sequence prediction (Jiang and Bansal, 2021). Our preliminary work presents a new approach that has many commonalities with these previous ideas.

2.2 Utilizing Linguistic Knowledge

Prior work has shown that a remarkable amount of linguistic structure emerges in the representations learned by large transformers self-supervised on natural language (Linzen and Baroni, 2021; Manning et al., 2020; Tenney et al., 2019), and that transformers can learn to approximate a compositional process for solving math problems (Russin et al., 2021). These findings may cast doubt on the idea that injecting explicit linguistic structure will aid these models in producing the kinds of system-

atic behavior observed in human language learners. However, given their poor systematic generalization performance observed on tasks like SCAN (Lake and Baroni, 2018), and their reliance on certain syntactic heuristics that lead to predictable failures on challenging sentences (McCoy et al., 2019; Linzen and Baroni, 2021), it stands to reason that these models may benefit from access to explicit linguistic knowledge (Sachan et al., 2021).

Some work has attempted to incorporate linguistically-informed labels such as part-of-speech tags or syntactic parses into the inputs or training regiments of deep learning models (Sachan et al., 2021; Sennrich and Haddow, 2016; Strubell et al., 2018), showing some improvements on machine translation (Sennrich and Haddow, 2016) and semantic role labeling (Strubell et al., 2018). A number of methods have been used to inject linguistic knowledge into these models, including the use of graph neural networks (Marcheggiani and Titov, 2017; Sachan et al., 2021) and multi-task learning (Strubell et al., 2018). In this work, we develop a novel approach that attempts to establish an explicit link between linguistic structure and the attention mechanism of transformers to improve their systematic generalization capabilities.

3 Methods

3.1 Architecture

The transformer architecture (Vaswani et al., 2017) utilizes multi-head attention layers that take as input query (Q), key (K), and value (V) vectors:

$$\text{Attn}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

where d_k is the dimension of the keys (K). Note that the probability distribution over the sequence length produced by the softmax is determined by the queries (Q) and keys (K) alone. We modified the existing transformer architecture by separating two streams of processing (see Figure 2): 1) the "filler" stream determines the values at each layer, which will be averaged according to the weights given by the attention distributions and contribute directly to the output of the model, and 2) the "role" stream determines at each layer the queries (Q) and keys (K) — and therefore the attention distributions — but otherwise does not directly contribute to the output of the model. This was achieved by introducing a separate set of embeddings for each input stream (M for the fillers and X for the roles). The existing attention mechanism was modified so that the roles in layer $l+1$ are determined from a weighted combination of the keys in layer l:

$$\begin{aligned} M &= \text{Attn}(Q, K, V) \\ X &= \text{Attn}(Q, K, K) \end{aligned} \quad (2)$$

This ensures that no information from the filler stream can enter into the determination of the attention distributions at each layer, and that the roles can only affect the output of the model through their control over the attention, similar to Russin et al. (2020). The attention at each layer can have multiple heads in the usual way (Vaswani et al., 2017), and the separation between the two streams is maintained throughout both the encoder and the decoder (see Figure 2). Because the role stream determines the way information from the input tokens will be combined throughout the architecture (through its influence on the attention distributions), positional encodings are added to the role embeddings rather than the filler embeddings.

Note that this setup allows us flexibility in terms of the kind of information that is passed to the role input stream. The original tokens themselves can be embedded separately and passed to the role stream, in which case the architecture becomes very similar to the original transformer, with the exception of the modification to the attention depicted in Figure 2. Here, we embed abstract roles for the tokens in the SCAN dataset to investigate the relationship between abstraction in the attention mechanism and systematic generalization behavior.

3.2 Role Auxiliary Loss

Each transformer layer returns two sets of vectors (X and M). The output of the filler stream (M) is a sequence of target predictions that are used to compute the usual cross entropy loss before backpropagation ("Filler loss"). The output of the role stream (X) can optionally be used in an auxiliary cross-entropy loss on the roles assigned to the target sequence ("Role loss"). We performed experiments with and without this auxiliary loss, and results are reported for both.

