Adding Grammatically Incorrect Instructions and Curriculum Learning to Improve The Novel Contextual Reference Test of gSCAN

Agustín Macaya
Pontificia Universidad Católica de Chile
aamacaya@uc.cl

Abstract

The Visual-and-Language Navigation (VLN) Problem is one of the biggest challenges for machine learning. One of the main difficulties is the Grounding Language Problem. Models need to learn to ground and generalize information that is contained in the instruction with the visual information given by the context. The 'Novel Contextual References Test' in the gSCAN benchmark shows that the model dramatically fails to understand relativity language and generalize the meanings of *big* and *small*. This paper proposes two solutions for the problem: a modification to the training dataset using grammatically incorrect instructions and a curriculum learning technique. Results show an improvement in exact match accuracy from 43.18% in the original benchmark to 67.10%. We conclude that the main factor affecting the "Novel Contextual References Test" in gSCAN and the Grounding Language Problem in VLN is not the model but the way the training examples are created and presented to the model during the training stage.

1 Introduction

Humans evolved to gather information from the environment and interact with it. We can do so by moving around, observing, and asking questions to other people. Since we are children, we rapidly learn to hear verbal instructions from other humans and perform those instructions correctly by moving through the space that we are in and interacting with the different objects that surround us. Furthermore, we can successfully generalize and deal with novel situations using only previous knowledge and language guidance. This integration between natural language, navigation, vision, and interaction is taken for granted by humans. However, the Vision-and-Language Navigation (VLN) Problem, where agents are asked in natural language to move through space and perform different actions, is still a challenge for machine learning algorithms.

Results at: https://github.com/aamacaya/IIC3692_project_gscan.git.

One of the biggest challenges for VLN is the Language Grounding Problem. This problem appears in VLN when the agent needs to ground the information that is contained in the instruction with the visual information. For example, the same object can be described as *big* or *small* depending on the size of the objects that are surrounding it.

The grounded SCAN (gSCAN) framework [11] is a benchmark that allows evaluation for systematic generalization in grounded language understanding in the VLN problem. In this benchmark, given an instruction in natural language, a 2D agent needs to move and perform actions on objects with different shapes, colors, and sizes. This benchmark has a test called *Novel Contextual References* that shows that the model dramatically fails to understand relativity language and generalize the meanings of *big* and *small*. An object that was always the biggest one in the situation during training cannot be recognized by the agent at test time when the same object is the smallest one in context. The hypothesis is that the model never learns the meanings of *big* and *small* and only relies on other characteristics that do not depend on context (shape and color) to find the target object.

This paper tries to show that the problem is not the model but the examples of the training dataset and the order in which those examples are shown during training. I propose two solutions for this problem that do not involve changing the model. First, I create a new expanded dataset that contains examples with instructions that, even though they are not grammatically correct, force the model to learn the relative size concepts of *big* and *small*. This is because some instructions no longer contain information about the color and shape of the target object, and only contain information about the relative size, forcing the model to learn those concepts and not rely on other intrinsic characteristics of the target object. Second, this paper proposes a curriculum learning technique [3] where examples are shown in a specific order during training to help the model learn the concepts of shape, color, and size. I start training the model using instructions that contain only one concept, so the model is forced to learn it without the ability to rely on the other ones, and then I add instructions that contain more than one concept so that the model learns to relate them. Results show that accuracy in the *Novel Contextual References* test set improves from 43,18% In the original benchmark to 67,10% using both solutions proposed.

This paper is organized as follows. Section 2 describes previous work and gives background information about the Visual-and-Language Navigation Problem, the Language Grounding Problem, the gSCAN benchmark, and Curriculum Learning. Section 3 explains the details of the problem with the original benchmark approach and the proposed solutions. Section 4, presents how the dataset was modified and how the model was trained. Section 5 shows and discusses the results of the experiments. Finally, Section 6 presents the main conclusions of this work.

2 Related Work

2.1 Vision-Language Navigation and Language Grounding Problem

Vision and language navigation (VLN) [1] is a sequence to sequence task [13] where the input is a natural language instruction accompanied by visual information. The objective is for an agent to navigate towards a goal location and possibly to perform a task in that location.

