Pacific Northwest
NATIONAL LABORATORY

PNNL-33551

MetaText

Compositional Generalization in Deep Language Models

October 2022

Scott Howland
Jessica Yaros
Noriaki Kono

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-33551

MetaText

Compositional Generalization in Deep Language Models

October 2022

Scott Howland
Jessica Yaros
Noriaki Kono

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Compositional generalization, the ability to infer and reason using novel combinations of previously encountered entities and structures, is a trait of great utility across a variety of deep learning tasks; it implies a certain ability to reason using consistent and therefore interpretable rules, to understand the constituent parts of its input at multiple levels of granularity, to be robust to spurious differences between semantically equivalent inputs, and more. We seek to understand the degree to which existing natural language models achieve or fall short of compositional generalization across a variety of tasks, whether there are distinct types of compositional generalization in practice, and to identify avenues of intervention through which we can improve models' compositional generalization ability. After exploring these questions using an array of models, tasks, and potential interventions we see that large, pretrained language models have encountered sufficient training data to account for a variety of fine-grained compositional behavior but still struggle to reason at the level of phrases or larger language structures. Non-intrusive, data-based interventions in the form of augmenting individual sequences with compressed versions of themselves or deriving new examples from induced grammars prove insufficient to encourage greater levels of compositional reasoning, indicating that future work might benefit most from focusing on changes to a model's inductive bias at the architectural or loss level, or by integrating compositionality-boosting data interventions into the large-scale pretraining process itself.

Acknowledgments

This research was supported by the **National Security Mission Seed**, under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

Contents

Abstract ii
Acknowledgments iii
Contents iv
1.0 Introduction 1
2.0 Local and Non-Local Compositionality 2
2.1. Initial Findings – COGS 2
2.2. Recursive Decoding – gSCAN 4
3.0 Data Interventions for Boosting Compositionality 7
3.1. Input Sequence Compression 7
3.2. Data Augmentation via Induced Grammars 8
4.0 References 11

Figures

Figure 1. GSCAN generalization splits with exact match accuracies for prior art and our transformer model. The 'Yellow Squares', 'Red Squares', and 'Class Inference' tasks require local compositional generalization. All other tasks demand global compositionality. 6

Tables

Table 1. Local compositional generalization for randomly initialized and finetuned transformers on COGS generalization splits. The best performing model results are in bold. 3
Table 2. Global compositional generalization for randomly initialized and finetuned transformers on COGS generalization splits. The best performing model results are in bold. 3
Table 3. Exact match accuracies for two key GSCAN splits requiring global compositional generalization. Best performing model results are in bold. 6
Table 4. ROUGE scores on the COGS dataset for transformers with and without supplementary input compression. Best performing model results are in bold. Lower is better for all ROUGE scores. 8
Table 5. Exact match accuracies for various baselines and our CSL-based method, using COGS as an intermediate fine-tuning task. Joint finetuning splits are formatted (synthetic split + naturalistic split). 10
Table 6. Exact match accuracies for various baselines and our CSL-based method using GeoQuery as an intermediate fine-tuning task. Joint finetuning splits are formatted (synthetic split + naturalistic split). 10

1.0 Introduction

Compositional (sometimes called systematic) generalization is the ability to work with and understand new combinations of atoms, rules, or structures one has previously encountered. Such generalization is of great interest to machine learning practitioners because successful compositional generalization implies a number of other desirable properties: robustness to spurious variations of an input data distribution, rule-based behavior that can be used to identify a model's understanding or gaps therein, and so on. Deep neural networks are a family of methods which have dramatically shifted the research landscape over the past decade. Despite the massive gains across many data modes and an even greater number of tasks, there is a mounting body of evidence that these models struggle to reason in compositional, systematic, and robust ways. This is acceptable for some problem spaces, namely those where lack of model explainability and failures due to spurious correlations bear minimal cost or ethical risk, but this limitation becomes more and more grating as deep learning becomes involved in an ever-greater roster of domains and problem spaces. Compositional generalization and the implied abilities of a model able to do it are then no longer desirable perks or possible upsides, but a practical goal or benchmark for a growing body of practitioners. We refer readers to (Lake and Baroni 2018) for an introduction to compositional generalization in general and its relevance to deep neural networks in particular. Given the degree of thought and effort that has gone into the study of compositionality in the realm of natural language, so too has most of the deep learning community's research into the issue been motivated by natural language models and tasks. We follow suit to leverage these earlier insights and for the sake of continuity with prior work.

