Compositional Networks Enable Systematic Generalization for Grounded Language Understanding

Yen-Ling Kuo, Boris Katz, Andrei Barbu
MIT CSAIL & CBMM
{ylkuo, boris, abarbu}@mit.edu

Abstract

Humans are remarkably flexible when understanding new sentences that include combinations of concepts they have never encountered before. Recent work has shown that while deep networks can mimic some human language abilities when presented with novel sentences, systematic variation uncovers the limitations in the language-understanding abilities of networks. We demonstrate that these limitations can be overcome by addressing the generalization challenges in the gSCAN dataset, which explicitly measures how well an agent is able to interpret novel linguistic commands grounded in vision, e.g., novel pairings of adjectives and nouns. The key principle we employ is compositionality: that the compositional structure of networks should reflect the compositional structure of the problem domain they address, while allowing other parameters to be learned end-to-end. We build a general-purpose mechanism that enables agents to generalize their language understanding to compositional domains. Crucially, our network has the same state-of-the-art performance as prior work while generalizing its knowledge when prior work does not. Our network also provides a level of interpretability that enables users to inspect what each part of networks learns. Robust grounded language understanding without dramatic failures and without corner cases is critical to building safe and fair robots; we demonstrate the significant role that compositionality can play in achieving that goal.

Introduction

One of the defining characteristics of human languages is that they are productive. We can combine together concepts in novel ways to express ideas that have never been thought of before. This is for a good reason: as children, we observe very little of our world before we must speak to others, meaning that even mundane language is novel and not just parroting back something already expressed for us. Similarly, even with massive data collection efforts, deep models can only have an opportunity to observe a small subset of the possible utterances and worlds. This problem becomes especially acute when those models must drive the behavior of a robot, because misunderstanding a command may pose a serious safety hazard.

Recently, there have been a number of attempts to probe the understanding of deep networks trained to perform linguistic tasks. Lake and Baroni (2018) point out that generalization to novel compositions of concepts is rather limited. This is not a matter of the amount of data available; for example, McCoy et al. (2019) find that even networks with the same test set performance can have very different generalization abilities. More recently, Ruis et al. (2020) released gSCAN for testing the generalization abilities of grounded language understanding. In gSCAN, an agent must follow a natural-language command in a 2D environment. Commands of specific types are systematically held out; for example, no command with a particular adjective-noun combination appears in the training set. When the test set distribution is similar to the training set, performance is phenomenal: 97% of commands are executed correctly. Yet, when combinations are missing from the training set, such as holding out an adjective-noun pair like "yellow squares", only 24% to 55% of commands are executed correctly.

Guided by the notion that compositionality is the central feature of human languages which deep networks are failing to internalize, we construct a compositional network to guide the behavior of agents. Given a command, a command-specific network is assembled from previously-trained modules. Modules are automatically discovered in the training set without any annotation. The network structure that combines those modules is derived from the linguistic structure of the command. In

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 216–226
November 7–11, 2021. ©2021 Association for Computational Linguistics

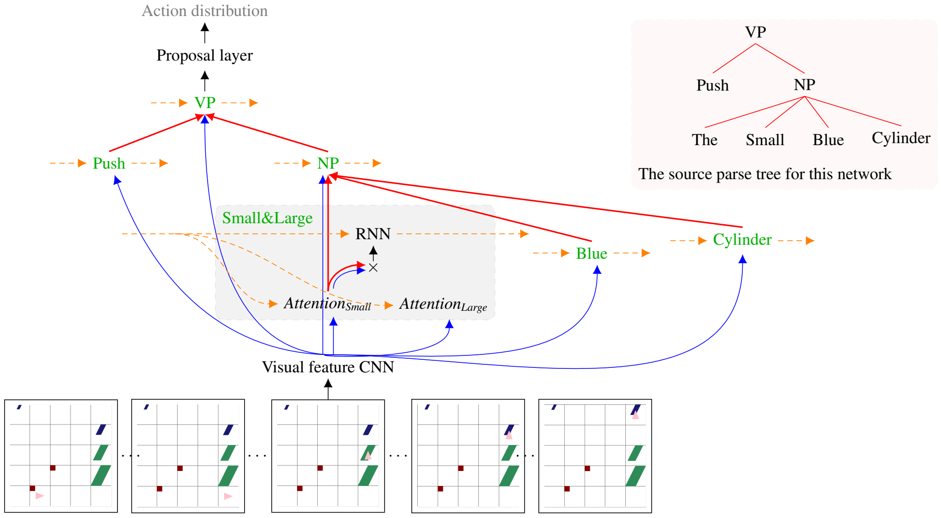


Figure 1: The structure of the model interpreting and following *Push the small blue cylinder*. In light red at the top right, we show the parse tree, as produced by a constituency parser. This tree is the source of the structure found within the compositional network; note the corresponding structure of the red lines. Each token in the parse becomes a recurrent network in the model, shown in green. Red lines show which recurrent networks are connected to one another through attention maps. Blue lines are visual observations, available to every node. Orange lines are recurrent connections allowing words to keep state. One module, Small&Large, is expanded, shown on a grey background. This module has two components which are trained to have opposite polarity. Each predicts an attention map which then updates the hidden state of the word and is passed to any subsequent words. The state of the root model is decoded into an action that the agent should execute next.

this way, the compositional structure of language is reflected in the compositional structure of the computations executed by the network.