3.3 Thresholded Attention

Drawing inspiration from Rahaman et al. (2021), we also experimented with thresholding the encoder-decoder attention:

$$\text{threshold}(A_{ij}) = \begin{cases} A_{ij} & \text{if } A_{ij} > \tau \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

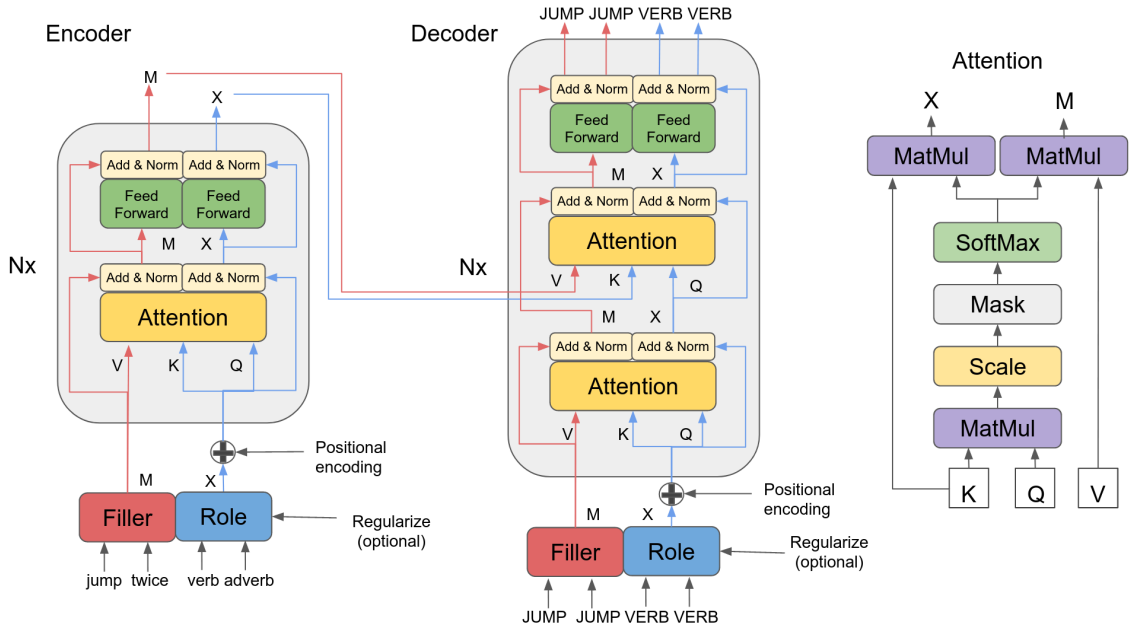

Figure 2: Modified transformer architecture. The architecture imposes two separate role and filler streams throughout the encoder (left) and decoder (middle). The filler stream determines the values (V) at each layer while the role stream determines the keys (K) and queries (Q), and therefore the attention distributions. This was accomplished by modifying the original attention mechanism (right).

Where τ is the attention threshold and $A = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})$. The thresholded attention matrix is then re-normalized and multiplied by the value matrix as in equation 1.

3.4 Implementation Details

The encoder and decoder had 2 layers with 8 attention heads and used a thresholding parameter (τ) of 0.08. The embedding dimension was 256, the hidden dimension was 512, and the dimension of the query, key and value vectors was 256. The model was optimized for 400 epochs using Adam (Kingma and Ba, 2015) with a learning rate of 2.5×10^{-4}. Experiments were performed using both absolute positional encodings (Vaswani et al., 2017) and relative positional embeddings (Dai et al., 2019); absolute positional encodings were found to lead to slightly better performance with reduced variance, so for simplicity we only report those results.

