SeqG(SC)AN

Chang Ye, Govind Mittal, Yada Pruksachatkun, Lee Qianqian Cui
New York University
{cy1365, gm2724, yp913, qc697}@nyu.edu

***Abstract*—Generalization from one domain to another is one of the hardest tasks in machine learning. Unlike humans, who can learn infer new concepts from limited data, machines necessitate large training datasets to learn effectively. This work explores grounded compositional generalization via the gSCAN dataset, and probing GANs for the ability to transfer in zero-shot domains. This builds upon prior work that uses supervised learning to train the model to infer sequences of commands grounded in a grid world. In this paper, we propose the use of a reinforcement learning model for this task, as well as conduct human baseline experiments on the task to understand how humans and languange models learn compositionality.**

I. Introduction

Humans have the ability of language acquisition from limited data [1, 2]. Children demonstrate acquisition of inflectional morphemes in language, as well as an understanding of contextual information at a young age. In addition, children are able to use contextual references to understand the size of a given object, and to generalize these references to sentences with novel words.

In this work, we explore the utility of generative adversarial network (GAN) based training for sequence generation on gSCAN, a dataset that tests for grounded linguistic generalization for machine learning models. Secondly, we run a human experiment on the dataset to approximate a human baseline to the various subsets of generalisation tested by the dataset.

Human development of representation learning has been reported to be item-based during early childhood [3], which becomes incrementally abstract over time [4]. Empirical studies have shown that generalization by young children is brittle across various kinds of language tasks [5].

One hypothesis, known as the critical mass hypothesis [6, 7], explains this phenomena as being due to the fact that adults have more exposure to the requisite data than young children.Other proposals suggest that there should be factors that significantly influence the ability of linguistic generalization, other than the amount of exposure to examples [4]. Even when measures are introduced to control for the exposure effect and to make the experiments similar across participants, adults and older children are able to form abstract constructional representations and outperform their younger counterparts. In addition, older participants perform well when the items in the test phase are relatively high in novelty [4], as well as when there are systematic reversal shifts involved (e.g., all blue shapes are labeled 'winners' and all yellow shapes labeled 'losers'). Children under four years old often fail to point out who is the winner or loser when the roles are reversed). As suggested by empirical researches, humans' inability in linguistic tasks featuring few-shot learning might be a possible result of low cognitive flexibility [8] or impairment in memory systems [9].

We are interested both the ability and inability to generalize with a focus of computational cognitive modeling approach. We are also curious about how a cognitive model performs in the few-shot learning condition. In this paper, we will propose a novel model and attempt to understand current obstacles in compositional generalization.

II. Related Work

A. Machine Learning

Goodfellow et al. [10] proposed Generative Adversarial Networks (GANs), which is based on a mini-max strategy in game theory. It is composed of two networks, one being the generator and the other being the discriminator. The role of the generator is to generate synthetic data that is as real as possible, while the role of discriminator is to discern fake data generated by generator from the real data in the training dataset. These two networks compete with each other to achieve the Nash equilibrium in the training process. Generative adversarial networks have shown great success in natural image generation. However, GANs cannot be easily applied to natural language generation, as GANs only work on continuous space whereas language lies in a discrete space [11, 12]. However, there has been some success with GANs in machine translation [13].

Reinforcement Learning is a subfield of machine learning that is based on the Markov property, in which the current state only depends on the previous state and action a model takes. There are two mainstream algorithms in reinforcement learning: model-based reinforcement learning, where the agent models the transition function and reward function, and model-free, where the agent learns the policy directly, such as with Q-learning [14] or Policy Gradient. Recently, Yu et al. [15] proposed the SeqGAN framework, which received great attention, and combines ideas from both GANs and reinforcement learning. It models the sequence generation problem as a sequence decision making process[16], modeling the generative model as a policy.