Compositionality is not specific to any one dataset – it is a general principle – and the implementation we provide here is not specific to gSCAN. Even though our base network achieves the same 97% performance in the random test set as the state-of-the-art models for gSCAN, it generalizes significantly better in a number of ways, including few-shot learning and longer action sequences. Where this approach shines is predicted well by the types of compositionality that exist in the network. For example, novel combinations of concepts related to individual objects perform well. An additional benefit of compositional networks is that they open the door to naturally including other linguistic principles. For example, it appears that not all parses are made equal. In our case, network structures derived from a semantic parser lead to better-performing agents compared to structures derived from a constituency or dependency parser. We show another example of this idea by incorporating the lexical semantics of words, e.g., antonyms, as an additional loss while training the network.

Our approach forgoes the most popular mechanism for increasing the generalization performance of neural networks: data augmentation. Data augmentation has substantial drawbacks: it is arbitrary, it slows down training time, and it is dataset and problem specific. In addition, data augmentation introduces many parameters that must be tuned and much knowledge that must be provided by humans. We show that the generic principle of compositionality can replace data augmentation without any of these drawbacks. It remains an open question whether every data augmentation approach has a corresponding compositional structure that can supplant and generalize it. Compositional approaches could be combined with data augmentation, potentially raising their performance even further.

Our work makes four contributions.

1. We demonstrate a class of compositional networks which generalize the ability of agents to execute commands that contain novel combinations of concepts.
2. We systematically replace data augmentation with compositionality resulting in both higher performance and a simpler, principled, and dataset-agnostic method.
3. We incorporate the lexical semantics of words (e.g., if they are antonyms or synonyms of each other) into compositional networks.
4. Our method addresses generalization tasks in gSCAN which no prior work does, such as learning from a few examples and generalizing to longer sequences.

Related Work

Command following Robots must ground in their surroundings. Previous work grounds concepts such as objects (Guadarrama et al., 2014), spatial relations, and object properties (Kollar et al., 2010). To turn a command into actions, Chen and Mooney (2011) and Matuszek et al. (2013) learn semantic parsers that convert instructions into plans. Mei et al. (2016) demonstrate a seq2seq network fused with a visual encoder to predict action sequences from input sentences. This type of seq2seq network is adopted by many supervised models and reinforcement learning agents (Fu et al., 2019; Shah et al., 2018). Blukis et al. (2018) present a U-Net architecture that predicts goal distributions conditioned on linguistic commands to control a drone. Predicting a single final goal may not always be ideal as language can describe the manner of interacting with objects & the world. Kuo et al. (2020) demonstrate that a compositional network structured according to the parse of the input command can combine with a sampling-based motion planner to guide the sampling process. Similar to Kuo *et al.*, we use RNNs as the base units of the model and compose networks from parses. Our approach is further compatible with any type of parsers and can encode lexical semantics of words, which allows us to investigate how compositional architectures generalize systematically.

Generalization in grounded language understanding Many methods have been proposed to test an agent's generalization capabilities in different perspectives of grounded language understanding. Yu et al. (2018) consider a multi-task setting and train an agent to navigate a 2D maze and to answer grounded questions. Pezzelle and Fernández (2019) focus on evaluating agents' abilities in assessing the meaning of adjectives in context. Chaplot et al. (2018) and Hermann et al. (2017) evaluate RL agents' capability to generalize to novel composition of shape, size, and color in 3D simulators. The BabyAI platform (Chevalier-Boisvert et al., 2018) evaluates RL agents in a grid world with tasks that demand an increasing understanding of the compositional structure of their domain. They show that RL agents generalize poorly when the tasks have a compositional structure. Bogin et al. (2021) learn latent trees to ground compositional reasoning in the visual question answering domain. Rather than focusing on one aspect of generalization as much of the prior work does, gSCAN (Ruis et al., 2020) takes ideas from meaning composition to create a systematic battery of tests for generalizing in grounded settings. A few recent approaches attempted to address the generalization challenges in gSCAN. Heinze-Deml and Bouchacourt (2020) add an auxiliary loss in the baseline seq2seq model to predict the location of the target object. However, it only improves in a few subsets related to target object predictions. Gao et al. (2020) use a language conditioned graph network to model the relation between the objects and natural-language context. While the graph network improves some subsets of novel compositions, they did not evaluate on few-shot learning and generalization to longer action sequences.