VLN is similar to Visual Question Answering (VQA) in the sense that both tasks have a natural language input, a visual input, and both of them have to output

another sequence. VLN differs from VQA in two ways. First, the output sequence of VQA is a natural language sequence and the VLN output is an action sequence. Second, in VQA the visual input is just one image, but in VLN, the visual input is updated as the agent navigates the environment. Both VLN and VQA meet in Embodied Question Answering (EQA) [6]. On EQA the input is a question on natural language and the agent must navigate through the environment, gather information, and then answer the question.

The symbol (language) grounding problem [7] appears in VLN when the agent needs to ground the information that is contained in the instruction with the visual information. There are two main challenges.

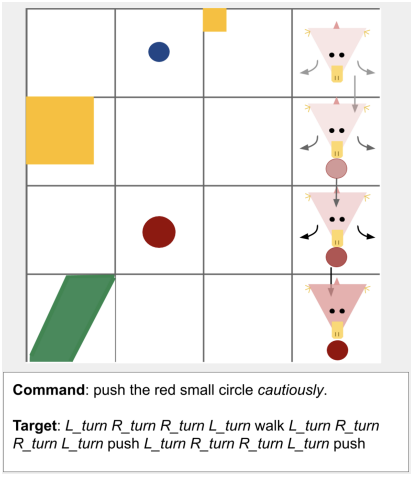
First, the agent needs to match words in natural language with the corresponding places or objects in the images, but the visual grounding of the word could not be on the first image that the agent sees. In this case, the agent needs to navigate the environment to obtain visual information. This is especially important in frameworks like ALFRED and Room2Room [10, 12] since the agent needs to find a specific object in the environment to reach the goal. This means that the model needs to hold for a long time the information of the object specified in the natural language instruction because the object itself won't appear in the visual range until the end of the agent's task. That requires the sequence to sequence model not to forget the instruction information during the activity.

Second, some of the concepts can not be grounded into a specific object and need to be grounded with the context of the object. It is easy to ground the concept of a *horse* with a visual representation of a horse and it's easy to ground the concept of *stripes* with a visual representation of stripes. This also makes it easy to compose concepts to get a visual representation of a new one. For example, if I tell that a *zebra* is a *horse* with *stripes*, I can make a visual representation of a zebra without ever seeing one in my life [7]. The problem comes when it is needed to ground concepts like *big* or *small* (as it happens on gSCAN) with a visual representation. In this case, the visual representation depends on the other things surrounding the corresponding object. This means that VLN needs to sort how to get a level of abstraction that enables the model to understand and ground concepts like *big* or *small* without associating them with specific objects of the visual input.

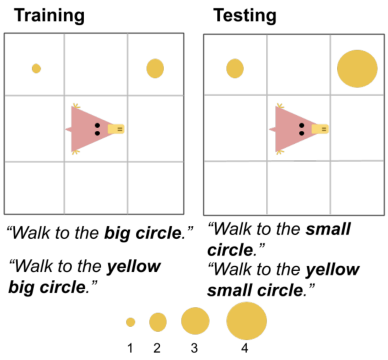
2.2 The Framework: gSCAN

There are several frameworks for the VLN problem like Room2Room with Matterport3D Simulator, The Interactive Navigator-Pointer Model with REVERIE and ALFRED with AI2-THOR 2.0 [1, 4, 10, 12, 8]. This paper focuses on The grounded SCAN (gSCAN) framework [11]. This framework differs from Room2Room, REVERIE, and ALFRED since it works in a 2D environment. It is based on SCAN dataset [9] and also works with GECA [2]. The environment is a 2D grid that has objects with different shapes (circles, squares, and cylinders), colors (red, green, yellow, and blue), and sizes (1, 2, 3, and 4). In this framework, there are two sources of information (instructions and visual) and one output (actions). Each one is listed and described ahead. Figure 1a shows an example of the gSCAN benchmark.