B. Cognitive Science

Previous cognitive science work on language [17, 18] and recent experiments on artificial compositional instruction learning from Brenden et. al. [19] have demonstrated that humans can generalize actions from a finite amount of examples. The artificial compositional instruction learning experiment also showed that humans have the impressive ability to learn functions from limited experience and generalize to novel inputs, and to compose functions together to interpret novel sequences of instructions.

Lake and Baroni [20] explore machine learning capabilities for generalization by introducing SCAN, a dataset that probes a model's ability to generalize on a range of different commands presented in simplified English grammar. Later, Ruis et al. [21] proposed a grounded SCAN dataset which in addition to the input and target commands has a visual input to provide a "world" for the agent to interpret the commands with. A multi-modal baseline model and a state-of the-art compositional method were tested for their efficiency and flexibility to extract compositional rules, but the results demonstrated that they fail in most cases when systematic compositional rules are involved in the generalization process.

III. Dataset

We use the gSCAN dataset[1], which consists of input commands $x_1, x_2, ..., x_n$, target commands $y_1, y_2, .., y_m$, and a vector representations of a world that consists of a 6×6 square grid and various objects placed inside the grid. The train dataset consists of 367933 examples, while the dev set consists of 3716 examples. gSCAN provides a test set with various splits of varying sizes (from 11460 to 112880 examples) that each test for various generalization capabilities, which are described as follows:

- **Random**: No systematic differences in training and test set
- **Yellow squares**: The training set contains yellow square targets that are referred without color, for example only as 'the square', 'the small square', or 'the big square'. The test set, on the other hand, refers to the yellow square with the 'yellow' adjective.
- **Red squares**: The training set only contains the red square as a non-target background target, while the test set also contains the red square as a target.
- **Novel Direction**: The training set holds out all examples where the target object is located to the south-west of the agent, while the test set includes this direction.
- **Relativity**: The size 2 objects are not targets and are not referred as 'small' in training set where the circle of size 2 is the target and the smallest circle in the world, paired with an instruction containing the word 'small'.
- **Class inference**: The training set contains verb 'pull' in the instruction with the target object square of size 3, whereas the test set has the same target but with verb 'push' in the instruction.
- **Adverbs**: There are two test splits that test for adverb generalization. The first tests on generalization on the adverb 'cautiously', while the second tests for generalization to a adverb-verb combinations, in which the training set contains the adverbs and verbs separately (such as 'while spinning' and 'pull', and the test set contains them together in the same input command ('pull while spinning'). The training set only contains a single example of instructions with the adverb 'cautiously'. And the test set contains all kinds of instruction with that adverb. The second is the training contains either tested adverb or verb. The test set contains both of them.

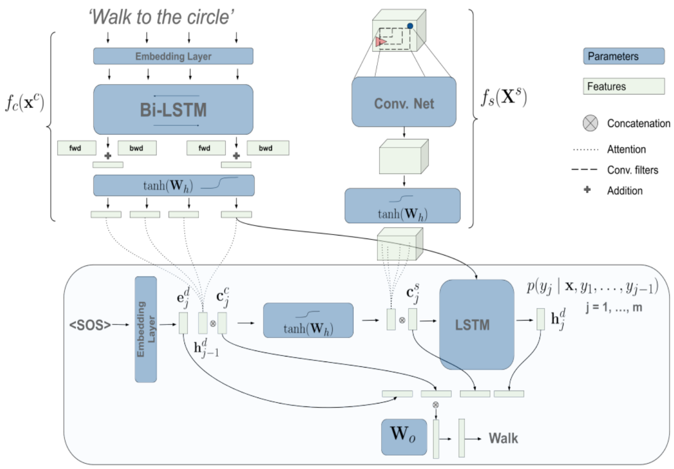


Fig. 1. The architecture diagram for the generator. The command encoder processes the command, the state decoder processes the world and the decoder generates target action sequence [21].