4 Experiments

To test our hypothesized link between attention, linguistic abstraction, and systematic generalization, we developed abstract roles to label each token in the SCAN vocabulary, and performed experiments testing our architecture with and without these abstract roles. We report results on the difficult add-jump split of the SCAN dataset, and compare against previous work. Our main purpose is to show that systematic generalization is improved in the transformer when linguistic abstractions are used as inputs to the role stream for determining attention, and that there is an asymmetry in the transformer such that these abstractions should be used to determine attention (i.e., keys and queries) and not to directly produce outputs (i.e., values).

4.1 SCAN Roles

The phrase-structure grammar used in SCAN is very simple, so the grammatical roles used as additional inputs were relatively straightforward to implement. In the case of the add-jump split, we hypothesized that the best abstract role scheme would be one that assigned all primitive verbs to a single role ("prim") in both the instructions (source) and the actions (target). Except where indicated (section 4.2.2), all results used this scheme.

4.2 Results

Our main results are shown in Table 1. We reproduce previous work and show that the baseline transformer (Vaswani et al., 2017) achieves perfect accuracy on the simple split of the SCAN dataset,

Model	Simple	Add jump
LSTM+Attn (Keysers et al., 2020)	99.9 ± 2.7	0.0 ± 0.0
Syntactic Attention (Russin et al., 2020)	100.0 ± 0.0	78.4 ± 27.4
CGPS-RNN (Li et al., 2019)	99.9 ± 0.0	98.8 ± 1.4
T5-11B (Furrer et al., 2021)	X	98.3 ± 3.3
Semi-Sup (Guo et al., 2020)	X	100.0 ± 0.0
LANE (Liu et al., 2020)	100.0 ± 0.0	100.0 ± 0.0
Aux. seq. (Jiang and Bansal, 2021)	X	98.32 ± 0.3
Transformer	100.0 ± 0.0	0.19 ± 0.18
Filler loss, no thresh (ours)	99.9 ± 0.01	16.2 ± 25.1
Filler loss, thresh (ours)	99.9 ± 0.01	85.6 ± 1.15
Filler + Role loss, no thresh (ours)	99.9 ± 0.02	87.4 ± 5.6
Filler + Role loss, thresh (ours)	100.0 ± 0.0	92.7 ± 3.3

Table 1: Performance (average accuracy $\pm$ standard deviation) on the simple and add-jump splits of SCAN.

but fails dramatically on the add-jump split testing its systematic generalization capabilities. Our architecture improves performance on the add-jump split when the role labels are used as inputs to the role stream. Marginal improvement relative to baseline was observed without the use of attention thresholding and without backpropagating the auxiliary role loss ("Filler loss, no thresh"). Each of these two tweaks improved performance ("Filler loss, thresh", "Filler + Role loss, no thresh") and when both were used ("Filler + Role loss, thresh"), the architecture achieved 92.7% accuracy on the test set of the add-jump split.

4.2.1 Abstraction in Roles vs. Fillers

To further investigate the connection between attention and systematicity, we varied the inputs used in each of the filler and role streams of the architecture (see Table 2). When the filler tokens (i.e., the words from the original SCAN vocabulary) were used as inputs to both the role and filler streams, our architecture resembled the original transformer architecture, as these inputs were used to simultaneously determine the outputs (i.e., the values) and the attention (i.e., the keys and queries) at each layer. This was confirmed in the performance on the SCAN task, where using the fillers in both streams ("Fillers-Fillers") resulted in similar performance to the baseline transformer.

As a sanity check, we also reversed the role and filler inputs, so that the role labels were inputs to the filler stream and the words from the original SCAN vocabulary were used as inputs to the role stream ("Roles-Fillers"). In this case, performance again matched the baseline transformer on the add-jump split, confirming our intuition that linguistic abstractions are best used to determine attention distributions, not values.