This work aims to answer the following questions:

1. Is there evidence of multiple *kinds* of compositional generalization? If so, how does model performance vary across them?

2. Much prior work has focused on synthetic datasets or on semantic parsing in particular as benchmarks for compositional generalization – are models and approaches designed for these tasks applicable to other, more naturalistic language tasks of interest?

3. What avenues of intervention do we have for increasing the compositional generalization abilities of deep language models, ideally without requiring architectural or pretraining modifications that would necessitate throwing away existing, large-scale pretrained models?

2.0 Local and Non-Local Compositionality

2.1. Initial Findings – COGS

We use the "**Co**mpositional **G**eneralization Challenge based on **S**emantic Interpretation", or COGS, dataset (Kim and Linzen 2020) as an initial testbed for understanding the compositional generalization behavior exhibited by existing transformer models of varying scales and training regimes. COGS is a synthetic, semantic parsing task which asks a model to take a natural language input sentence derived from an English Probabilistic Context-Free Grammar, or PCFG (Collins 2011), then return a parse of its semantic relationships. A simple example would be the input sentence "The cobra froze" and its corresponding parse "cobra(x_1) AND freeze.theme(x_2,x_1)". We see that the model must bind all entities and relations from the input sentence to variables of the form x_i; the model must use these variables in an internally consistent fashion to produce a correct parse. COGS is a useful benchmark in our case because it provides a suite of test splits built specifically to test for compositional generalization with respect to its training split. For example, one split tests whether a model can parse nouns used only as subjects in training when they are used as objects in the generalization split. Another tests whether a model can successfully parse sentences built with recursive prepositional phrases ("I grabbed my pen on the notebook on my table in Tim's house within…"). This diversity of train-test splits gives us an opportunity to see whether different types of transformer language models exhibit different strengths or weaknesses across various compositionally demanding problems, and whether patterns emerge among different *types* of generalization splits.

We present the most informative results from two of the transformers trained and evaluated on COGS: the first is the small, 4-layer, 4-attention head transformer used in (Kim and Linzen 2020), hereafter referred to as the baseline model, while the second is a pretrained T5-Large model introduced in (Raffel et al. 2020). The baseline model was trained on COGS from a random initialization and has roughly 9.5 million parameters. In contrast, the T5-Large model was pretrained on a variety of supervised and unsupervised sequence-to-sequence language tasks and has approximately 770 million parameters. We use previously reported results for the baseline model and adhered to the training procedures outlined in (Kim and Linzen 2020) when finetuning the T5-Large model. Note that we are not inherently interested in a fair comparison – we are more interested in understanding whether larger, pretrained models used throughout much of the literature exhibit qualitatively different behavior from the small models typically trained from scratch in synthetic benchmark papers. We trained a finer gradient of model sizes (T5-Small, T5-Medium, and so on) both pretrained and randomly initialized, but omit them for brevity because their behaviors adhered strongly to the trends most clearly represented by these two models at opposite ends of our model suite.

Turning our attention to Table 1, we see that the T5-Large model exhibits dramatically improved generalization performance on a variety of COGS splits, namely those based on more "local" compositionality – shifting of noun parts-of-speech. This is not terribly surprising, given that T5 pretraining could have given the model access to language that violated the strict gaps between the COGS train and generalization splits (i.e., seeing a common proper noun used as both subject *and* object in the pretraining corpus). We see this as a useful insight; our primary interest is in a model's ability to exhibit compositional generalization on practical tasks, not whether a model can do so without any of the benefits of pretraining or scale modern deep learning regimes provide us. It is reassuring that large models can handle these relatively simple forms of compositional generalization, though their utility as benchmarks may be limited

if they can be easily sidestepped by increased scale and standard pretraining. A more useful generalization split would be one that continues to fail *despite* these common measures.