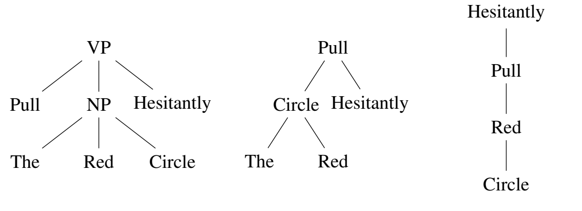
Compositional networks The idea that linguistic structures and compositionality can be reflected in the internal workings of a model to enable better generalization is not itself new (Liang and Potts, 2015). Tellex et al. (2011) and Barbu et al. (2012) mirror the linguistic structures produced by a constituency parser in the structure of a graphical model to respectively execute robotic commands and recognize actions. Similarly, Socher et al. (2011) and Legrand and Collobert (2014) build neural networks based on parse trees. Andreas et al. (2016) demonstrate a procedure to compose a collection of network modules based on a semantic parser for visual question answering. Not all modular networks are derived from language; for example, prior work has modularized sub-policies and sub-goals in embodied question answering (Das et al., 2018) or transfer learning (Alet et al., 2018) according to other task-specific principles.

Technical Approach

We first describe how the compositional networks can be constructed from any linguistic parses. Then, we show how a linguistic notion, such as a known relationship between words, can be incorporated in the model.

Parsing natural-language commands

Given a natural-language command, a parser produces a hierarchical structure of that command revealing its part-based compositional structure, i.e., which words modify one another, and the nature of that modification. Different approaches to analyzing linguistic utterances lead to different structures.



(a) Constituency parse (b) Dependency parse (c) Semantic parse

Figure 2: Parses for the command "Pull the red circle hesitantly." in three formalisms. Each leads to different compositional networks which have radically different generalization abilities.

Here we consider three kinds of parsers: a constituency parser (Joshi et al., 2018), a dependency parser (Dozat and Manning, 2016), and a semantic parser; see Figure 2 for an example of the different structures produced. In what follows, we use the language of constituency parsing: that a parse is a collection of nodes arranged in a tree; dependency parses consist of words and binary relationships between words; while semantic parses in this work consist of a formula in propositional logic. This is purely for linguistic convenience, as no shared lexicon exists between these parsers. Our approach treats all parses as labeled directed acyclic graphs and is agnostic to the source of the parse. In the Results, we discuss the differences between these parsers.

Constructing compositional networks

Given the parse of a command, the nodes in the parse tree are replaced with RNNs connected to one another according to the structure of the parse. An example network structure is shown in Figure 1. This compositional network is used to predict actions for the agent to follow based on the visual observation at every time step.

Recurrent word modules We use RNNs as the basic building blocks of the compositional networks because the hidden states provide the capacity to maintain the context of task progression, for example, pushing heavy objects twice in order to move them. Each word or predicate/function in the semantic parse corresponds to a specific RNN, forming a lexicon of RNNs. In the case of dependency or constituency parses, we create a separate model for each word depending on the arity of that word in the parse tree. Most parses are trees as described above, rooted in one node, corresponding to one word, predicate, or operator in the parse. Some parses can consist of multitrees, one or more trees that can share nodes. In this case, we can synthesize a dummy root node. Note that this operation of inserting a dummy root has linguistic precedent; for example, dependencies are considered by some to have a phantom root (Ballesteros and Nivre, 2013). The word that is the root plays a special role: its hidden state is decoded by a linear layer that computes a distribution over the next action.

Connecting word modules The information that flows between nodes always follows the reverse direction of the arcs in the parses. In the cases of parse trees described above, the information flows from children to its parent, i.e., from leaves to the root. Labels on the arcs are used to keep consistent the input to nodes with more than one argument. For example, the word "grab" usually involves two arguments, the agent and the patient; the RNN for "grab" takes as input the output of the RNN that corresponds to the agent first and the one for the patient second, consistently. Words with multiple arguments use a linear layer to combine together the input embeddings; the arc labels determine the arbitrary but consistent order in which the multiple input vectors should be combined before this linear layer.

Attention mechanism in word modules Within each word module, the RNN maintains a state vector and this internal representation is always used to predict an attention map before being accessed by other word modules. At each time step t, the module for word w receives as input an embedding obs_t of the agent's surroundings, the attention maps from its children $att_t^{c_1} \cdots att_t^{c_n}$, and its own state vector h_{t-1}^w from the previous time step. The embedding obs_t is computed by a CNN which is co-trained with the rest of the network. The attention map for word w is computed as follows:

$$att_t^w = softmax(MLP(h_{t-1}^w, obs_t \odot att_t^{c_1}, \\ \cdots, obs_t \odot att_t^{c_n}))$$

The observation is weighted by the attention maps from children first and combined with the hidden state to predict where to attend, i.e., the meaning of a word is grounded in the map. Inside the *MLP*, the weighted observations and the hidden state are mapped to the same dimension before being combined together. The attention map is normalized with softmax and adds up to 1. The RNN then takes this attention map to update its hidden state:

$$o_t^w, h_t^w = RNN^w(obs_t \odot att_t^w, h_{t-1}^w)$$

Attention maps are the only mechanism by which nodes communicate with one another. We demonstrate in the Results that this is critical to performance. It provides a common representation for all words, which in a sense makes all words compatible with one another. Without this restriction, words might never develop the ability to understand one another's representations.