Model	Simple	Add jump
Transformer	100.0 ± 0.0	0.19 ± 0.18
Fillers-Fillers	100.0 ± 0.0	2.8 ± 1.6
Roles-Fillers	100.0 ± 0.0	0.22 ± 0.16
Fillers-Roles	100.0 ± 0.0	92.7 ± 3.3

Table 2: Performance on the add-jump split only improved when abstract annotations were used in the role stream ("Fillers-Roles").

4.2.2 Varying the Level of Abstraction

We believe that the previous result highlights a strength of our setup, as it allows us the flexibility to diverge from the original transformer in a continuous way by varying the amount of abstraction used in the inputs to the role stream. For example, in a natural language task it would be possible to vary the kinds of abstract labels or annotations supplied as input to the role stream from highly abstract part-of-speech tags to more complex annotations from more sophisticated automated parses.

To test this idea in the SCAN setting, we experimented with different schemes for assigning roles that varied in their level of abstraction, as measured by the empirical entropy of the resultant source role vocabulary (see Figure 3). After our initial role-assignment scheme, we made roles progressively more abstract by assigning additional instruction words to the same role (e.g., "left" and "right" to "dir", "twice" and "thrice" to "num", etc.). Results validated the assumption that the best scheme was one that used a single role for each of the primitive verbs, and assigned a different role to each of the

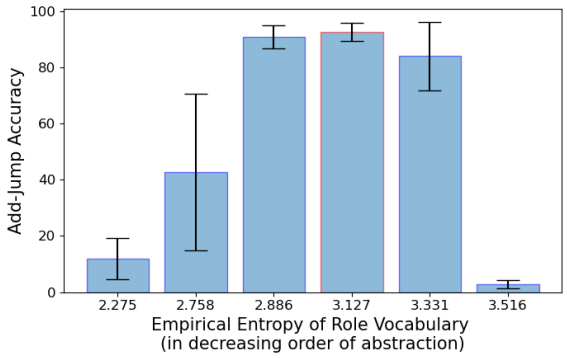

Figure 3: Add-jump performance varies with the level of abstraction in the inputs to the role stream (highest performance outlined in red).

other words (entropy = 3.127). This experiment shows that there is an ideal level of abstraction to use in the role stream: too much abstraction results in an inability to distinguish relevant distinctions, and too little results in the unsystematic memorization typical of the vanilla transformer.

5 Conclusion

Our preliminary work establishes a connection between linguistic abstraction, the attention mechanism used in transformers, and systematic generalization behavior as measured by performance on the SCAN dataset: when abstract roles are assigned to inputs and used to determine the attention at each layer, systematic generalization improves. We developed an architecture that may facilitate greater understanding of the original transformer (Vaswani et al., 2017) by allowing more precise investigation into the relative contributions of attention distributions and representation learning. Future work will test our setup on other compositional or systematic generalization tasks (Keysers et al., 2020; Kim and Linzen, 2020) and determine the kinds of linguistic abstraction that allows success on these tasks. In addition, future work will experiment with using our novel architecture on natural language datasets using varying levels of linguistic abstraction.

The extent to which human-level language understanding requires stronger inductive biases than those currently implemented in deep learning systems remains an open question. Our work shows that utilizing linguistic abstraction in the attention mechanism of transformers may be a promising approach for improving the systematic generalization capabilities of deep neural networks.

Acknowledgements

We would like to thank the members of the Computational Cognitive Neuroscience lab at UC Davis, Paul Smolensky, Roland Fernandez, and other members of the Deep Learning Group at Microsoft Research, as well as reviewers for helpful comments and discussions. The work was supported by: ONR grants ONR N00014-20-1-2578, N00014-19-1-2684 / N00014-18-1-2116, N00014-18-C-2067.

References

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1791):20190307.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261 [cs, stat]*.