Fortunately, Table 2 shows us that there are indeed such splits within COGS: those focused on the shifting grammatical role of prepositional phrases rather than simple nouns, as well as on recursion of either sentential complements or prepositional phrases. In all cases the T5-Large model sees negligible or even zero improvement over the far smaller, randomly initialized baseline. Compared to the "local" compositionality splits from Table 1, these are oriented around more "global" reasoning: understanding the independence of recursing phrases and being able to reason about phrasal semantics in a unified, cohesive manner.

We are left with multiple, useful insights. Increasing a transformer's scale and the scope of its pretraining allows us to sidestep many issues of "local" compositionality, presumably so long as some critical mass of variations appear within the pretraining corpus. Given the sheer scale of modern pretraining text corpora, this diversity assumption seems likely to hold. However, there is no such benefit for compositionality based on more phrasal or structural components; further work is needed to pinpoint and resolve the bottlenecks around a model's ability (or incentive) to learn more structural rules.

Table 1. Local compositional generalization for randomly initialized and finetuned transformers on COGS generalization splits. The best performing model results are in bold.

Local Tasks	Noun Type	Baseline Transformer (Kim and Linzen 2020)	T5-Large (Raffel et al. 2020)
Subject → Object	Common	31%	**99.3%**
	Proper	30%	**73.5%**
Object → Subject	Common	87%	**99.6%**
	Proper	45%	**99.6%**
Primitive → Subject	Common	17%	**99.3%**
	Proper	0%	**99.9%**

Table 2. Global compositional generalization for randomly initialized and finetuned transformers on COGS generalization splits. The best performing model results are in bold.

Global Tasks	Baseline Transformer (Kim and Linden 2020)	T5-Large (Raffel et al. 2020)
Sentential Complement Recursion	**0%**	**0%**
Preposition Recursion	0%	**10.1%**
Object Preposition → Subject Preposition	0%	**0%**

2.2. Recursive Decoding – gSCAN

Parallel to our preliminary work with COGS we also explored the Grounded SCAN (or gSCAN) dataset (Ruis et al. 2020), itself inspired by the earlier SCAN dataset (Lake and Baroni 2018). gSCAN is a grounded navigation task wherein a model receives as input a simple grid world filled with representations of the model's current position and a variety of other objects, as well as a natural language instruction the model is meant to carry out. The model must cross-reference its instruction with the grid world representation and produce, in one shot, a navigation plan that satisfies the instruction's request given the initial grid world state. The grid world entities, aside from the model agent, consist of single-cell shapes of various colors, sizes, and other properties: "big red square", "little blue circle", or "heavy yellow circle" are all examples of such entities. Examples of natural language instructions include "Push the small square", "Walk to the circle while spinning", or "Push the red small circle *cautiously*". The model's navigation plan is simply a string of movement commands in one of the four cardinal directions. All instructions map to a single, golden navigation plan: adverbs such as "while spinning" or "cautiously" all map to discrete action sequences. Like COGS, gSCAN has a variety of generalization splits meant to test different types of compositionality: understanding a new type of primitive (red or yellow squares) built as an intersection of types encountered during training, navigating in a direction (southwest) that never came up during training, extrapolating to scenarios that require longer navigation plans than those produced during training, and so on. We refer readers to (Lake and Baroni 2018) for full details on the gSCAN dataset and to (Setzler, Howland and Phillips 2022) for more details on the methodology and analyses outlined below.