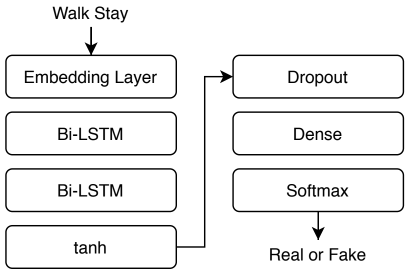


Fig. 2. The architecture diagram for discriminator. A generated sequence from the generator is passed on as input and label is returned by discriminator.

IV. Method

We frame this task as a reinforcement learning problem, where each command token predicted in the predicted sequence is framed as an action, and each state is a sequence of tokens. With that in mind, $p(s_{t+1}|s_t, a_j) = 1$ for actions a_j and state s_t, since s_{t+1} consists of the predicted sequence of tokens $b_1, ..., b_{t+1}$. We employ a method inspired from SeqGAN [15] to train our model as a GAN. The algorithm was chosen because it enables us to update the loss with partially decoded sequences and allow discriminators to assess partially decoded sequences too, which has been one of the weaknesses of traditional GAN based training procedures.

The generator is similar to the model in the gSCAN paper (shown in Figure 1). It consists of three components:

- The **command encoder** embeds the input into a vector space and engineers features using a bi-directional LSTM.

[1] https://github.com/LauraRuis/groundedSCAN

- The **state encoder** is a convolutional network which processes the grid world state.
- The **decoder** produces the target action sequence using joint attention on the output of command encoder and state encoder.

For the discriminator (shown in Figure 2), we use a bi-directional LSTM model, as they take into account the highly sequential nature of a command sequence output and have been shown to be effective in natural language processing classification tasks [22, 23]. The input to the discriminator is embedded in a multi-dimensional vector space, and these vector representations are passed to bi-directional LSTM layers with dropout.

We follow the training formulation purposed by Yu et al. [15]. The following steps are taken in order:

1) Pre-train generator G using negative log likelihood loss.
2) Labeling samples from pre-trained generator G as negative and labeling samples from training dataset as positive ones, we pre-train the discriminator based on a supervised objective. Its goal is to predict which sequences are from the real dataset and which are generated by the generator.
3) After pre-training G and D, we start adversarial training where G and D are trained alternately. Adversarial training proceeds as follows:
   - G is trained using the REINFORCE algorithm. After sampling from G, we calculate a reward for the sample. For each incomplete sequence in the sample sequence, we generate the reward with the below equation,

     $$R = \frac{1}{N} \sum_{j \in [1,n]} D(\text{MCrollout}(y_1, .., y_j)). \tag{1}$$

     where $y_1, y_2, ..., y_j$ is the incomplete sequence and D is the discriminator.

     We use the generator as our Monte Carlo Rollout (MCRollout) policy. The policy will complete each incomplete sequence by generating the prospective remaining tokens based on the incomplete token sequence. For the completed sequences, we simply return the predictions from discriminator on all the possible completed predictions.
   - We then use policy gradient to propagate back the loss to G, where the loss is defined as follows for the action y_t and previous actions $y_1, y_2, ..., y_{t-1}$:

     $$\nabla L = E_{y_t \sim G(y_t|Y_{1:t-1})} \nabla_\theta R(Y_{1:t-1}, y_t) \log G_\theta(y_t|Y_{t-1}) \tag{2}$$

   - D is trained in a similar way to its pre-training (Step 2), by using the binary cross entropy loss on positive samples from training set and negative samples from G. We only train this discriminator once after training the generator during adversarial training, following the SeqGAN paper.

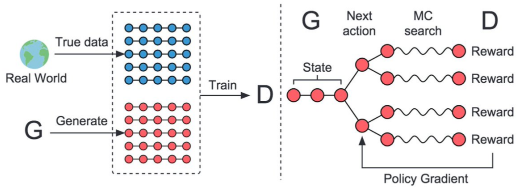


Fig. 3. The architecture diagram for SeqGAN [15].