Yoshua Bengio. 2017. The Consciousness Prior. *arXiv:1709.08568 [cs, stat]*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the 2019 Conf. of the NA Chapt. of the Assoc. for Comp. Ling.*, pages 4171–4186, Minneapolis, MN, USA. Association for Computational Linguistics.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2021. Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures. *arXiv:2007.08970 [cs]*.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. 2019. Recurrent Independent Mechanisms. *arXiv:1909.10893 [cs, stat]*.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing Machines. *arXiv:1410.5401 [cs]*.

Yinuo Guo, Hualei Zhu, Zeqi Lin, Bei Chen, Jian-Guang Lou, and Dongmei Zhang. 2020. Revisiting Iterative Back-Translation from the Perspective of Compositional Generalization. *arXiv:2012.04276 [cs]*.

Yichen Jiang and Mohit Bansal. 2021. Inducing Transformer's Compositional Generalization Ability via Auxiliary Sequence Prediction Tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6253–6265, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. page 38.

Najoung Kim and Tal Linzen. 2020. COGS: A Compositional Generalization Challenge Based on Semantic Interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

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*.

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

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proc. of the 35th Intern. Conf. on Mach. Lear.*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888, Stockholmsmässan, Stockholm, Sweden. PMLR.

Brenden M. Lake, Tal Linzen, and Marco Baroni. 2019. Human few-shot learning of compositional instructions. In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019*, pages 611–617. cognitivesciencesociety.org.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. *The Behavioral and Brain Sciences*, 40:e253.

Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. 2019. 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)*, pages 4293–4302, Hong Kong, China. Association for Computational Linguistics.

Tal Linzen. 2020. How Can We Accelerate Progress Towards Human-like Linguistic Generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Tal Linzen and Marco Baroni. 2021. Syntactic Structure from Deep Learning. *Annual Review of Linguistics*, 7(1):null.

Qian Liu, Shengnan An, Jian-Guang Lou, Bei Chen, Zeqi Lin, Yan Gao, Bin Zhou, Nanning Zheng, and Dongmei Zhang. 2020. Compositional Generalization by Learning Analytical Expressions. *arXiv:2006.10627 [cs]*.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*.

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. *arXiv:1902.01007 [cs]*.

R. C. O'Reilly, Charan Ranganath, and Jacob L. Russin. 2021. The Structure of Systematicity in the Brain. *arXiv:2108.03387 [q-bio]*.

Nasim Rahaman, Muhammad Waleed Gondal, Shruti Joshi, Peter V. Gehler, Yoshua Bengio, Francesco Locatello, and Bernhard Schölkopf. 2021. Dynamic inference with neural interpreters. *CoRR*, abs/2110.06399.

Jacob Russin, Roland Fernandez, Hamid Palangi, Eric Rosen, Nebojsa Jojic, Paul Smolensky, and Jianfeng Gao. 2021. Compositional Processing Emerges in Neural Networks Solving Math Problems. In *Proceedings for the 43rd Annual Meeting of the Cognitive Science Society*.

Jacob Russin, Jason Jo, Randall C O'Reilly, and Yoshua Bengio. 2020. Systematicity in a Recurrent Neural Network by Factorizing Syntax and Semantics. In *Proceedings for the 42nd Annual Meeting of the Cognitive Science Society*, page 7.

Devendra Singh Sachan, Yuhao Zhang, Peng Qi, and William Hamilton. 2021. Do Syntax Trees Help Pre-trained Transformers Extract Information? *arXiv:2008.09084 [cs]*.

Imanol Schlag, Paul Smolensky, Roland Fernandez, Nebojsa Jojic, Jürgen Schmidhuber, and Jianfeng Gao. 2019. Enhancing the Transformer with Explicit Relational Encoding for Math Problem Solving. *arXiv:1910.06611 [cs, stat]*.

Rico Sennrich and Barry Haddow. 2016. Linguistic Input Features Improve Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2):159–216.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. *arXiv:1905.05950 [cs]*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Taylor W. Webb, Ishan Sinha, and Jonathan D. Cohen. 2021. Emergent Symbols through Binding in External Memory. *arXiv:2012.14601 [cs]*.