We begin by comparing the LSTM model introduced by (Ruis et al. 2020) with the previously state-of-the-art messaging passing model from (Kuo et al. 2020) and a small transformer of similar size to the baseline of our COGS experiments. We see in Figure 2 that the message-passing and transformer architectures reliably outperform the LSTM baseline across most splits, though neither one of the message-passer or the transformer are better than the latter in all cases. Both of these higher-performing architectures do quite well on more "local" compositional splits: red squares, yellow squares, and class inference, though as with COGS both perform poorly on most splits built around more "global" compositionality.

To address this persistent problem with more globally-focused splits, we identified two key conceptual bottlenecks: the lack of diversity in the model's initial position and orientation, as well as the need for the model to generate its entire navigation plan in a single shot. By default, the model *always* begins facing east during training; to eliminate the possibility that a model performs poorly due to overfitting surface-level statistics early on in its navigation, we experimented with randomizing the agent's initial position and revising its target navigation plans accordingly. We refer to this as a Randomized condition.

To address the bottleneck of one-shot planning, we relax this requirement and instead allow the baseline LSTM model to "recursively decode" its navigation plan by taking a single step at a time, updating the world state, then querying the model with this new world state and the same language instruction as before. This makes some aspects of the problem easier – it bears repeating that we are more interested in understanding where challenges with compositional reasoning arise than we are on making strict one-to-one comparisons with prior art. However, we control for the increase in training sample diversity that this recursive approach introduces by training the baseline LSTM with an "intermediate state" version of the dataset. In this setting, the training set is expanded to include all intermediate world states produced by the ground truth navigation paths used to train our recursive decoding model. We saw no significant

differences between a non-recursive decoding model trained on the original split versus this intermediate state split.

We compare our recursive decoding LSTM, with and without a randomized initial orientation, to the baseline LSTM and message-passing models in Table 3. Recursive decoding leads to huge performance increases on the length extrapolation task, though only the combination of a randomized initial orientation and recursive decoding are enough to increase performance on the novel direction (southwest) split. We observe empirically that the randomized recursive decoding model navigates to the correct destination cell significantly more often than these results indicate. If we include “non-canonical solutions” which arrive at the correct destination, typically differing from the ground truth navigation plan by only one or two commands, this approach is successful 86.3% ($\pm$9.0) of the time – over 80 points of accuracy better than the prior state of the art.

The results on the novel direction split are particularly significant. While the length extrapolation task is made trivially more straightforward when adopting recursive decoding, novel direction has no analogous, intuitive benefit from breaking planning down into individual steps. Indeed, the control experiment using our intermediate state split shows that the increase in directional compositionality could *only* have come from treating every training example as a decomposed, but still cohesive, sequence. The baseline model needs to perform all state tracking and other forms of “bookkeeping” using the same machinery and state it needs to actually put together the rest of its navigation plan. Recursive decoding offloads all of this state tracking into the world state itself. This is a useful insight even for purely language tasks: introducing an explicit separation, or division of labor, between state tracking and problem-specific reasoning could free up language models to reason more compositionally at the phrasal or structural level.