V. Experiments

A. GAN experiments

The input sequence to the discriminator is embedded in a *300-dimensional* vector space, and these vector representations are passed to two bi-directional LSTM layers with 512 hidden units each and dropout rate of 20%.

The generator takes input commands from a vocabulary of size 21 ({SOS, EOS, 'walk', 'to', 'a', 'yellow', 'small', 'cylinder', 'hesitantly', 'while spinning', 'while zigzagging', 'circle', 'big', 'green', 'square', 'red', 'blue', 'push', 'pull', 'cautiously', PAD}) and outputs samples from a vocabulary of size 6 ({SOS, EOS, 'turn right', 'walk', 'stay', 'turn left', 'push', 'pull', PAD}), with Bahdanau joint attention [24]. The command encoder embeds the input command onto a 25-dimensional vector space and uses 100 hidden units in the bi-directional LSTM. The state encoder consists of a stack of three two-dimensional convolutional neural networks with increasing kernel size, constant stride of 1 and same padding.

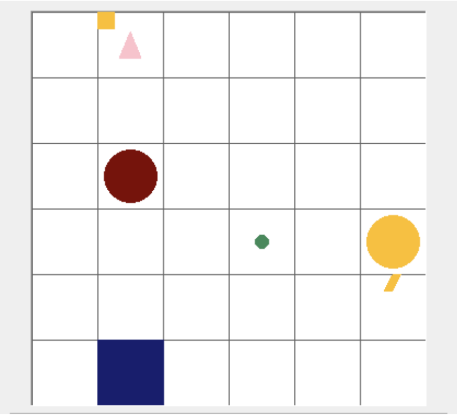
Both generator and discriminator get a separate optimizer and scheduler. We pre-train the generator for 10 epochs each, before training the discriminator for 10 epochs with the pretrained generator. We performed a hyperparameter search for the learning rate with values from {1-e4, 1-e3, 1-e2, 5e-4, 5e-5, 0.1}, with learning rate decay steps of 200 steps and batch size of 1500 samples. The adversarial training is run for 6 epochs, when the loss of the discriminator and generator converges to almost a constant value.

The training data was always shuffled after seeding with a value of 66.

B. Human trials

Additionally, we ran an experiment with human participants to glean insight into how the process of learning in humans differs from that of machine learning models. In order to reduce bias of the common words used in the dataset, we mapped verbs, adverbs, and adjectives to gibberish words, for example, 'walk' with 'fraule'. We sampled 30 examples from the training set, and 22 examples from each of the test set splits. An example of the mapped gibberish words is displayed in Table I.

We recruited 14 volunteers, the majority of whom are college students. They were presented with the training set command alongside the corresponding grid animation and target command during the learning phase. Each training example was viewed only once. An example of the grid



Input command: fraule to a eigeracy ribunk
Target sequence: fraule, leftme, fraule,fraule, fraule

Fig. 4. A visualized example in the human experiment

TABLE I
AN EXAMPLE OF THE MAPPED GIBBERISH WORDS USED IN THE HUMAN SUBJECT EXPERIMENT

Original Word	**Corr. Gibberish Word**
Yellow	eigeracy
Cylinder	gligatte
Small	kindark
Walk	fraule
While spinning	quototapely
Cautiously	doctly
Walk to a circle	fraule to a hammano

animation is shown in Figure 4. After the training, we gave subjects a grid and input command, and asked them to generate the target sequences in a Google Spreadsheet, with each row corresponding to each test set example. At the end of this experiment, we also evaluated their declarative knowledge about the gibberish words by asking them to guess the meaning of each word. We display the instructions shown in the Appendix.

VI. RESULTS

A. GAN-based trials

Our results are in Table II. We see that while our models are unable to exactly match the target sequence, they were able to start to learn some generalization knowledge, as shown by the accuracy, even with 6 epochs of training. Most significantly, for the class inference test split, our model was able to get an accuracy of 4.0 for the random split and negligible performance on other splits. This means that on average, 4% of the predicted target sequence generated for each example matched with the gold sequence for that example.