Training compositional networks We train the CNN to encode the observations, the RNNs and attention modules for each word jointly. At training time, the input consists of pairs of commands and corresponding trajectories. The parser is pretrained, and in the case of the constituency and dependency parsing, an off-the-shelf general-purpose English model is used. The command is parsed and a corresponding network is instantiated. The word modules that have not been discovered in previous commands are instantiated with random weights. No information is provided as to which which part of the trajectory and relationships between the trajectory and other objects, and what each word in the command might refer to. The parameters of the resulting compositional network are trained without knowing the mapping between words and meanings. This knowledge must be inferred during training, thereby disentangling the meanings of each word. During training, the agent is provided with the ground-truth action at each time step to compute the maximum log-likelihood loss of the distribution over the next action and update the network.

Incorporating lexical semantics

Humans bring to bear tremendous prior knowledge to any new learning problem. Dubey et al. (2018) show that depriving humans of that knowledge by, for example, making dangerous objects look safe and vice versa, significantly impairs the ability of humans to learn and generalize. To this end, we demonstrate how to naturally add weak constraints, automatically derived from WordNet (Fellbaum, 2012), about the meanings of words.

Given a token in a parse, we search WordNet for related synonyms and antonyms. When creating the lexicon of RNNs, we consider the transitive closure of synonyms and antonyms as a single RNN for that concept. The combined RNN, e.g., "Small&Large" RNN in Figure 1, has two attention map outputs, but only one of the two is used depending on which variant of the concept appeared in the input. Intuitively, the computations to determine the relative sizes of objects are closely related to one another, regardless of whether one is checking if an object is small or large; this approach shares those computations between synonyms and antonyms. Critically, at training time, we add an additional loss, that the attention maps of these two concepts should be inverse of one another. This is done by optimizing the negative Hausdorff distance, which for grayscale maps minimizes the total intensity in the product of the two attention maps. A simple negation of maps would be ineffective as it would force one concept to be true when the other is not, which is not what being an antonym means. Not all objects that are not small, are large; some are merely irrelevant or their size is indeterminate. But, relative to a single reference object, the same object cannot usually be large and small at the same time. Hence, during training time, we add an auxiliary loss by computing the negative Hausdorff distance of attention maps of antonyms. This loss is used to avoid both attention maps paying attention to the same regions without disturbing one another when one of the two concepts is irrelevant. In general, knowledge about relationship between words can be used to augment the network, perhaps as derived from word embeddings.

Experiments

We evaluate the compositional network on the gSCAN dataset (Ruis et al., 2020) which was designed to systematically test the generalization ability of grounded agents. Our vocabulary size and trajectory distributions are the same as in gSCAN. The observation space for the agent is a 6×6 grid and the agent can choose from six random actions: walk, turn left/right, push, pull, and stay.

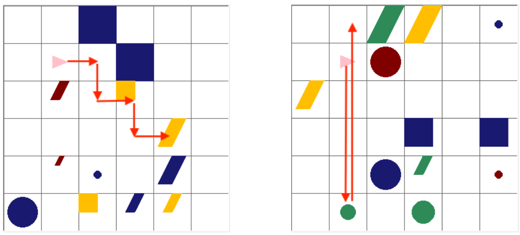
Figure 3 shows examples of two gSCAN commands in different environments. At test time, an agent receives a command and an environment (randomly placed objects with random sizes and colors). It predicts a sequence of actions to carry out that command. gSCAN includes adjectives that describe an object's color and size, nouns, verbs, prepositional phrases, and adverbs. We summarize the generalization conditions in gSCAN below.

A *Random*: all concepts and combinations appear in the training set to put other results in context.

B *Yellow squares* holds out types of references to an object, e.g., it is never referred to as "yellow square" but only as "small square".