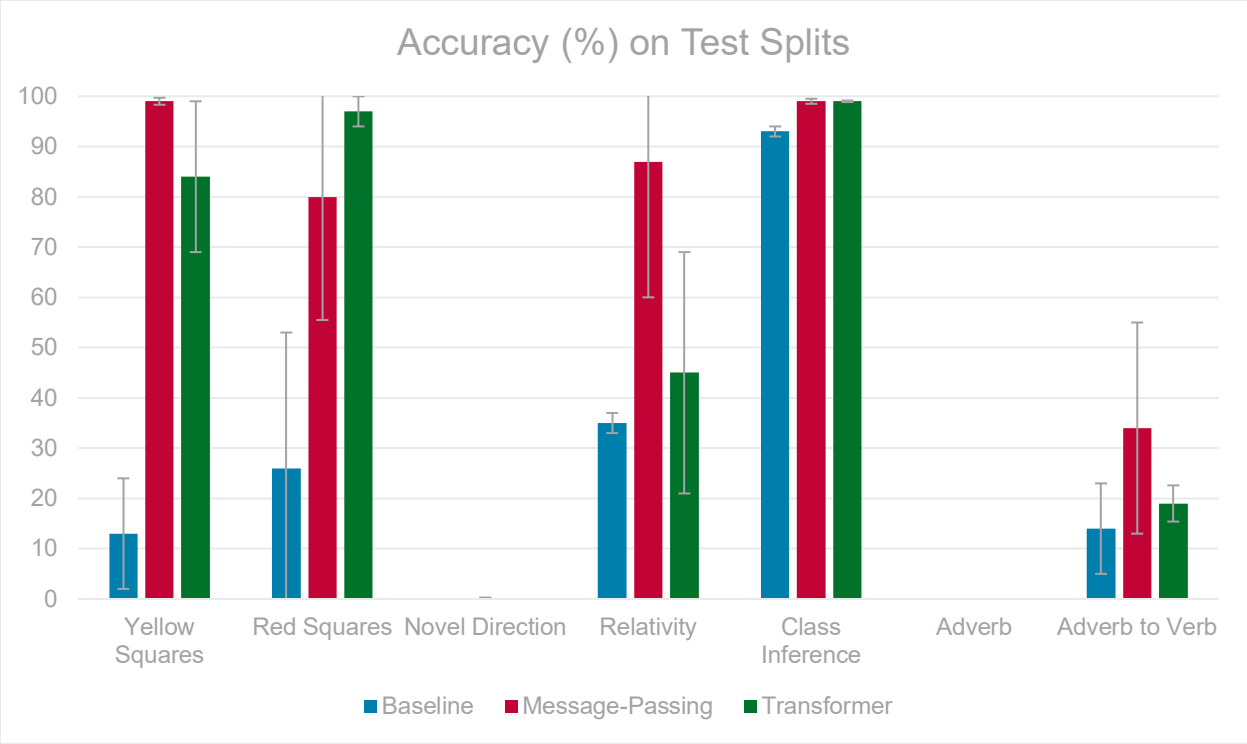

Figure 1. GSCAN generalization splits with exact match accuracies for prior art and our transformer model. The 'Yellow Squares', 'Red Squares', and 'Class Inference' tasks require local compositional generalization. All other tasks demand global compositionality.

Table 3. Exact match accuracies for two key GSCAN splits requiring global compositional generalization. Best performing model results are in bold.

Generalization Split	Baseline (Ruis et al. 2020)	Prior SotA (Kuo et al. 2020)	Recursive Decoding (Ours)	Recursive Decoding (Randomized) (Ours)
Novel Direction	0.0% (±0.0)	5.7% (±0.0)	3.11% (±0.9)	**43.6% (±6.1)**
Length	2.1% (±0.1)	2.1% (±0.1)	**84.4% (±3.2)**	79.3% (±0.4)

3.0 Data Interventions for Boosting Compositionality

3.1. Input Sequence Compression

Our experiments with both COGS and gSCAN have illuminated a key limitation of existing large language models: they can reason compositionally in narrow terms, such as adapting part-of-speech rules for nouns or interpolating class attributes for gSCAN entities, but consistently struggle to achieve similar behavior when reasoning at the phrasal or structural level. We hypothesize that this may be the case because task inputs are themselves strictly fine-grained and non-phrasal; models must build up phrasal representations through progressively deeper layers, with no additional memory or computational machinery to account for hierarchical representations of their input. Inspired by the use of feature pyramids in computer vision domains (Dollár et al. 2014), we augment COGS input sequences with a compressed (average pooled) version of the original sequence. Since transformer attention is capable of learning global-scale relations between tokens, we focus on learning spatially self-contained compressed tokens and produce the augmented input by applying a sliding window of length three to the input sequence. The compressed sequence is then concatenated to the original input, with the rest of the model architecture unchanged. The experimental methodology is otherwise identical to our baseline COGS experiments.