Instructions: The instructions in gSCAN are made with a grammatical rule. This rule allows the instructions to be clear and consistent. Usually, in natural language, there are different ways of phrasing the same instruction, but by using a grammatical rule, gSCAN does not have that level of difficulty. The downside is that the flexibility of the model is reduced and it can only deal with a finite set of instructions and words.



(a) The gSCAN benchmark. The agent must identify the correct target, walk to it 'cautiously' (meanin looking to the left and to the right before moving), and finally push it.



(b) Novel Contextual Reference Test: Generalizing from calling an object "big" to calling it "small".

Figure 1: The gSCAN benchmark.

Visual input: Most of the VLN frameworks work with images, but in gSCAN the visual input is a state matrix of the whole environment that contains the objects' locations and characteristics with one-hot encoding. This also means that the agent can see all the environment at once without the need to move to gather new information.

Actions: The possible actions for the agent are: walk, push, pull, stay, R_turn, L_turn. With this set of instructions the agent not only has to reach a location, find an object, and perform an action on that object but also it has to do it in a certain way (zigzagging, cautiously). Also, bigger objects need to be pushed or pulled twice to move them from one cell to the next one. This adds the concept of *weight*. It is worth mentioning that other frameworks have a grid input with the possible places that the agent can go, but in a 2D world like gSCAN this is not necessary because the places the agent can move are limited by the set of actions. In other words, with the set of instructions available in gSCAN the agent can only move one space up, down, left, or right.

The focus of gSCAN is to understand the grounded language problem in VLN. The goal of the model is to be able to generalize new concepts that were not present at training from other concepts that already knows. There are seven baseline tests on the framework that allows the study of the grounding language problem. This paper focuses on the *Novel Contextual Reference* test. In this test, the model is trained in a scenario where objects of a specific size (size 2) are never targets correctly picked by the *small* modifier. In other words, the object is always the biggest one in the environment and is always referred to as the *big* object. At the moment of testing, the model is presented with a situation where now the object is the smallest one in the environment. The results show that the agent is unable to correctly identify the target object that now is identified by the *small* modifier. This test allows us to know if the model was able to learn the relativity concepts of *big* and *small*. See figure 1b for an example of this situation.

2.3 Curriculum Learning

Curriculum Learning works on the idea that "humans and animals learn much better when the examples are not randomly presented but organized in a meaningful order" [3]. The objective is to start training models by showing them the simpler examples and then gradually add more complex ones. This changes the training paradigm and puts a special emphasis on having a good training strategy. In VLN, Curriculum Learning could be very useful since the instructions, tasks, and environments have significant complexity differences.

Curriculum Learning has been used before in the VLN problem for studding synthetic language acquisition on the BabyAI framework [5]. In this platform the agent has to go through an extensible suite of 19 levels of increasing difficulty. The agent has to acquire new competences through the different levels, facilitating its learning.

3 Problem Definition

The language grounding problem is a real challenge for VLN. It seems that machine learning models cano ground natural language concepts with abstract representations that are tied tightly to the object, but they are not able to ground natural language concepts that are tied to the relationship between different objects. These models can't create an abstraction of a concept that does not have a specific object tied to it. I find it very strange for machine learning models to have this problem. In the case of humans, we don't have a problem differentiating abstract concepts that are tied to an object from the ones that are tied to the environment or the relation between objects [7].

Out of the 7 experiments that were held on gSCAN, the *Novel Contextual Reference* experiment poor results are tied directly to the language grounding problem. As it was previously discussed, at testing time the model is unable to identify an object with the *small* modifier that was always referred to with the *big* modifier during training. Furthermore, it is found that the model is able to select the correct shape and color of the target object (which are properties tied to the object), but at the moment of deciding which size to go with, it picks a random answer (50% chance between two possible objects).

This situation could mean that the model is always trying to ground the concepts with an object and it is not creating a separate abstraction for the *small* and *big* modifiers. I find it odd that the model can't come to a function for distinguishing size relationships between objects, especially in an environment that is simple as gSCAN where all the properties are one-hot encoded. There are a lot of functions that the model could come up with to know which object is bigger: $f(x, y) = x - y$ will tell if $x > y$ with a positive number, and in the one-hot encoding case $f(\vec{x}, \vec{y}) = \vec{W}(\vec{x} - \vec{y})$ will tell the difference (note that these functions are not tied to the inputs, they work with the context information). This means that there is something else that is blocking the model to learn these type of functions that could be used to learn the concepts of *small* and *big*.

I have come with one idea that could improve the performance of the model on the task of learning concepts like *small* and *big*. This idea is related to modifying the training dataset by using new instructions in a strategic way to train the model, and the use of curriculum learning.

3.1 Improving the training dataset

On gSCAN each object has a *shape*, a *color* and a *size*. These three characteristics can be used on the instructions in seven ways, but the dataset only uses them on four: (1) just *shape* ("Walk to the square"), (4) *shape* and *color* ("Walk to the red square"), (5) *shape* and *size* ("Walk to the big square") and (7) *shape*, *color* and *size* ("Walk to the red big square"). The model always has the *shape* of the target inside the instruction. It is possible that the first thing that the model learns is *shape*, and then it learns *color* and *size*. It is also possible that the model ties up the concepts of *color* and *size* to the *shape*, since the *shape* is always present. The model probably can't understand the *size* and *color* without the presence of the *shape*. This could be a problem. For some situations it is not necessary to tell the *shape* in order to correctly identify the target. The only reason for the *shape* to always be on the instruction is to get a grammatically correct sentence.

The dataset could have instructions that are understandable even though they are not grammatically correct: (2) just *color* ("Walk to red"), (3) just *size* ("Walk to big") and (6) *color* and *size* ("Walk to big red"). If the agent needs to walk to a big red square, and that object is the biggest one in the environment, the only information that it needs to correctly identify the target is the *size*. In that case *"walk to big"* is enough even if is not grammatically correct. We as humans sometimes make this kind of grammatical mistakes when we forgot the name of an object and we verbally identify it to other people using other characteristics.

There is a chance for the model to learn the concepts of *size* and *color* without grounding them on the *shape*, making a separate abstraction of what *size* and *color* is. The model could be trained with instructions like 2, 3, and 6 in addition to instructions 1, 4, 5, and 7. At test time, the model should be presented only with instructions 1, 4, 5, and 7, to see if the model improves.

3.2 Adding Curriculum Learning

Furthermore, if we use curriculum learning we can first show the model instructions of type 1, 2, and 3, so it can learn *shape*, *color* and *size* as separate concepts, then add instructions 4, 5, and 6, so it can start relating concepts and finally adding instruction 7, so it can learn to relate the three concepts at the same time.

In short, I hypothesize that the best way to solve the grounding problem in the *Novel Contextual References* test on gSCAN is not by changing the model architecture but by adding better examples to the training set and using curriculum learning.

4 Method

The target object has three characteristics: *shape*, *size* and *color*. Depending on the situation, the target object can be correctly identified using one characteristic only, two characteristics, or three. This gives us seven possible situations, but the original paper uses only four. I created a new expanded dataset using the original dataset used in [11] and adding all the seven possible situations. Table 1 shows a comparison between the old and the new dataset.

	Original Training Set	New Training Set	Shape	Color	Size	Example
(1)	x	x	x			"Walk to the circle"
(2)		x		x		"Walk to the red"
(3)		x			x	"Walk to the big"
(4)	x	x	x	x		"Walk to the red circle"
(5)	x	x	x		x	"Walk to the big circle"
(6)		x		x	x	"Walk to the big red"
(7)	x	x	x	x	x	"Walk to the big red circle"

Table 1: Different types of instructions with 3 possible characteristics.

4.1 Datasets

4.1.1 Training Datasets

As a baseline I will use the same dataset used by gSCAN (I will refer to this dataset as the *original training dataset*). In this dataset, objects of a specific size (size 2) are never targets correctly picked by the *small* modifier. This dataset contains 367.933 examples. Of those examples there are situations where the instruction given to the agent contains: (1) only the *shape*, (4) *shape* and *color*, (5) *shape* and *size* and, (7) *shape*, *color* and *size*.

The *new training dataset* contains 725.391 examples divided into three subsets. The *level 1 subset* contains situations where the target can be correctly identified using only one characteristic: (1) only the *shape*, (2) only the *color* and (3) only the *size*. If the characteristic is shape, let's say that the target is the only square, the instruction will only use the shape to identify the object, for example: "Walk to a square". If the characteristic is color, the instruction will only use the color to identify the object, for example: "Walk to a red". If the characteristic is size, the instruction will only use the size to identify the object, for example: "Walk to a big". Even though the last two examples have grammatically incorrect instructions, those instructions contain all the information needed to correctly identify the target. Hopefully, adding these grammatically incorrect examples will help the model to learn *big* and *small* without tying the concept of relative size to the shape and color of the target object. This subset contains 230.045 examples.

This subset was build by searching for all the situations in the original dataset where the target object could be identified using just one characteristic. If there were not enough examples, I would look for examples where I could delete one object from the situation and make the example work. Then, the redundant words were deleted from the original instruction. For example, if there was an example where an object of size 3 was the biggest one in context and the instruction was "Walk to the big circle cautiously", then the instruction was modified to "Walk to the big cautiously". In this case, this instruction has all the information to reach the target even though is not grammatically correct. Also, the model is forced to learn the relative size concept since it can only rely on the word (big) to correctly identify the target object and it can no longer rely on the shape (circle) anymore.

The *level 2 subset* contains situations where the target can be correctly identified using two characteristics: (4) *shape* and *color*, (5) *shape* and *size* and (6) *color* and *size*. The cases that only use shape and color ("walk to a red circle") and only

shape and size ("walk to a big square") will have grammatically correct instructions, but the case of using color and size ("walk to a red big") will have grammatically incorrect instructions. This subset contains 393.232 examples. Finally, the *level 3 subset* contains situations where the target can be correctly identified using three characteristics: (7) *shape*, *color* and *size*. This subset contains 102.114 examples with only grammatically correct instructions.

Since the *new training set* is almost twice as big as the *original training set* I also created a *short new training set* which contains half of the examples from each subset selected randomly. This will help in the time used for training and it will also be useful to know if the amount of training data affects the result. Is important to notice that in all these datasets there are no examples where the target with size 2 is correctly picked by the *small* modifier.

4.1.2 Test Datasets

To evaluate the experiments I use two test datasets from gSCAN. The first test dataset is the *Relativity Test Set*. This test set contains only examples where objects of a specific size (size 2) are now always targets correctly picked by the *small* modifier. This test set is the same one used in gSCAN to test the "Novel contextual references" experiment to check if the model was able to learn the concepts of "big" and "small". The second test set is the *Random Split Test Set*. This set contains all types of examples. Is important to keep an eye on this test set because we don't want the model to worsen its general performance when enhancing the performance in the specific relative size recognition task. Table 2 contains all the information about the different datasets.

Set	N Examples
Original Training Set	367.933
New Training Set	725.391
Level 1 Subset	230.045
Only Shape	89.340
Only Size	60.195
Only Color	80.510
Level 2 Subset	393.232
Shape & Size	192.393
Shape & Color	153.825
Color & Size	47.014
Level 3 Subset	102.114
Short New Training Set	362.695
Short Level 1 Subset	115.022
Short Level 2 Subset	196.616
Short Level 3 Subset	51.057
Random Split Test Set	19.282
Relativity Test Set	16.808

Table 2: Datasets Information.

4.2 Training

I did five experiments that are listed below. Each experiment is repeated 1 time:

- **Experiment 0:** Consist of recreating the original gSCAN paper experiment. I trained the model on the *Original Training Set* in the same way as is described by [11].
- **Experiment 1:** I trained the same model but now using the *Short New Training Set*. All the examples of the different levels were shuffled.
- **Experiment 2:** I trained the model using the *Short New Training Set* but this time using curriculum learning. At the first stage, I only train using *Level 1* examples. Later, I use *Level 1 & 2* examples. Finally, I use *Level 1, 2 & 3* examples.
- **Experiment 3:** Same as Experiment 1, but with the *New Training Set*.
- **Experiment 4:** Same as Experiment 2, but with the *New Training Set*.

5 Results

The results are shown in table 3. The first thing that is possible to notice by comparing experiment 0 with experiment 1 and 3 is that just adding the new grammatically incorrect examples to the dataset does not improve the performance in the "Relativity Split", but neither it gets worse.

The second thing that it is possible to notice is that curriculum learning plays a key role in improving accuracy when using the new grammatically incorrect examples. We can see this by comparing the results of the "Relativity Split" in experiment 1 with experiment 2 and experiment 3 with experiment 4.

The third thing that is important to notice is that a bigger dataset improves the accuracy in the "Random Split". This can be seen by comparing experiments 1 and 2 with experiments 3 and 4, respectively. In the case of the "Relativity Split", the improvement is only present in the experiments that use curriculum learning.

Finally is possible to see that in the best-case scenario, the new training set and the use of curriculum learning helps to improve the exact match accuracy from 43.18% in the original baseline to 67.10%. That is a 23.92% improvement.

Is important to understand the role of the new grammatically incorrect examples, the curriculum learning techniques, and the size of the dataset. Is interesting to notice that even though the new training examples are essential for the good results of experiments 2 and 4, they can not improve anything by themselves. The new examples need to be used in a certain order with the use of curriculum learning to work for the model. Also, the role of the size of the dataset shows that even though more data improves the general performance, it does not always help in specific tasks performance.

	Random Split Test Set	Relativity Test Set
0: Original Training Set	95.88	43.18
1: Short New Training Set + Shuffled	95.59	46.72
2: Short New Training Set + Curriculum	95.83	62.83
3: New Training Set + Shuffle	**98.40**	46.51
4: New Training Set + Curriculum	97.96	**67.10**

Table 3: Summary of results for each experiment, showing exact match accuracy.

6 Conclusions

The Vision-and-Language Navigation Problem is still a big challenge for machine learning algorithms. Is still complex to combine natural language, visual processing, and sequence generation in one single model that can successfully perform navigation tasks. This is especially valid when VLN faces the Grounding Language Problem. Sometimes, the information given in natural language depends on what is seen in the image and the context plays a key role in giving meaning to the objects and the situation in which they are embedded.

gSCAN is one of the benchmarks that aims to look for systematic generalization in grounded language understanding through seven different tests, but the model performs poorly on some of these tests. This work was able to improve the results in the "Novel Contextual Reference Split", where the challenge is to learn the terms for relative size *big* and *small* when referring to a target object in a given context. This improvement is possible by two modifications in the training stage. First, adding new examples with grammatically incorrect instructions that force the model to learn the *relative size* concepts of *big* and *small* without the ability to rely on other characteristics like *shape* or *color* and, second, using curriculum learning by showing the examples in a particular order so it helps the model to learn better.

From the results, I draw three principal conclusions. First, sometimes the best solutions to the grounding language problem in VLN is not to change the architecture but to rethink how the model is trained, how are the examples made, and the order in which they are shown to the model, helping it to learn the concepts needed to perform the navigation task. Sometimes the training dataset does not need to have the same distribution or the same examples as the test set and, if those differences are well thought, the results at test time and the power of generalization can be improved. Second, although adding new examples to the dataset is essential for the results, these new examples can not give useful information if they are not shown in the correct order. This means that the curriculum learning technique is one of the most powerful tools in machine learning to achieve generalization. Finally, it is possible to conclude that even though a bigger dataset improves general performance in the VLN task, it does not necessarily improve performance in the specific task of learning and generalizing the concepts of *big* and *small*.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

[2] Jacob Andreas. Good-enough compositional data augmentation. *arXiv preprint arXiv:1904.09545*, 2019.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[5] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: First steps towards grounded language learning with a human in the loop. *arXiv preprint arXiv:1810.08272*, 2018.

[6] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2054–2063, 2018.

[7] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.

[8] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[9] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR, 2018.

[10] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991, 2020.

[11] 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.

[12] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10740–10749, 2020.

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.