	Seq2seq	GECA	AuxLoss	LCGN	State	+Attention	Constituency	Dependency	Semantic
					Ours	Ours	Ours	Ours	Ours
A	97.69 ± 0.22	87.60 ± 1.19	94.19 ± 0.71	**98.60** ± 0.95	49.83 ± 5.05	96.06 ± 1.40	96.20 ± 1.68	96.91 ± 1.86	96.73 ± 0.58
B	54.96 ± 39.39	34.92 ± 39.30	86.45 ± 6.28	**99.08** ± 0.69	4.37 ± 2.90	79.36 ± 32.71	80.82 ± 7.34	58.42 ± 18.31	94.91 ± 1.30
C	23.51 ± 21.82	78.77 ± 6.63	81.07 ± 10.12	**80.31** ± 24.51	5.53 ± 1.75	43.93 ± 15.42	40.33 ± 7.63	64.23 ± 6.04	67.72 ± 10.83
D	0.00 ± 0.00	0.00 ± 0.00	-	0.16 ± 0.12	1.62 ± 0.79	3.41 ± 1.21	3.66 ± 2.93	5.29 ± 3.36	**11.52** ± 8.18
E	35.02 ± 2.35	33.19 ± 3.69	43.43 ± 7.0	**87.32** ± 27.38	27.18 ± 3.75	68.84 ± 34.72	52.96 ± 15.19	28.34 ± 16.13	76.83 ± 2.32
F	92.52 ± 6.75	85.99 ± 0.85	-	**99.33** ± 0.46	43.29 ± 1.90	90.09 ± 14.81	97.25 ± 0.17	96.99 ± 1.79	98.67 ± 0.05
G k=1	0.00 ± 0.00	0.00 ± 0.00	-	-	3.38 ± 3.66	**1.79** ± 0.69	-	-	1.14 ± 0.30
k=5	0.47 ± 0.14	-	-	-	4.87 ± 1.22	6.31 ± 5.66	-	-	**8.85** ± 1.87
k=10	2.04 ± 0.95	-	-	-	8.48 ± 4.72	34.28 ± 6.59	-	-	**36.91** ± 5.13
k=50	4.63 ± 2.08	-	-	-	13.19 ± 2.53	45.79 ± 13.53	-	-	**46.30** ± 11.69
H	22.70 ± 4.59	11.83 ± 0.31	-	**33.60** ± 20.81	9.80 ± 0.74	13.27 ± 8.75	20.84 ± 1.87	0.00 ± 0.00	20.98 ± 1.38
I	See table 2; only the original publication and this work address generalization condition I.								

Table 1: Performance on gSCAN including models from the original publication Seq2seq and GECA (Ruis et al., 2020; Andreas, 2019) as well as other recent models AuxLoss (Heinze-Deml and Bouchacourt, 2020) and LCGN, the language conditioned graph network (Gao et al., 2020). The first row, condition A, does not represent generalization performance; it is the performance when the training and testing sentence distributions are the same. AuxLoss, LCGN, and our work are able to generalize to B and C. Our model is the only one to show any generalization in D. LCGN and our model have similar performance on E; note the very high variance of LCGN. Our model is the only one that addresses generalization condition G aside from Seq2Seq. While LCGN outperforms our model in H, we note its extremely high variance. No other work addresses generalization condition I.



(a) Walk to a big yellow cylinder while zigzagging

(b) Pull a small green circle

Figure 3: Two examples from gSCAN. The pink triangle is the agent with the tip of the triangle pointing forward. Red arrows show a trajectory. (a) A sentence that contains an action modifier. When testing novel adverb-verb combinations, the agent might separately see the concept "walk" and the concept "while zigzagging" in different sentences, but must infer what to do when concepts are combined during testing. (b) The agent must understand the target object, but size is relative. What is large on one map, might be small on another, depending on what other objects are available. In these test conditions, certain object sizes never appear labeled as large or small; this must be inferred from the context and then generalized to new sizes.

C *Red squares* holds out any references to an object, e.g., red squares are never referenced.

D *Novel direction* never refers to a object in a selected direction, e.g., the target is located at south-west of the agent.

E *Relativity* never refers to objects with a given relative size, e.g., what is small while training may be large when testing.

F *Class inference* requires inferring unstated properties, e.g., object size determines how many PULL actions are required to move it.

G *Adverbs* requires learning a word such as "cautiously" from a small given number of examples.

H *Adverb to verb* holds out pairs of verbs and action modifier, e.g., "walking" while "spinning".

I *Sequence length* generalizes to longer action sequences.

Models

We evaluate several variations of our compositional networks [1] against baseline models described in Ruis et al. (2020) (a seq2seq model and GECA introduced in Andreas (2019)) as well as two recent models discussed in the Related Work (Heinze-Deml and Bouchacourt, 2020; Gao et al., 2020). The seq2seq model encodes both the commands and the environment separately using a BiLSTM and a CNN. This is a common architecture used in many publications. GECA is a variant of the baseline seq2seq model which employs data augmentation to improve generalization.

We consider three variants of our full model, each using different parsers to structure the compositional networks. We use a pretrained constituency parser from AllenNLP (Gardner et al., 2017); a pretrained dependency parser from Stanza (Qi et al., 2020); and a semantic parser which rewrites the original grammar used to create gSCAN. These three models communicate using attention maps. All compositional networks presented in the evaluation contain a CNN with kernel size 7 and 50 channels and are trained with lexical semantics. Each word module is a GRU with 2 hidden layers and 20-dimensional hidden states. A component uses a linear layer to map the input observation and hidden state to dimension of 10, and the ReLU activation in MLP to predict the grayscale attention for each grid cell. We select hyperparameters that increase exact matches in the validation set. We train all networks using the Adam optimizer with the initial learning rate 0.001, β_1 0.9, and β_2 0.999.

[1] Source code is available at https://github.com/ylkuo/compositional-gscan

To demonstrate how critical a mechanism that makes modules mutually intelligible to one another when testing compositionality is, we test two other models. The first is a model that passes a 20-dimensional state vector instead of an attention map; it is referred to as *State*. Since the capacity of an attention map and an n-dimensional state vector cannot be matched, no matter what value n takes, we give the next variant an even more powerful representation. *+Attention* includes both the state vector and the attention map. This is strictly more powerful but fails to generalize well because other models cannot understand this side channel muddling the information being exchanged. Both ablations employ the semantic parser to compose the networks.

Results

Experiments were carried out on two machines, each with 80-core Intel Xeon 6248 2.5GHz CPUs, 768 GB of RAM, and 8 Titan RTX 24GB GPUs. Training and testing the models including ablations took approximately four days. Each model trained for 150,000 steps with batch size 200. Table 1 summarizes percentage of exact match and the standard deviation over three runs by generalization condition. Overall, the model using the semantic parser and attention maps significantly outperforms all other variants. An arbitrary state vector, *State*, passed between words performs very poorly, far worse than the non-compositional seq2seq model. It appears to be critical that there exists a method to make representations interpretable to models which have not been exposed to one another. Adding attention maps to the state, *+Attention*, results in better performance but still worse than passing attention maps only.

Only when we remove all arbitrary state and only exchange attention maps does compositionality shine through. The three models in the rightmost three columns of Table 1 generalize in most conditions. In cases where prior work such as AuxLoss and LCGN demonstrate generalization, our models achieve state of the art or close to state of the art performance. In cases such as conditions, D, G, and, as will be shown later, I, our model generalizes when others do not. AuxLoss and LCGN do not report results on G and I. Note that, our results in condition G show that our model only needs a handful of examples to achieve reasonable performance. This capability allows our model to scale to novel words and objects more quickly.

In most cases, networks based on the semantic parses outperform those based on syntactic parses. This may be because semantic parses are more stable than syntactic parses, i.e., similar concepts can have very different surface representations but their relationship is revealed in a deeper analysis. It could be that this phenomenon occurs for a much more interesting reason: semantic parses are designed to be useful for extracting the meaning of sentences. Perhaps, in the future, grounded agents can provide a completely independent and novel test for linguistic representations – a good representation is one where a robot is able to learn to perform well.

gSCAN includes a condition, I, that extends the dataset to longer sequences. Table 2 summarizes our performance on this condition comparing with Ruis *et al.* (Ruis et al., 2020). Note that other models do not report results for condition I. We have retrained Gao *et al.*'s model (Gao et al., 2020) for condition I and received 1.36 ± 0.34 exact match on the test set for over three runs. When the training sequence length and the test sequence length are the same, our model performs well, in line with the state of the art. As the sequence length increases, baseline seq2seq models lose all of their performance almost immediately. The performance of our compositional model does decay, but at a far slower rate. We can train with even shorter sequence lengths, 13 instead of 15, and still vastly outperform the state of the art at predicting move sequences of length 18.

Interpretability and acquisition

Our model is interpretable in two ways. (1) The structure of the network overtly encodes the structure of the sentence so a parser error can be observed directly. (2) The internal reasoning of the network proceeds by passing attention maps between modules. These maps can be directly inspected to see what different words or phrases are physically referring to. If an agent picks up the wrong object because the words that refer to that object attend to the wrong part of the map, this error will be evident from the attention maps. Figure 4 shows example attention maps which can be viewed as a series of selectors to filter the goal object to interact with. Furthermore, since the Small&Large attention maps are cotrained to have the opposite semantics, we can use them to infer the absolute scale of object sizes by post-processing

Target length	Seq2seq (Ruis et al., 2020)	Ours w/ Semantic parses		
	Train length ≤ 15	Train length ≤ 15	Train length ≤ 14	Train length ≤ 13
15	94.98 ± 0.12	92.06 ± 1.94	75.02 ± 8.33	66.11 ± 8.46
16	19.32 ± 0.02	89.05 ± 2.60	67.39 ± 9.61	59.93 ± 9.75
17	1.71 ± 0.38	85.08 ± 4.02	62.43 ± 9.84	56.14 ± 11.06
≥ 18	< 1	53.67 ± 4.00	34.01 ± 9.78	31.33 ± 10.28

Table 2: Performance on gSCAN as a function of the length of action sequences in the training set. State-of-the-art methods fail to generalize to longer sequences. Our model does, although not perfectly. Even as the training action sequence length is decreased, our model continues to generalize.

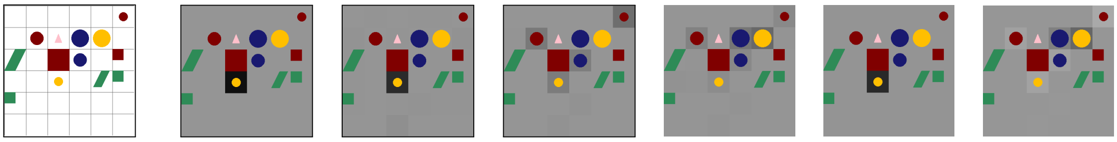

Initial environment *Walk* *Yellow* *Small* *Circle* *While spinning* *Size ordering*

Figure 4: Attention maps while executing *Walk to a small yellow circle while spinning*. Darker cells are areas of interest to models. Models refine the attention maps they receive as input from their children, e.g., *circle* attends to all circles, "small" filters there to small circles, and "yellow" focuses on the combination of all three. Since the models are informed by lexcial semantics (small and big are antonyms), we can infer the size ordering map, where lightness correlates with circle size.

the two maps: $(-att_{Small}+att_{Large})/2$. We can also inspect the attention maps across training epochs to see if the network acquires the meaning of the word and how the representations change over time. Figure 5 demonstrates the learning progression of the network through attention maps.

Conclusion

We have presented a model that addresses many of the compositionality challenges found in gSCAN, a dataset designed to challenge networks. When the compositionality inherent in a problem is reflected in the computation of a network, the resulting network is far better able to understand the target domain. This is only critical at test time, when generalizing to new combinations. An important caveat is that a mechanism for making representations of different word modules compatible with one another is key. Here we do this by constraining all communication through attention maps.

Performance of compositional approaches depends on what is being composed and how. When the compositionality does not capture part of a problem, such as condition D here, it does not meaningfully improve results. When compositionality is relevant, it appears that it can supplant data augmentation and provide a faster, principled, dataset-agnostic method to achieve better results. When compositionality is derived from language, it enables the inclusion of linguistic notions, e.g., synonyms and antonyms, in models.

The most suggestive and admittedly tenuous implication of this work is that perhaps we can use this approach to test linguistic representations. Many formalisms exist in linguistics for encoding semantics. Without an independent test for which is better, convergence to one formalism is unlikely. It appears that when compositional models are trained to perform tasks, some representations are significantly better than others. In our experiment, abstract representations, i.e., ones further from the surface syntax of language, result in better models. Perhaps in the future a meta-learning approach could allow grounded robotics to come full circle: from borrowing ideas from linguistics to contributing to our understanding of semantics.

In the meantime, robots and conversational agents will continue to be deployed. It is critical that we have confidence in our systems and that input merely being out of the training set does not cause catastrophic failure. We demonstrate one step toward achieving this goal: a principled way to enable networks to generalize out of the training set. Many open problems remain, key among them: is there a way to convert a data augmentation approach into a network architecture that sees through the problem and generalizes better for a principled reason without the data augmentation. This would be a powerful tool, which we suspect exists, but have not yet found.

Ethics and broader impacts

Robots that can competently understand natural language will provide access to technology for those who need it most: those who have physical limitations, those with limited access to education, etc. This can have tremendous positive impact as well as negative consequences. For example,

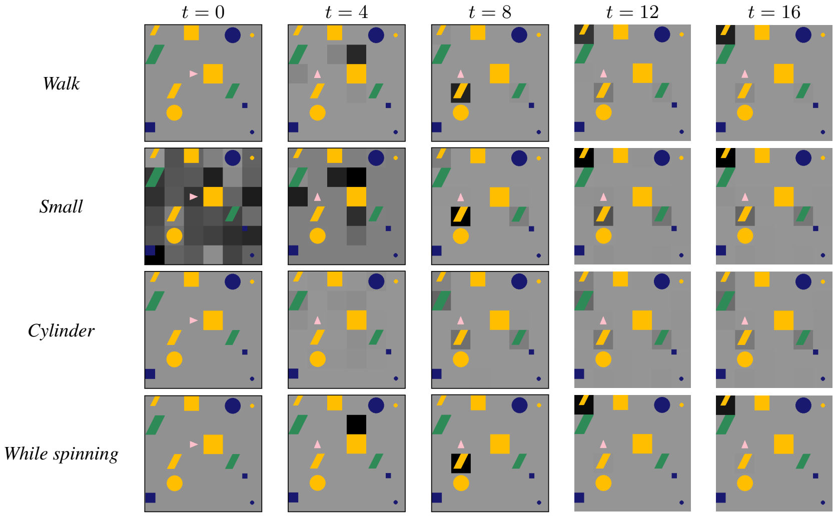


Figure 5: Visualization of the learning progression for the command "Walk to a small cylinder while spinning." Each row contains the attention maps produced by the word modules at different training epochs. The maps on the left are the beginning of the training, where the attentions are uniform or random. Toward the right, as the training epochs increase, the "Cylinder" module first identifies the shapes of the objects, and then the "Small" module takes longer time to learn to sort by the size of cylinders. By inspecting the attention maps over time, we can track if a module acquires the meaning of the word and what it is confused about, for example, the "Small" module at $t = 8$ can identify the smaller cylinders but confused about the size ordering.

such robots may displace human workers leading to widespread job loss. We already see this in that bots are taking over many interactions that would otherwise have gone through a customer support representative. The future impact of language-driven robots and conversational agents will depend on a combination of researchers who tailor systems to augment rather than displace workers as well as politicians who create safety nets and training for displaced workers.