Table 4 presents our results, comparing T5-Tiny (a T5 built to be of similar size to the baseline presented in (Kim and Linden 2020)) trained from a random initialization with and without our augmented compression inputs. We also include results for finetuned T5-Small, the smallest pretrained model available from (Raffel et al. 2020), to account for any interactions between pretrained models and supplementary compression. We report various Rouge scores (Lin 2004) in lieu of exact match accuracy due to a collapse of model performance under the compression regime. A particularly interesting result is that we actually see a decrease in both train and generalization losses for models trained with compression despite a dramatic decrease in all Rouge scores. This is not explained by the simple inclusion of additional input tokens since the loss score is based entirely on the decoded output sequence. This low-ROUGE, low-loss behavior would be consistent with compression-augmented models predicting output distributions that are *on average* closer to ground truth outputs, but which rarely have the correct output token as the single most likely prediction. In short, consistent with models which are consistently “close, yet no cigar”.

While we were unable to perform a deeper quantitative analysis of compression-augmented model behavior, the overall decrease in generalization loss is significant and we believe the compression approach, or an approach like it, is worth examining more carefully in future. It could be the case that transformers, whose attention mechanism is essentially a form of content-based matching, could be thrown off by learning to process input sequences directly side-by-side with what are essentially blurry downsamples of themselves; if these sequences are not well distinguished from each other, models need to learn such disentanglement themselves in order to effectively leverage the compressed inputs. This might be addressed by adding a “compression embedding” to our compressed inputs, indicating the degree to which they were pooled from the original sequence. Such an embedding also opens the door to multiple layers of compression for a given example: so long as the content for different compression levels is distinct in feature space, the transformer should have little difficulty interpreting what level of granularity an input token has. Our straightforward approach also has the compressed and original inputs densely interacting with each other at every step –

constraining the degree or frequency with which these different representations impact each other could alleviate confusion around repeated, blurred content in the sequence.

Table 4. ROUGE scores on the COGS dataset for transformers with and without supplementary input compression. Best performing model results are in bold. Lower is better for all ROUGE scores.

Architecture	Compression	Train Loss	Gen. Loss (average)	Rouge1 (average)	Rouge2 (average)	RougeL (average)
T5-Tiny	No	**1.302**	2.275	**2.231**	**0.085**	**2.199**
T5-Tiny	Yes	**1.28**	**1.723**	0.584	**0.091**	0.579
T5-Small (Fine-tuned)	Yes	**1.268**	**1.731**	0.0	0.0	0.0

3.2. Data Augmentation via Induced Grammars

Architectural improvements are not the only way we can try to train models to improve compositional generalization ability; we can also examine techniques to build training datasets that demand, or at least encourage from an optimization perspective, compositionality. The Compositional Semantic Learner, or CSL (Qiu et al. 2021) is one such approach. The authors induce a quasi-synchronous context-free grammar (Smith and Eisner 2006) over a synthetic training split, in their case various semantic parsing tasks, then sample a number of additional training examples generated from the induced grammar to augment the original training split. This approach encourages compositionality in a number of ways: the grammar itself is regularized to learn small, highly reusable rules which are by definition composable, sampling from the grammar allows practitioners to bias the augmented examples to have increased length or other statistically rare properties, and the increased volume of examples derived from composable rules encourages models to memorize those rules and their exceptions rather than memorizing examples non-compositionally. This last observation is motivated by empirical evidence of the tolerance principle (Yang et al. 2017) (Schuler et al. 2021) in the linguistics literature. For additional background and implementation details for CSL and its underlying grammar, we refer interested readers to the relevant referenced works.

While the success of CSL is impressive, we ask an important follow-up question: are the gains of CSL transferable to non-synthetic, naturalistic language tasks? For a method to be broadly applicable we would like it to improve performance across a variety of tasks that should benefit from more compositional models, without requiring those tasks to be strictly synthetic or built around semantic parsing. A straightforward solution to this problem is challenging – if we could reliably fit such a grammar to a diverse array of naturalistic tasks, we would have already solved many core issues of language modeling! To work around this constraint, we ask a narrower question: are the gains of CSL-driven finetuning transferable to downstream, *later-stage* finetuning on naturalistic tasks of interest?