TABLE II
EXPERIMENT RESULTS (PERCENTAGE OF EXACT MATCH)

SPLIT	BASELINE (EXACT MATCH)	GAN (EXACT MATCH (%))	GAN (ACCURACY (%))	HUMAN SUBJECTS (ACCURACY (%))
RANDOM	97.69	12.0	0.40	87.93
YELLOW SQUARES	54.96	0.0	0.20	78.25
RED SQUARES	23.51	0.0	0.21	82.61
NOVEL DIRECTION	0.00	0.00	0.00	71.38
RELATIVITY	35.02	1.0	0.16	85.33
CLASS INFERENCE	92.52	0.0	4.00	85.24
ADVERB ‘CAUTIOUSLY’	0.0	0.0	0.00	25.70
ADVERB ‘WHILE SPINNING’	22.70	0.0	0.00	21.93

B. Human trials

The results reveal that subjects successfully form a mental association between the input command and the actual actions and objects in the grid world. Figure 5 illustrates the average accuracy rates based on different task splits. Human subjects reach a high average accuracy (above 80%) on tasks involving relativity and red squares. However, the mean accuracy for tasks involving adverbs, such as, ‘cautiously’ and ‘while spinning’ drop dramatically. The average accuracy for adverb ‘cautiously’ split is 25.7% and the average accuracy for adverb ‘while spinning’ split is 21.9%.

A drop in the adverb accuracy is presumably caused by the limited training and testing trials. During the learning phase, the frequencies of the adverbs were restricted according the task splits. Novel adverbs were introduced in the testing phase. Subjects were expect to have limited prior knowledge on testing examples in the tasks such as ‘while cautiously’ split. We expect that reaching a better performance on generating sequences with those adverbs possibly requires more exposure. Yet it is still significant that humans form both implicit associations and explicit knowledge with limited learning examples in a short time period.

A post task questionnaire asked the subjects to guess the meaning of each mapped gibberish word. We labeled a response as correct when it contains the exact right answer or any corresponding synonyms. As shown in Figure 6, subjects displayed significantly higher accuracy for nouns (mean accuracy = 94.7%), verbs (mean accuracy = 92.8%), adjectives (mean accuracy = 85.7%) and adverbs (mean accuracy = 35.7%).

It is worth noting that 64.2% of the subjects correctly guessed that the gibberish word ‘quototapely’, which means ‘while spinning’. However, over half of them failed to generate