Our adoption of methods which attempt to be transparent, i.e., by forcing the models to reason in the open through attention maps, can help with pinpointing errors. Currently, complex systems, end-to-end models in particular, have an attribution problem. One is largely uncertain about why they fail. A robot that harms someone, one that discriminates overtly or covertly, etc. should be designed in such a way that one can determine why these actions were taken, to assign financial and legal liability as we do with all other engineered systems.

Acknowledgments

This work was supported by the Center for Brains, Minds and Machines, NSF STC award 1231216, the MIT CSAIL Systems that Learn Initiative, the CBMM-Siemens Graduate Fellowship, the DARPA Artificial Social Intelligence for Successful Teams (ASIST) program, the United States Air Force Research Laboratory and United States Air Force Artificial Intelligence Accelerator under Cooperative Agreement Number FA8750-19-2-1000, and the Office of Naval Research under Award Number N00014-20-1-2589 and Award Number N00014-20-1-2643. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Ferran Alet, Tomas Lozano-Perez, and Leslie Pack Kaelbling. 2018. Modular meta-learning. In *Conference on Robot Learning (CoRL)*.

Jacob Andreas. 2019. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Miguel Ballesteros and Joakim Nivre. 2013. Squibs: Going to the roots of dependency parsing. *Computational Linguistics*, 39(1):5–13.

Andrei Barbu, Alexander Bridge, Zachary Burchill, Dan Coroian, Sven Dickinson, Sanja Fidler, Aaron Michaux, Sam Mussman, Siddharth Narayanaswamy, Dhaval Salvi, et al. 2012. Video in sentences out. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Valts Blukis, Dipendra Misra, Ross A Knepper, and Yoav Artzi. 2018. Mapping navigation instructions to continuous control actions with position-visitation prediction.

Ben Bogin, Sanjay Subramanian, Matt Gardner, and Jonathan Berant. 2021. Latent compositional representations improve systematic generalization in grounded question answering. *Transactions of the Association for Computational Linguistics*, 9(0):195–210.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. 2018. Gated-attention architectures for task-oriented language grounding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2018. BabyAI: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations*.

Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Neural Modular Control for Embodied Question Answering. In *Proceedings of the Conference on Robot Learning (CoRL)*.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Tom Griffiths, and Alexei Efros. 2018. Investigating human priors for playing video games. In *International Conference on Machine Learning*.

Christiane Fellbaum. 2012. Wordnet. *The encyclopedia of applied linguistics*.

Justin Fu, Anoop Korattikara, Sergey Levine, and Sergio Guadarrama. 2019. From language to goals: Inverse reinforcement learning for vision-based instruction following. In *The International Conference on Learning Representations*.

Tong Gao, Qi Huang, and Raymond Mooney. 2020. Systematic generalization on gSCAN with language conditioned embedding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform.

Sergio Guadarrama, Erik Rodner, Kate Saenko, Ning Zhang, Ryan Farrell, Jeff Donahue, and Trevor Darrell. 2014. Open-vocabulary object retrieval. In *Robotics: Science and Systems*.

Christina Heinze-Deml and Diane Bouchacourt. 2020. Think before you act: A simple baseline for compositional generalization. *arXiv preprint arXiv:2009.13962*.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. 2017. Grounded language learning in a simulated 3D world. *arXiv preprint arXiv:1706.06551*.

Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. 2010. Toward understanding natural language directions. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.

Yen-Ling Kuo, Boris Katz, and Andrei Barbu. 2020. Deep compositional robotic planners that follow natural language commands. In *International Conference on Robotics and Automation*.

Brenden Lake and Marco Baroni. 2018. Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks.

Joël Legrand and Ronan Collobert. 2014. Joint rnn-based greedy parsing and word composition. *arXiv preprint arXiv:1412.7028*.

Percy Liang and Christopher Potts. 2015. Bringing machine learning and compositional semantics together. *Annual Review Linguistics*, 1(1):355–376.

Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. 2013. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*. Springer.

R Thomas McCoy, Junghyun Min, and Tal Linzen. 2019. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2016. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Sandro Pezzelle and Raquel Fernández. 2019. Is the red square big? MALeViC: Modeling adjectives leveraging visual contexts. 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)*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. 2020. A benchmark for systematic generalization in grounded language understanding. In *Advances in Neural Information Processing Systems*.

Pararth Shah, Marek Fiser, Aleksandra Faust, J Chase Kew, and Dilek Hakkani-Tur. 2018. FollowNet: Robot navigation by following natural language directions with deep reinforcement learning. *arXiv preprint arXiv:1805.06150*.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y Ng, and Christopher D Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. In *International Conference on Machine Learning*.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Twenty-fifth AAAI Conference on Artificial Intelligence*.

Haonan Yu, Haichao Zhang, and Wei Xu. 2018. Interactive grounded language acquisition and generalization in a 2D world. In *The International Conference on Learning Representations*.