All networks finetuned for our CSL experiments are T5-Base models from (Raffel et al. 2020). We use the COGS (Kim and Linzen 2020) and GeoQuery (Zelle et al. 1996) semantic parsing benchmarks as our synthetic datasets and the CNN-Daily Mail (summarization) (See et al. 2017), PAWS (paraphrase identification) (Zhang et al. 2019), and SquAD (question answering) (Rajpurkar et al. 2016) as naturalistic, downstream datasets that should benefit from

improved compositional generalization ability. We use the standard train/test splits for all datasets except for GeoQuery, whose seven ‘length’, ‘template_[1,2,3]’, and ‘tmcd_[1,2,3]’ splits we use in lieu of a traditional training split. Keeping with prior literature using GeoQuery, we report ‘template’ and ‘tmcd’ results by averaging the results of their three constituent sub-splits.

As a simple baseline, we initially finetuned T5-Base models on each naturalistic dataset for 20,000 steps. We also trained separate two-stage, ‘hybrid’ baselines by first finetuning a T5-Base instance on a given synthetic dataset for 10,000 steps, then finetuning for an additional 10,000 steps on a given naturalistic dataset. There were twenty-one such hybrid baselines, one for each synthetic-naturalistic dataset combination.

Our contribution comes from extending the hybrid condition above, by augmenting the synthetic dataset with examples derived from a corresponding CSL grammar. In all cases we sampled 100,000 additional examples from a CSL model fit to the synthetic dataset and combined them with the original synthetic training samples, upsampling the latter set as needed to achieve balance between CSL and non-CSL examples. Aside from the augmentation of the synthetic training set, training methodology is unchanged from the hybrid baseline condition.

Tables 5 and 6 depict results for all three of the above conditions for models using COGS and GeoQuery as their synthetic datasets, respectively. While average exact match performance accuracy varies across model conditions, with the single-stage and CSL two-stage models performing better than the non-CSL two-stage setting, large standard deviations make it difficult to draw strong conclusions. The dip in performance for the two-stage, non-CSL models could simply be due to overfitting on the (non-upsampled) synthetic training set.

Prior work, including (Qiu et al. 2022), indicates that models trained using CSL-augmented data are significantly more capable of compositional generalization, though our results indicate that this compositionality might only be directly applicable to the CSL-augmented task itself. This leaves us in a quandary. Applying a grammar-fitting method like CSL to the naturalistic tasks themselves would be the most desirable course of action, though as we discussed earlier this remains out of reach. It might be possible to use a more flexible grammar formulation to *approximately* fit a CSL-like model on such naturalistic tasks, but the algorithmic enhancements needed would be extensive and are beyond the scope of our work here.

Another avenue for future work would be finding ways to more firmly “bake in” the compositionality which models demonstrably learn from CSL-augmented tasks. If our results are due to some form of catastrophic forgetting (French 1999) wherein the model unlearns the knowledge gained from the first fine-tuning step, jointly fine-tuning over both tasks might be a sufficient remedy. The model might also have sufficient capacity to simply memorize compositional rules which apply only to the CSL-augmented task rather than learning rules applicable to language more broadly. If this is the case then, sticking within the realm of data-based interventions, it may be necessary to integrate data augmentation or example mining to encourage compositional reasoning at the level of model pre-training itself, where the model develops its underlying behaviors independent of any particular downstream task.

Table 5. Exact match accuracies for various baselines and our CSL-based method, using COGS as an intermediate fine-tuning task. Joint finetuning splits are formatted (synthetic split + naturalistic split).