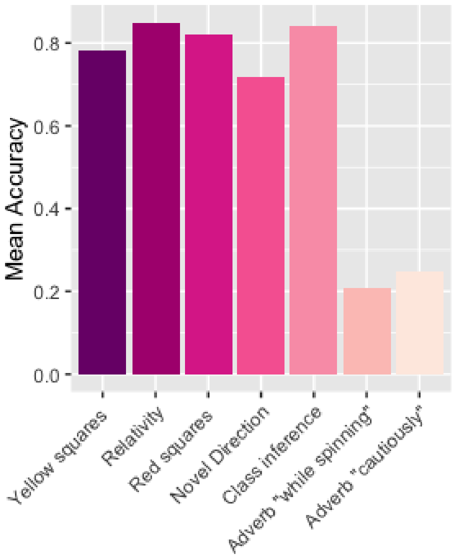


Fig. 5. Mean Accuracy for Task Split

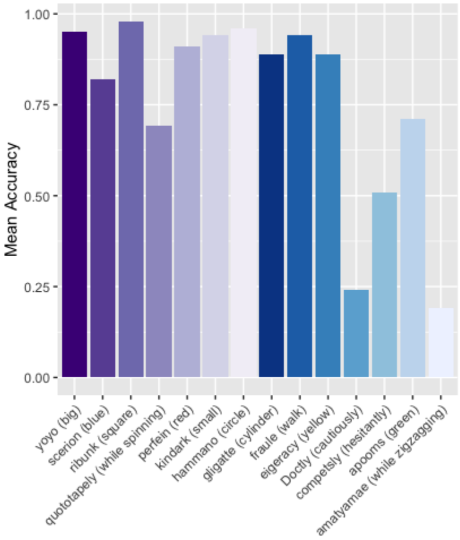


Fig. 6. Mean Accuracy of Post-Experiment Questionnaire on Word Meanings

the exact correct sequences for the ’while spinning’ tasks during the test phase. This suggests that people are able to make some above chance predictions for words in the few-shot learning condition, but they fail to demonstrate a sophisticated understanding of the detailed rules when training examples are limited. This indicates a potential gap between forming an abstract mental representation and learning the exemplar-specific rules.

VII. Discussion

Treating the compositional generalization task as a reinforcement learning problem did not yield good results. The GAN-based sequence generator did not converge to an expected global minima but even after multiple tries and several epochs always converged to the same sub-optimal local minima. This may be due to time and resource constraints, as Yu et al reported that SeqGAN only starts to see significant increases in performance after 10 epochs. There are also optimization difficulties in training GANs on discrete spaces such as commands and. language.

In terms of our human trials, we found that while on average, there was high accuracy in tasks, there was a drop in average accuracy in adverb-related tasks. Interestingly, over half of the subjects correctly guess the meaning of the gibberish adverb corresponding to ‘while spinning’, but the majority of them fail to produce the exact correct sequences in the ‘while spinning’ task. This may suggest a gap between forming an abstract mental representation and understanding the exemplar-specific rules. This may be because in being presented with adverbs in conjunction with other parts of commands, humans may struggle to discern between various components of a command. Thus, this may suggest further research direction in using continual learning and presenting various sub-tasks to machines, where each subtask is to learn one component of a command (e.g., adjective, action, or adverb), fixing for all other components in the training of each subtask.

VIII. Conclusion

This work demonstrates the failure of using reinforcement learning to train sequence generation problem, and the ability of humans to learn compositionality. Our human experiments clearly indicate that prior knowledge is important to few shot learning. Future work could adopt meta-learning or continual learning into this problem, in order to learn the subtasks necessary to undertanding input commands. The prior learned from other tasks might inherently contain some relation or latent information for that non-target object that would help models perform new tasks in a few-shot manner. From our human trials, we also see that humans also relatively perform worse in generalization of adverbs, in comparison to other aspects of a command. This is consistent with the findings in neural network models on gSCAN, suggesting that adverbs require more explicit training than other aspects of a command.

There are a few optimizations that could be done to improve performance. Firstly, it may make sense to pass in the input command in conjunction with the target command into the discriminator, as the correctness of a target sequence can only be evaluated based on the input command it is given. Secondly, in the original formulation of SeqGAN, the discriminator is always trained by sampling the generator for negative samples. However, as the generator improves over time, this may not be the most optimal. Thus, there may be an annealing-type of scheduler that over time, will randomly assume more and more of the sampled model outputs as positive samples.

There is also another option to adopt the reinforcement learning technique. Instead of providing the initial observation to network, we can feed the output action into the gym environment at every step when generating new tokens and then feed the observation returned from gym environment back into the visual encoder. This approach might be able to generalize better as the agent will receive an updated observation.

Thirdly, the results from our human experiments clearly indicate a possible gap between constructing mental representation and understanding exemplar-specific rules. The importance of prior knowledge in the few-shot learning condition is also suggested. Future work could adopt meta-learning into this problem. There are lots of similarities in each task. Such as the novel adverb task where the network might encounter the situation before but is not the target of that task. So the prior learned from other tasks might inherently contains some relation or latent information for that non-target object and is able to learn to perform new tasks in a few-shot manner.

Code

The code is available at `https://github.com/pruksmhc/SeqgSCAN`.

References

[1] Adele E Goldberg, Devin M Casenhiser, and Nitya Sethuraman. Learning argument structure generalizations. *Cognitive linguistics*, 15(3):289–316, 2004.

[2] Devin Casenhiser and Adele E Goldberg. Fast mapping between a phrasal form and meaning. *Developmental science*, 8(6):500–508, 2005.

[3] Michael Tomasello. The item-based nature of children's early syntactic development. *Trends in cognitive sciences*, 4(4):156–163, 2000.

[4] Jeremy K Boyd and Adele E Goldberg. Young children fail to fully generalize a novel argument structure construction when exposed to the same input as older learners. *Journal of child language*, 39(3):457–481, 2012.

[5] Martin DS Braine and Melissa Bowerman. Children's first word combinations. *Monographs of the society for research in child development*, pages 1–104, 1976.

[6] Elizabeth Bates, Virginia Marchman, Donna Thal, Larry Fenson, Philip Dale, J Steven Reznick, Judy Reilly, and Jeff Hartung. Developmental and stylistic variation in the composition of early vocabulary. *Journal of child language*, 21(1):85–123, 1994.

[7] Michael Tomasello. First steps toward a usage-based theory of language acquisition. *Cognitive linguistics*, 11(1/2):61–82, 2000.

[8] Laura J Pauls and Lisa MD Archibald. Executive functions in children with specific language impairment: A meta-analysis. *Journal of Speech, Language, and Hearing Research*, 59(5):1074–1086, 2016.

[9] Barbara J Knowlton and Larry R Squire. Artificial grammar learning depends on implicit acquisition of both abstract and exemplar-specific information. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(1):169, 1996.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[11] Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015.

[12] Goodfellow. Generative adversarial networks for text.

[13] Jiatao Gu, Daniel Jiwoong Im, and Victor O. K. Li. Neural machine translation with gumbel-greedy decoding. In *AAAI*, 2018.

[14] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[15] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[16] Philip Bachman and Doina Precup. Data generation as sequential decision making. In *Advances in Neural Information Processing Systems*, pages 3249–3257, 2015.

[17] Noam Chomsky and David W Lightfoot. *Syntactic structures*. Walter de Gruyter, 2002.

[18] Richard Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970.

[19] Brenden M Lake, Tal Linzen, and Marco Baroni. Human few-shot learning of compositional instructions. *arXiv preprint arXiv:1901.04587*, 2019.

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

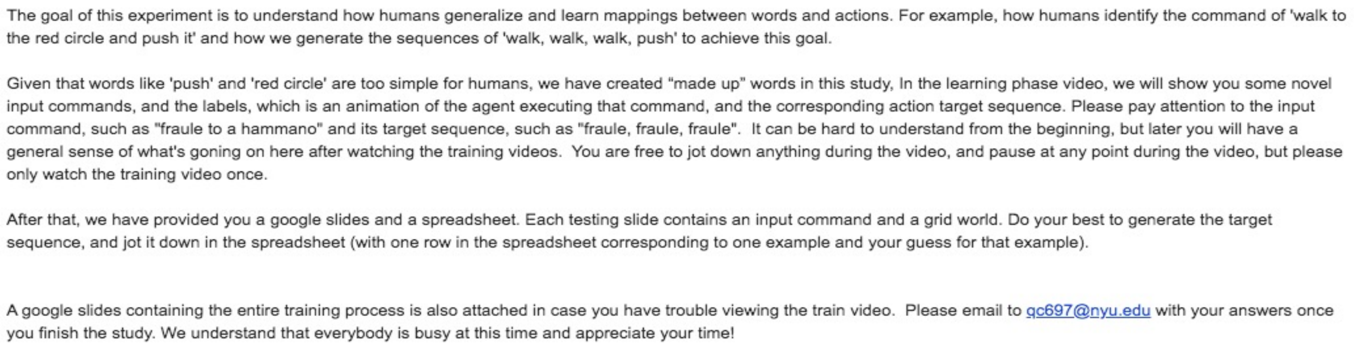
[21] Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. *arXiv preprint arXiv:2003.05161*, 2020.

[22] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *ArXiv*, abs/1812.09449, 2018.

[23] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45:2673–2681, 1997.

[24] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

IX. APPENDIX



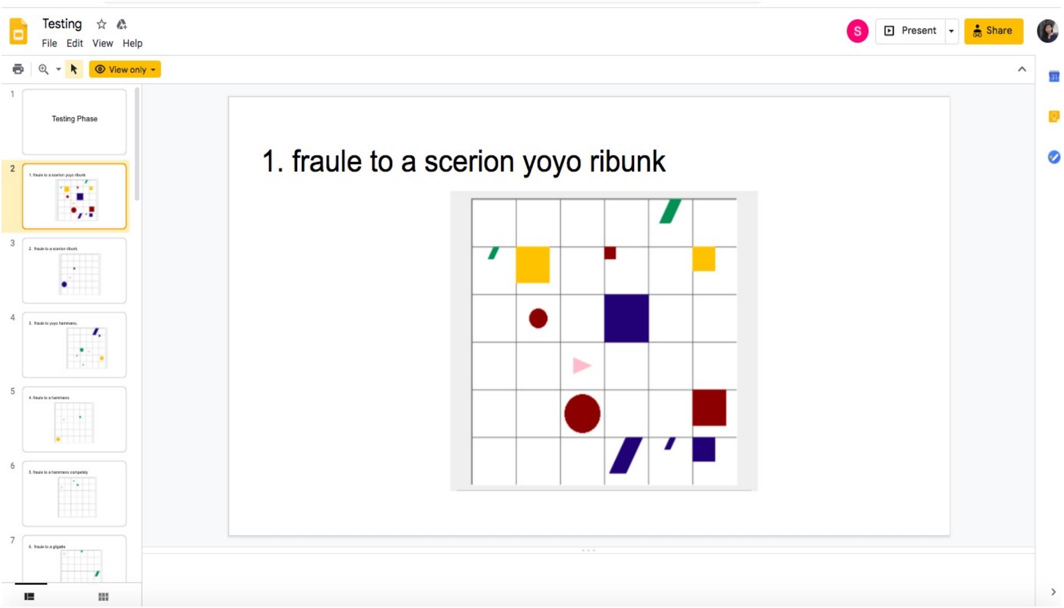
The goal of this experiment is to understand how humans generalize and learn mappings between words and actions. For example, how humans identify the command of 'walk to the red circle and push it' and how we generate the sequences of 'walk, walk, walk, push' to achieve this goal.

Given that words like 'push' and 'red circle' are too simple for humans, we have created “made up” words in this study, In the learning phase video, we will show you some novel input commands, and the labels, which is an animation of the agent executing that command, and the corresponding action target sequence. Please pay attention to the input command, such as "fraule to a hammano" and its target sequence, such as "fraule, fraule, fraule". It can be hard to understand from the beginning, but later you will have a general sense of what's goning on here after watching the training videos. You are free to jot down anything during the video, and pause at any point during the video, but please only watch the training video once.

After that, we have provided you a google slides and a spreadsheet. Each testing slide contains an input command and a grid world. Do your best to generate the target sequence, and jot it down in the spreadsheet (with one row in the spreadsheet corresponding to one example and your guess for that example).

A google slides containing the entire training process is also attached in case you have trouble viewing the train video. Please email to qc697@nyu.edu with your answers once you finish the study. We understand that everybody is busy at this time and appreciate your time!

Fig. 7. Email with instructions sent to participants


Testing
File Edit View Help
View only
S
Present
Share
1
Testing Phase
2
1. fraule to a scerion yoyo ribunk
3
2. fraule to a scerion ribunk
4
3. fraule to yoyo hammano,
5
4. fraule to a hammano
6
5. fraule to a hammano competely
7
6. fraule to a gigatte

1. fraule to a scerion yoyo ribunk

Fig. 8. Example of a test example shown to participants for prediction