Finetuning Scheme	Finetuning Split(s)	PAWS (Paraphrase Identification)	SQUAD (Question Answering)	CNN-DailyMail (Summarization)
Single-stage (Baseline)	N/A	93.5 (± 24.7)	67.7 (± 46.8)	0.02 (± 1.6)
Two-stage (Baseline)	Train	91.5 (± 27.9)	65.8 (± 47.5)	0.00 (± 0.93)
Two-stage (CSL)	Train	93.5 (± 24.7)	67.3 (± 46.9)	0.02 (± 1.6)

Table 6. Exact match accuracies for various baselines and our CSL-based method using GeoQuery as an intermediate fine-tuning task. Joint finetuning splits are formatted (synthetic split + naturalistic split).

Finetuning Scheme	Finetuning Split(s)	PAWS (Paraphrase Identification)	SQUAD (Question Answering)	CNN-DailyMail (Summarization)
Single-stage (Baseline)	N/A	93.5 (± 24.7)	67.7 (± 46.8)	0.02 (± 1.6)
Two-stage (Baseline)	Length	93.4 (± 24.9)	66.1 (± 47.3)	0.0 (± 1.9)
	Template	93.3 (± 25.0)	66.3 (± 47.3)	0.0 (± 1.4)
	TMCD	93.4 (± 24.9)	66.1 (± 47.3)	0.0 (± 1.2)
Two-stage (CSL)	Length	93.6 (± 24.5)	66.0 (± 47.4)	0.0 (± 1.6)
	Template	93.6 (± 24.5)	66.2 (± 47.3)	0.0 (± 1.6)
	TMCD	93.4 (± 24.8)	65.8 (± 47.4)	0.0 (± 1.3)

4.0 References

Chang, Yingshan, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2020."Webqa: Multihop and multimodal qa." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16495-16504.

Collins, Michael. “Probabilistic Context-Free Grammars (PCFGS).” Columbia university course notes, 2011. url: http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/pcfgs.pdf

Dollár, Piotr, Ron Appel, Serge Belongie and Pietro Perona, "Fast Feature Pyramids for Object Detection." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 8, pp. 1532-1545, 2014. doi: 10.1109/TPAMI.2014.2300479.

French, Robert M. "Catastrophic forgetting in connectionist networks." Trends in cognitive sciences 3, no. 4 , pp. 128-135. 1999.

Gao, Tong, Qi Huang, and Raymond J. Mooney. 2020. "Systematic generalization on gscan with language conditioned embedding." arXiv preprint.

Kim, Najoung and Tal Lizen. 2020. “COGS: A Compositional Generalization Challenge Based on Semantic Interpretation.” Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): 9087-9105.

Lake, Brenden, and Marco Baroni. "Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks." In International conference on machine learning, pp. 2873-2882. PMLR, 2018.

Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In Text summarization branches out, pp. 74-81. 2004.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. "Exploring the limits of transfer learning with a unified text-to-text transformer." J. Mach. Learn. Res. 21, no. 140: 1-67.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250. 2016.

Ruis, Laura, and Brenden Lake. 2020. "Improving Systematic Generalization Through Modularity and Augmentation." arXiv preprint.

Schuler, Kathryn, Charles Yang, and Elissa Newport. "Testing the Tolerance Principle: Children form productive rules when it is more computationally efficient." 2021.

See, Abigail, Peter J. Liu, and Christopher D. Manning. "Get to the point: Summarization with pointer-generator networks." arXiv preprint arXiv:1704.04368. 2017.

Smith, David A., and Jason Eisner. "Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies." In Proceedings on the Workshop on Statistical Machine Translation, pp. 23-30. 2006.

Yang, Charles, and Silvina Montrul. "Learning datives: The Tolerance Principle in monolingual and bilingual acquisition." Second Language Research 33, no. 1, pp. 119-144. 2017

Zelle, John M., and Raymond J. Mooney. "Learning to parse database queries using inductive logic programming." In Proceedings of the national conference on artificial intelligence, pp. 1050-1055. 1996.

Zhang, Yuan, Jason Baldridge, and Luheng He. "PAWS: Paraphrase adversaries from word scrambling." arXiv preprint arXiv:1904.01130. 2019.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov