Refer360° : A Referring Expression Recognition Dataset in 360° Images

Volkan Cirik[1] **Taylor Berg-Kirkpatrick**[2] **Louis-Philippe Morency**[1]
[1]Carnegie Mellon University [2]University of California San Diego
{vcirik,morency}@cs.cmu.edu {tberg}@eng.ucsd.edu

Abstract

We propose a novel large-scale referring expression recognition dataset, Refer360°, consisting of 17,137 instruction sequences and ground-truth actions for completing these instructions in 360° scenes. Refer360° differs from existing related datasets in three ways. First, we propose a more realistic scenario where instructors and the followers have partial, yet dynamic, views of the scene – followers continuously modify their field-of-view (FoV) while interpreting instructions that specify a final target location. Second, instructions to find the target location consist of multiple steps for followers who will start at random FoVs. As a result, intermediate instructions are strongly grounded in object references and followers must identify intermediate FoVs to find the final target location correctly. Third, the target locations are neither restricted to predefined objects nor chosen by annotators; instead, they are distributed *randomly* across scenes. This "point anywhere" approach leads to more linguistically complex instructions, as shown in our analyses. Our examination of the dataset shows that Refer360° manifests linguistically rich phenomena in a language grounding task that poses novel challenges for computational modeling of language, vision, and navigation.

1 Introduction

Imagine a scenario in which you are asked to retrieve medication from a bathroom. 'First, face the sink, then find the second drawer in the cabinet to your left. The pills should be inside that drawer behind the toothbrush." Interpreting instruction sequences in order to locate targets in novel environments is challenging for AI systems (e.g. personal robots and self-driving cars). First, the system needs to ground the instructions into visual perception (Anderson et al., 2018b; Hu et al., 2019). This often requires identification of the mentioned object (Plummer et al., 2015) through physical relationships with surrounding objects (Hu et al., 2017b; Cirik et al., 2018a). Second, since human visual perception has limited field-of-view, instructions are often sequential: First, the correct FoV should be identified before searching for the final target. In many situations, the target location is not visually unique (e.g. in the middle of a plain wall), and several intermediate instructions are required.

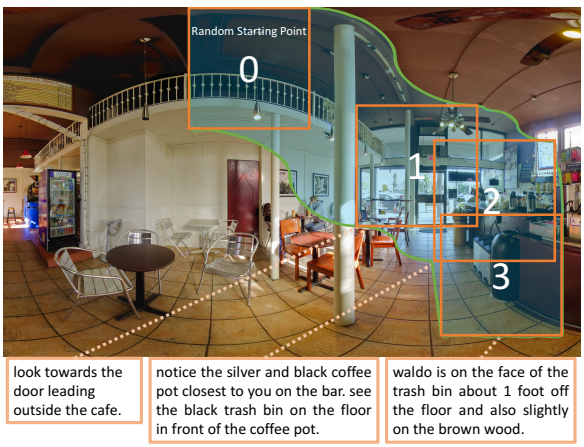
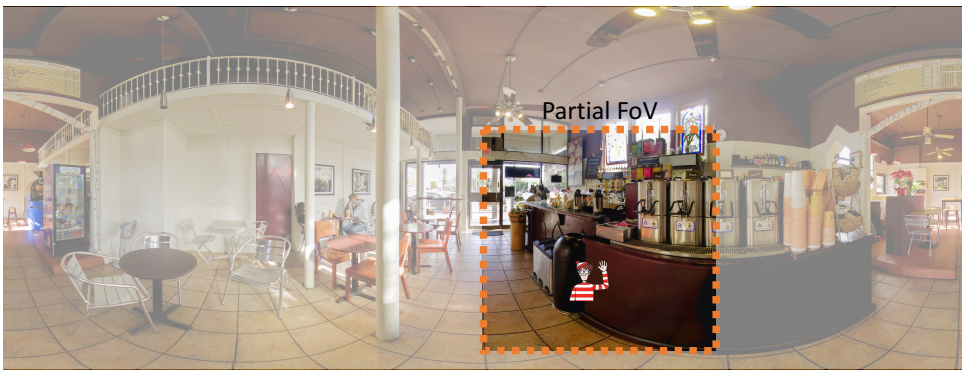


Figure 1: An example from Refer360° . Orange frames represent the field-of-view (FoV) of the follower after interpreting each instruction. Numbers in the frames represent the sequential order. Green lines show how FoVs change continuously. After each instruction, the follower changes the FoV to align with what the instruction describes. Please see Figure 2a to see the correct location of Waldo.

To study these challenges, we introduce a novel dataset, named Refer360°[1], for the task of localizing a target in 360° scenes given a sequence of instructions. Figure 1 presents an example scenario

[1]The annotations, learning simulator, and annotation setup are publicly available for further research https://github.com/volkancirik/refer360.



(a) An example scene from the Refer360° dataset. Note that both annotators and systems cannot observe the shaded area. They only observe a partial field of view which can be updated dynamically.

(b) An example scene from Touchdown-SDR where the bullseye is pointing to the target location. Instructions for this instance are "a black doorway with red brick to the right of it, and green brick to the left of it. it has a light just above the doorway, and on that light is where you will find touchdown."

(c) An example image from Google-Ref dataset with the referring expression "a young elephant nudges its head into that of a slightly taller one.".

Figure 2: Examples are from (a) Refer360° (b) Touchdown-SDR, and (c) Google-Ref datasets. In Refer360° , the target location could be *any random location* on the image. In (b), annotators chose an existing object as the target location. In (c), boxes for objects were used as targets. Refer360° also seeks to increase the complexity of instruction following, making it more realistic by introducing a partial and dynamic FoV rather than providing a holistic oracle-like view of the image.

from Refer360° . For this scenario, finding the target location requires first finding the door leading outside, then looking at the coffee pot, and finally finding the trash can, which is the nearest object to the target. Here, instructions are given from the perspective of a partial field of view (FoV) of the scene, and these FoVs can dynamically be changed. Thus, the correct interpretation of the sequence of instructions will require reasoning about what is currently visible in the FoV (e.g., grounding of objects) but also what is not visible yet. These scenarios will often require adjusting the FoV based on intermediate instructions. An important feature of the Refer360° dataset is that the target location is not an object; instead, it can be any point in the scene, which makes the grounding task more challenging since it is harder to describe a location when we cannot readily refer to it with the name of an object.

Refer360° consists of 17,137 instruction sequences with ground-truth actions to complete these instructions in 360° scenes. Refer360° has some unique characteristics which differentiate it from prior work. First, Refer360° allows the scene to be viewed through a *partial* FoV that can be *dynamically* changed as instructions are followed. This is in contrast with existing 360° scene-based datasets such as Touchdown-SDR (Chen et al., 2018) and 2D image-based referring expression datasets (Kazemzadeh et al., 2014; Hu et al., 2016; Mao et al., 2016), where the visual input is either fixed, corresponding to a holistic, oracle-like view, or consists of fixed, cardinal FoVs. The partial and dynamic FoV in Refer360° poses new challenges for language grounding (see Figure 2a, 2b, and 2c for an illustrative comparison). For instance, the mentioned objects may not be visible in the current FoV, and language may refer to the FoV itself. Further, since our annotators generate instructions while observing a partial and dynamic FoV, and do so for a follower whose first FoV will be initially located at random, the instruction following task is strongly sequential. To interpret the sequence of instructions to find the target correctly, a follower must reason about the sequence of FoVs referenced by the instructor.

Dataset	Target Location Selection	Field of View (FoV)	Action Space	Intermediate Steps
Refer360°	Random Points	Dynamic with partial FoV	4 Directions	✓
Touchdown-SDR (Chen et al., 2018)	Human Selected Points	Oracle: Holistic & Static, 360°scenes	✗	✗
Google-Ref (Mao et al., 2016)	Annotated Objects	2D Images	✗	✗
Ref-UNC (Kazemzadeh et al., 2014)	Annotated Objects	2D Images	✗	✗

Table 1: Comparison of referring expression datasets, including our proposed Refer360° dataset. Refer360° poses a more challenging scenario where the system observes only a partial and dynamic FoV. Refer360° also has includes explicit alignments between intermediate instruction steps and human follower actions which can be used as an auxiliary evaluation metric or source of supervision.

Second, unlike other datasets, the target locations in Refer360° are randomly distributed and thus may occur anywhere – not just on predetermined objects. As a result, target locations are less prone to bias (Devlin et al., 2015; Agrawal et al., 2016; Jabri et al., 2016; Goyal et al., 2016; Cirik et al., 2018b). These random locations lead to more linguistically complex instructions, as shown in our analyses – when instead annotators choose the target location, they are likely to be biased towards locations that are more easily described (e.g. on top of a named object). Table 1 shows a comparison of similar datasets. In the following section, we motivate Refer360° dataset in more detail.

2 Motivation

The vision behind Refer360° is to build systems that perform localization of any point in 3D space, bringing us closer to human-like reasoning. This is an important milestone towards better collaboration between AI systems (e.g. personal robots) and humans, allowing them to act within the same space. It might also pave the way for AI-agents interacting with virtual worlds. The Refer360° dataset was designed to address three technical challenges towards this vision.

First, learning environments we create need to reflect the characteristics of human's perception of 3D space. In such an environment, the agent only observes a partial FoV of the scene. This requires adjusting the FoV in accordance with instructions so that current view and instructions are aligned. The agent's FoV can be changed in a continuous manner, moving smoothly left, right, up, and down. This is analogous to a real-world robot performing motor actions to change its camera position, or a human changing their head's pitch and yaw. Further, real scenes are 3D, but the FoV is represented in 2D in our task. Thus, interpreting some instructions will require inferences about depth.

Second, the paradigm of 360°scenes with partial FoV will almost always necessitate instructions that consist of multiple intermediate steps. As the first intermediate step, the follower and instructor need to find a common referential FoV. Then, the instructor can continue giving guidance towards the target location, often by identifying objects that are physically related to the target location. This multi-step process can serve as a natural benchmark for measuring whether systems achieve localization through a human-like process of progressively getting closer to the target location by interpreting intermediate steps. In other words, this setup may helps researchers make sure that our systems are arriving at the referred location for the right reasons.

Third, since any point in the scene could be of interest, instructions will be more complex: many points in the scene will not correspond to easily named objects, and thus, when such points are allowed as targets, more sophisticated instructions will be required to unambiguously refer to them. The instructor may rely on description of physical relationships with the closest easily named locations in the scene (Nagaraja et al., 2016; Hu et al., 2017b; Cirik et al., 2018b). For instance, in Figure 2a, the target location is on the side of a trash bin, which is difficult to unique describe with a single word or a short phrase. In this case, the instructor may use the distance to the floor or to another object in the scene in order to describe the exact location of the target. This will additionally introduce description of degree (e.g. 'slightly above', 'a few inches away from') rather than more discrete spatial relationships (e.g. 'on top of the desk').

3 Related Work

Referring expression recognition. Grounding a short phrase or a sentence into a visual modality such as video (Khoreva et al., 2018; Anayurt et al., 2019) or imagery (Kong et al., 2014; Plummer et al., 2015, 2018; Yu et al., 2018a) is a well

studied problem in intelligent user interfaces (Chai et al., 2004), human-robot interaction (Fang et al., 2012; Chai et al., 2014; Williams et al., 2016), and situated dialogue (Kennington and Schlangen, 2017). Kazemzadeh et al. (2014), Hu et al. (2017a), and Mao et al. (2016) introduce two benchmark datasets for the real-world 2D images. Nagaraja et al. (2016) propose a model where the target and supporting objects (i.e. objects that are mentioned in order to disambiguate the target object) are identified and scored jointly. Hu et al. (2017b) introduce a compositional approach where they assume that the referring expression can be decomposed into a triplet consisting of the target object, the supporting object, and their spatial relationship. Similarly, Cirik et al. (2018a) propose a type of neural modular network (Andreas et al., 2016) where the grounding of referring expression depends on the parse tree of the input referring expression to learn to ground an unconstrained number of supporting objects.

360° Scenes. Although 360° scenes are well studied in the computer vision domain (Xiao et al., 2012; Su et al., 2016; Wijmans and Furukawa, 2017; Yang and Zhang, 2016; Xu et al., 2018; Yang et al., 2018; Yu et al., 2018b), few studies explore the challenges of 360°scenes in the context of language grounding. Chou et al. (2018) introduce a dataset where 360° videos are narrated. They address the task of predicting the field of view for the given narration. Anderson et al. (2018b) introduce the vision and language navigation task for simulated indoor environments where an agent is placed in a location in a house and follows the instructions to go to a target location. Here the agent observes a discretized view of the current location (i.e. the 360° scene is split into a fixed number of field of views). The most related work to Refer360° is Touchdown (Chen et al., 2018) which introduces two tasks: a vision and language navigation task and a spatial description resolution (SDR) task (i.e. a referring expression recognition task for a simulated outdoor environment). In contrast with Touchdown, in our setup instructors, followers, and learning systems observe a partial FoV of the scene, but they can change the FoV continuously to explore the scene. This approach yields instructions with a stronger sequential dependencies and with stronger reference to the FoV itself. We demonstrate some of these differences in analysis in Section 5. Concurrent work studies visual question answering (Chou et al., 2020a) and object detection (Chou et al., 2020b) for 360°scenes. Another concurrent study (Qi et al., 2020) combines vision-and-language navigation and referring expression recognition into one task where the system is asked to localize the referred object after navigating to another point in a real images of rendered buildings.

4 Refer360° Dataset

In this section, we describe the details of the Refer360° dataset, a vision-and-language benchmark for localizing a target point in a panoramic image. Refer360° consists of 17,137 instruction sequences that describe randomly distributed target locations in 2,000 panoramic scenes from the SUN360 (Xiao et al., 2012) dataset. We first explain the annotation procedure for collecting and validating the instruction sequences. Later, we discuss the statistics of the Refer360° dataset.

4.1 Annotation Procedure

Annotation of the Refer360° dataset was carried out in three stages on Amazon Mechanical Turk with two tasks, namely a description task and a finding task. First we describe the two tasks in more detail.

Description Task. Our main goal is to collect instructions for finding any point in a 360° image. Annotators started this task looking at the ceiling of the 360° image with a random yaw[2]. We asked them to find the target location for which we use an icon of Waldo[3]. Target locations are choosen randomly – we discuss the details of this design choice in Section 4.2. The target can be at any longitude and can have a latitude within a range of 45 degrees from the top and bottom of the 360 image. This restriction in latitude is made for two reasons: (1) visual distortions happen at extreme points, and (2) during the finding task, the starting point is the "ceiling" of the 360 image. Annotators were asked to give instructions to find the target location using at least three instructions[4].

Finding Task. We design this task to verify the quality of instruction sequences provided by anno-

[2]We wanted to avoid introducing any bias by beginning the same position each time for each scene.

[3]https://en.wikipedia.org/wiki/Where%27s_Wally%3F

[4]Please see Figure 5 in Appendix to see a screenshot of the user interface we build for this task.

tators in the description task. We asked annotators to complete the instruction sequences sentence by sentence. The initial field of view of annotators is always pointing at the ceiling of the 360° image with a random yaw. We asked annotators to change the FoV after each instruction so that the center of the FoV points to the location the intermediate instruction is describing. After moving the FoV to the correct position, annotators clicked a button to read the next instruction. We recorded the spherical coordinates of the center of the FoV after each instruction. As a result, our annotations include aligned intermediate steps that find the target location. After the final instruction, the annotators predicted the target location by changing the center of FoV or clicking on the FoV.

We collected and verified the quality of our data in three stages using description and finding tasks. In the first stage, we sought a pool of annotators providing high-quality annotations. For the second, aimed to collect a large number of annotations and verify their quality. In the third stage, we further verified instruction sequences that were not verified in the second stage.

Stage I. In this stage, we asked annotators to complete the finding task for four different scenes. We wrote the instruction sequences for this stage's finding task to give annotators an example of instruction sequences for describing the target location. Then, annotators completed the description task for 4 different scenes. A total of 256 annotators participated in this first stage. We manually inspected each instruction sequences provided by these annotators for their quality of descriptions of the target location and reduced the pool of annotators to 86.

Stage II. In this stage, for each annotation session, we asked annotators first to find the target location for four different scenes, and later, describe the target location four times for different scenes[5]. We used the finding task to verify the quality of the instruction sequences. If an annotator predicts the target location within a radius of 11 degrees in spherical coordinates, which is roughly equal to the size of the Waldo icon we used, we counted that instance as verified.

Stage III. After the second stage, we have some instructions where the annotators could not find the target accurately. This could mean either the instructions are not clear, or it is actually harder to find the target location with these instruction sequences. In the third stage, we did another round of the finding tasks to verify these harder instruction sequences.

After these three stages, we have a total of 17,137 instruction sequences in which at least one annotator was able to find the target location accurately. Statistics for data collection in these stages and the payment structure is in the Appendix.

[5]Annotators never observed their own instruction sequences while doing the finding tasks.

Scene Type	Scene Location	# of Images
Restaurant	Indoor	500
Shop	Indoor	250
Expo Showroom	Indoor	250
Living Room	Indoor	250
Bedroom	Indoor	250
Street	Outdoor	250
Plaza Courtyard	Outdoor	250

Table 2: Statistics for Panoramic Images used in Refer360° dataset.

4.2 Dataset Statistics

We split our presentation of dataset statistics into two parts: namely, scene statistics and language statistics.

Scene Statistics: To investigate the challenges in localizing a target location for both indoor and outdoor scenes as well as for different kinds of indoor and outdoor scene categories, we use seven scene categories from the SUN360 (Xiao et al., 2012) dataset. We use total of 2,000 scenes. Table 2 shows the distribution of scene categories that comprise the Refer360° dataset.

We want to analyze the richness of the scenes in the Refer360° dataset and compare it with Touchdown-SDR. The domain of the scenes will affect the instruction one needs to use to describe a target location. To be more specific, when annotators give instructions, they use supporting objects as anchor points to help guide the attention of the follower. Thus, the availability of a rich set of objects is essential for describing the target location. Since the annotation of objects in 360° images is a laborious task itself, we use an off-the-shelf object detection method (Anderson et al., 2018a) to annotate scenes with objects. We split 360° images into 12 different 2D images covering the 360° view[6]. This provides us a proxy to analyze the

[6]We fixed the confidence threshold for detection of objects

kind of objects usually observed in 360° images in Touchdown-SDR and Refer360° .

Dataset	Avg # of Objects	Object Type PPL
Touchdown-SDR	93.81	15.93
Refer360°	62.44	42.93

Table 3: Statistics for detected objects per image in Touchdown-SDR and Refer360° . On average, Refer360° images contain fewere of objects. However, these objects are from a wider variety of object types.

Table 3 shows the average number of objects and the perplexity of the distribution of detected objects per 360° scene used in Refer360° and Touchdown-SDR datasets.[7] As expected, the average number of detected objects in Touchdown-SDR scenes is higher than in Refer360° because all scenes depict outdoor settings from Google's StreetView API. However, this analysis shows that Refer360° has much larger diversity of object types and therefore will likely have greater lexical diversity in instructions.

Scenes	Dataset	Avg. Text Length	Vocab. Size	Size
360°	Refer360°	43.80	11220	17,137
360°	Touchdown-SDR	26.97	5705	9325
2D	Guess What?!	24.99	27713	160745
2D	Google-Ref	8.46	12108	142210
2D	Refer-UNC	3.51	21305	414138

Table 4: Language statistics for Refer360° dataset and other referring expression recognition datasets.

Language Statistics: Refer360° contains a total of 17,137 instruction sequences (8.57 per scene) describing target locations. Table 4 shows language statistics for Refer360° and other referring expression recognition datasets. Refer360° is bigger than Touchdown-SDR, yet, smaller than other datasets. This is because it is a more time-intensive and costly process to annotate and validate 360° images compared with 2D images.

Figure 3 shows the distribution of text length for the instructions. Compared to other referring expression recognition and image captioning datasets, Refer360° contains the longest instructions on average. This is a result of two differences with previous tasks. First, previous datasets use the entire scene as a single field of view. Thus, there is reduced need to describe how to find the target location sequentially. In Touchdown-SDR, the recognition system or human annotator needs to find an FoV that includes the target location. In Refer360° , the finding task is carried out sequentially; thus, each instruction needs to be completed accurately to be able to find the target location. Second, in Refer360° , the target location is randomly distributed in scenes. As seen in Table 6, when the target location is randomly selected, the target location is on average further from other objects (we discuss this in more detail in Section 5.1).

to 0.5 and maximum number of objects to 20.

[7] In the appendix, Figure 6 shows the most detected objects for Refer360° and Touchdown-SDR datasets.

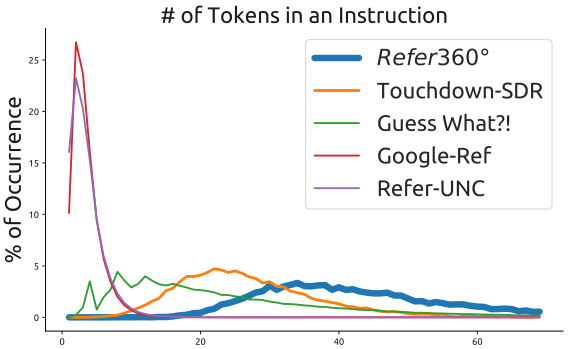


Figure 3: Distribution of the number of tokens for vision-and-language datasets similar to Refer360°

Split	# of Instances
Train	13287
Validation Seen	900
Validation Unseen	1009
Test Seen	900
Test Unseen	1041

Table 5: Statistics for dataset splits in Refer360° dataset.

Dataset Splits: We use a similar train, validation, and test split strategy as the Room-to-Room dataset (Anderson et al., 2018b). We reserve a subset of images from each scene category for validation and test splits for unseen scene evaluation i.e. these scenes are not observed in the train split to study generalization capabilities of models. The remaining scenes are pooled together for training, validation, and test splits for seen scenes evaluation. Table 5 shows statistics for the splits. Following the previous studies, the ground-truth annotations for test splits will not be released. Instead, we will provide an evaluation server where model predictions may be uploaded for scoring.

5 Analyses

We conduct four analyses of the Refer360° dataset. First, we investigate if the random selection process of target locations can mitigate possible bias issues. Recent studies (Devlin et al., 2015; Agrawal et al., 2016; Jabri et al., 2016; Goyal et al., 2016) show that design decisions for collecting annotations may introduce bias into datasets. High-capacity machine learning models can exploit these issues which hinders the meaningful progress towards real language understanding (Zhou et al., 2015; Cirik et al., 2018b). Second, we study whether each instruction in an instruction sequence is critical in finding the target location, or whether some instruction sequences are overcomplete. It may be very well the case that, by just understanding the last instruction, one can easily locate the target location. Third, we perform a qualitative analysis of Refer360° to provide the types of linguistic reasoning required to find the target location accurately. Finally, we analyze the performance of the state-of-the-art on Refer360° .

5.1 Target Locations

The selection method for the target location plays a crucial role in the kind of language one needs to use to describe that location. Earlier studies on referring expression recognition datasets (Kazemzadeh et al., 2014; Hu et al., 2016; Mao et al., 2016; Strub et al., 2017) select the target location as object boxes annotated by humans. In Touchdown-SDR (Chen et al., 2018) instead, annotators decide the location of the target rather than choosing one of the pre-defined lists of object boxes[8].

This could introduce a location bias to the dataset – i.e. if annotators get to select the target location, they may choose targets that are easy to describe, sometimes leading to trivial or uninteresting examples, and more broadly to artificially simple language overall. For instance, if there is only one pink object in the scene, annotators usually preferred describing that region rather than some other obscure location in the scene. Instead of letting annotators decide where to place targets in the scene, we *randomly* picked a target location in the scene and asked them to describe how to find that location. As a result, our instruction sequences are complex as we show next.

[8]In our initial iterations for the data collection, we followed this procedure. However, we observed that in many cases, annotators chose the most salient, or unique object or region in the image. Figure 7 in the appendix compares the distribution of instruction sequence lengths for random and manual selection of targets.

Comparison	Touchdown-SDR	Refer360°
The perplexity of the distribution of an object that the target is located on	9.53	17.86
The perplexity of the distribution of the closest objects	17.80	46.84
The average distance to the closest objects	8.64	23.88

Table 6: Statistics for target locations image in Touchdown-SDR and Refer360° . Target is located on or near the wider variety of objects and further away from other objects.

To measure the differences in instructions for randomly or manually choosen targets, we compute three quantities. First, we compute the variety of objects that the target is located on using the perplexity of object frequencies. Similarly, we also compute the variety of objects closest to the target objects. Since we use objects near to the target location as anchor points, this is also another useful metric. The higher the perplexity of both metrics, the harder it is to predict the target location using just the object type or the closest object. Third, we measure the average distance between the target location and the nearest object. The closer the target location to another object, the easier it is to describe using the closest object as an anchor point.

Instructions	Average Distance	Accuracy
Last Sentence	73.01	0.37
Last 2 Sentences	42.32	0.63
All Sentences	11.35	0.88

Table 7: Results for instruction ablation human study. Annotators need all instructions to complete the task accurately.

Table 6 shows statistics for target locations in Touchdown-SDR and Refer360° . For both perplexity metrics, we observe that the target is located near or inside a wider variety of objects in Refer360° . Also, on average, the target location is further away from other objects for Refer360° . These statistics show that randomly choosing the target location helps us address possibly bias towards simple instructions and makes recognition more challenging.

5.2 Ablation of Instruction Sentences

While collecting instructions, we asked annotators to describe the target location using at least three and at most five sentences. It might be possible to

Phenomenon	c	μ	Example from Refer360°
Coreference	96	1.6	*on the very upper left corner of the blue part of* ***that window***
Comparison	15	0.1	*the* ***smaller*** *building to the right of the spire*
Sequencing	13	0.1	*go right just a smidge and* ***then*** *go up above*
Counting	30	0.3	*shaped like a football and has* ***3 silver legs***
Allocentric Spatial Mention	46	0.6	*find the shelves with books nearest to* ***you***
Egocentric Spatial Mention	35	0.5	*waldo is sitting on the* ***right side of the window***
Direction	92	1.6	*look at the knife on the wall to* ***the left***
Temporal Condition	13	0.1	*turn right* ***until*** *you see a mirror on the wall*
3D understanding	22	0.2	*counter with the two bar stools sitting* ***in front of*** *it*
Inexact/Approximate Language	28	0.2	*in front of the white strip at the bottom* ***slightly off*** *center*
More than 2 Supporting Objects	47	0.5	*now look on* ***the floor*** *in between* ***the table*** *and* ***the chair***

Table 8: Linguistic analysis of 100 randomly sampled examples from Refer360° . We annotate each example for the presence and count of each phenomenon. c is the total number of instructions out of the 100 containing at least one example of the phenomenon. μ is the mean number of times each phenomenon appears per instruction sequence.

find the target location using only the last instruction, which may make the first sentences unnecessary. Such redundancy makes it harder to study the core challenges of grounding instructions to visual perception and actions. Thus, we conducted an ablation study with the same pool of annotators using 1K instructions from the dataset. Here we check whether Refer360° has strong dependencies between instructions.

We ran two ablation studies to examine the necessity of using all instruction sentences. For the first study, we ran a finding task with the same pool of annotators, where we provided only the final instruction. For the second study, similarly, we ran another finding task where we provided only the penultimate and the final instruction. We compare the average euclidean distance between the predicted locations and the target location, and the accuracy, i.e. for what percentage of the time the distance between the predicted location and the target location is less than 11 degrees.

Table 7 shows the result of our ablation analysis. Annotators' performance significantly dropped when they can only read the last instruction. They could find the target object only 37% of the time. Using the penultimate instruction helped them a lot, and they achieved 63% accuracy. The best performance is achieved when they observe the full instructions. These results show that each instruction is necessary for accurately finding the target location.

5.3 Linguistic Phenomena Observed

Before designing a system to address a language-related task, it is important the understand different kinds of linguistic phenomena observed in the task. We follow the procedure described in Touchdown-SDR (Chen et al., 2018), and added a few novel phenomena including 3D understanding, inexact language, and the use of more than two supporting objects as linguistic phenomena. Table 8 shows the result of our analyses for 100 randomly sampled instances. Refer360° requires reasoning for a rich set of linguistic phenomenon including the resolution of the coreference chains, counting objects, a rich set of spatial language phenomena such as multiple-supporting object mentions and 3D scene understanding.

5.4 Localization Experiments

Our analyses in the previous subsections suggest that Refer360° poses several challenges. In Section 5.1, we show that since the target locations are randomly chosen, it is harder to exploit possible location bias. In Section 5.2, we show that it is essential to model the sequential nature of the instructions. Section 5.3 shows that there are lots of interesting linguistic phenomena observed in Refer360° . We want to verify these claims by training the state-of-the-art model and measure its performance on our Refer360° dataset.

We use the same experimental setup in Touchdown-SDR using the scenes provided in the concurrent work (Mehta et al., 2020), where we slice 360scene into 8 FoVs covering the scene. We

pass each of these FoVs to a pre-trained model (He et al., 2016), and extract features from fourth to the last layer before classification to get a feature map representation of the FoVs. We concatenate 8 FoV slices to a single tensor to represent the 360° scene.

We use the LingUNet model (Chen et al., 2018; Misra et al., 2018; Blukis et al., 2018), which performs the state-of-the-art results on TouchDown-SDR dataset. LingUNet is an image-to-image encoder-decoder model where a language and image representations are fused to predict a probability over the input image. Instructions are fed to bi-directional Long Short-Term Memory (LSTM) recurrent neural network to induce a language representation. To induce fused image-text representations, the input image tensor is passed to a convolutional neural network conditioned on the test representations. The fused representation is then fed to deconvolution layers to predict the location of the target. We use the same accuracy and distance metrics described in Section 5.2.

Dataset	Accuracy (%)	Distance
Touchdown-SDR (reported)	26.1	708
Touchdown-SDR (replication)	23.5	715
Refer360°	13.0	1235

Table 9: Results for the LingUNet on two benchmark datasets. Since LinGUNet designed for observing the full instruction set and the holistic view of the scene, and it performs significantly worse on Refer360° .

As we can see in Table 9, LingUNet performs significantly worse on Refer360° [9]. This might be due to the difference we highlighted in earlier sections. First and foremost, instructions must be completed sequentially. However, LingUNet does not model the sequential nature of the task for Refer360° , rather uses all instruction sequence and oracle-view of the 360° scene. Second, the scenes in Touchdown-SDR is from a single domain, but in Refer360° , we have a richer set of scenes for both indoor and outdoor.

6 Conclusion

We designed Refer360° to study 3D spatial language understanding for real scenes. We collected a fine-grained set of annotations that support study at many levels of language grounding. Refer360° is a versatile dataset and enables investigation along three axes:

- **Language:** Refer360° enables modeling tasks that study single instruction, multiple instructions, or interactive language where the next instruction is revealed only after reaching an intermediate milestone.

- **Vision:** Refer360° enables modeling tasks that try to predict targets at different granularities: at the object level if trying to identify the closest object to the target, at the region level in a similar style to Touchdown-SDR, and finally, at the pixel level.

- **Action:** Refer360° enables modeling tasks where the action space is static with the whole 360 image given upfront, where the action space consists of a sequence of discrete choices between fixed views, and when the action space is continuous, consisting of angles for rotation.

In our experiments, we presented one of these scenarios (single instruction, static, and pixel-level) since it was the closest to the pre-existing Touchdown-SDR system. However, one can also study a much larger number of scenarios and modeling tasks using Refer360° .

[9]We used publicly available code provided by authors to run the experiments. We could not replicate the exact numbers reported in the paper, yet, we use exactly the same setup for both Refer360° and Touchdown-SDR for a fair comparison.

Acknowledgments

We are thankful to anonymous ACL conference reviewers for providing valuable feedback. We thank members of MultiComp Lab at CMU and Berg Lab at UCSD for useful discussions. We thank Howard Chen for helping us replicate their experiments. This material is partially supported by Siemens and the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Siemens or the National Science Foundation, and no official endorsement should be inferred.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960. Association for Computational Linguistics.

Hazan Anayurt, Sezai Artun Ozyegin, Ulfet Cetin, Utku Aktas, and Sinan Kalkan. 2019. Searching for ambiguous objects in videos using relational referring expressions. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018a. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018b. 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.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48.

Valts Blukis, Dipendra Misra, Ross A Knepper, and Yoav Artzi. 2018. Mapping navigation instructions to continuous control actions with position-visitation prediction. *arXiv preprint arXiv:1811.04179*.

Joyce Y Chai, Pengyu Hong, and Michelle X Zhou. 2004. A probabilistic approach to reference resolution in multimodal user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 70–77. ACM.

Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. 2014. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 33–40. ACM.

Howard Chen, Alane Shur, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2018. Touchdown: Natural language navigation and spatial reasoning in visual street environments. *arXiv preprint arXiv:1811.12354*.

Shih-Han Chou, Wei-Lun Chao, Wei-Sheng Lai, Min Sun, and Ming-Hsuan Yang. 2020a. Visual question answering on 360° images. In *The IEEE Winter Conference on Applications of Computer Vision*.

Shih-Han Chou, Yi-Chun Chen, Kuo-Hao Zeng, Hou-Ning Hu, Jianlong Fu, and Min Sun. 2018. Self-view grounding given a narrated 360 video. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shih-Han Chou, Cheng Sun, Wen-Yen Chang, Wan-Ting Hsu, Min Sun, and Jianlong Fu. 2020b. 360-indoor: Towards learning real-world objects in 360° indoor equirectangular images. In *The IEEE Winter Conference on Applications of Computer Vision*.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. 2018a. Using syntax to ground referring expressions in natural images. In *AAAI*, volume to appear.

Volkan Cirik, Louis-Philippe Morency, and Taylor Berg-Kirkpatrick. 2018b. Visual referring expression recognition: What do systems actually learn? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 781–787. Association for Computational Linguistics.

Jacob Devlin, Saurabh Gupta, Ross Girshick, Margaret Mitchell, and C Lawrence Zitnick. 2015. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*.

Rui Fang, Changsong Liu, and Joyce Yue Chai. 2012. Integrating word acquisition and referential grounding towards physical world interaction. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 109–116. ACM.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *arXiv preprint arXiv:1612.00837*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017a. Learning to reason: End-to-end module networks for visual question answering. *arXiv preprint arXiv:1704.05526*.

Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. 2019. Are you looking? grounding to multiple modalities in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6551–6557, Florence, Italy. Association for Computational Linguistics.

Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2017b. Modeling relationships in referential expressions with compositional modular networks.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564.

Allan Jabri, Armand Joulin, and Laurens van der Maaten. 2016. Revisiting visual question answering baselines. In *European conference on computer vision*, pages 727–739. Springer.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. 2014. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*.

Casey Kennington and David Schlangen. 2017. A simple generative model of incremental reference resolution for situated dialogue. *Computer Speech & Language*, 41:43–67.

Anna Khoreva, Anna Rohrbach, and Bernt Schiele. 2018. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision*, pages 123–141. Springer.

Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. 2014. What are you talking about? text-to-image coreference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3558–3565.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20.

Harsh Mehta, Yoav Artzi, Jason Baldridge, Eugene Ie, and Piotr Mirowski. 2020. Retouchdown: Adding touchdown to streetlearn as a shareable resource for language grounding tasks in street view. *arXiv preprint arXiv:2001.03671*.

Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. 2018. Mapping instructions to actions in 3d environments with visual goal prediction. *arXiv preprint arXiv:1809.00786*.

Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. 2016. Modeling context between objects for referring expression understanding. In *ECCV*.

Bryan A. Plummer, Kevin J. Shih, Yichen Li, Ke Xu, Svetlana Lazebnik, Stan Sclaroff, and Kate Saenko. 2018. Revisiting image-language networks for open-ended phrase detection. *arXiv:1811.07212*.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Yuanka Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Florian Strub, Harm de Vries, Jeremie Mary, Bilal Piot, Aaron C. Courville, and Olivier Pietquin. 2017. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Yu-Chuan Su, Dinesh Jayaraman, and Kristen Grauman. 2016. Pano2vid: Automatic cinematography for watching 360° videos. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

Erik Wijmans and Yasutaka Furukawa. 2017. Exploiting 2d floorplan for building-scale panorama rgbd alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 308–316.

Tom Williams, Saurav Acharya, Stephanie Schreitter, and Matthias Scheutz. 2016. Situated open world reference resolution for human-robot dialogue. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, pages 311–318. IEEE Press.

Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2012. Recognizing scene viewpoint using panoramic place representation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2695–2702. IEEE.

Yanyu Xu, Yanbing Dong, Junru Wu, Zhengzhong Sun, Zhiru Shi, Jingyi Yu, and Shenghua Gao. 2018. Gaze prediction in dynamic 360 immersive videos. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5333–5342.

Hao Yang and Hui Zhang. 2016. Efficient 3d room shape recovery from a single panorama. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5422–5430.

Yang Yang, Shi Jin, Ruiyang Liu, Sing Bing Kang, and Jingyi Yu. 2018. Automatic 3d indoor scene modeling from single panorama. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3926–3934.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018a. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*.

Youngjae Yu, Sangho Lee, Joonil Na, Jaeyun Kang, and Gunhee Kim. 2018b. A deep ranking model for spatio-temporal highlight detection from a 360○ video. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2015. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*.

A Appendix

This section presents details omitted in the main document. It includes the details about the annotation task, screenshots for the MTurk interface for annotation tasks, the most detected objects in Refer360° and Touchdown-SDR, and text length for instructions collected with different methods.

A.1 Payment and Incentive Structure

One session of annotation consisted of finding task for 4 scenes and describing task for 4 scenes which took about 15 minutes to complete on average. The base pay for one session was $2.25. For each instruction sequence that was accurately found by another annotator, we paid a bonus of $0.10 to both the annotator who found the location and the annotator who wrote the instruction sequence. Thus, for both the finding and describing task annotators have an interest in performing the task accurately. Next, we provide statistics of the Refer360° dataset.

Annotation Stage	# of Annotators	# of Collected Instructions	# of Verified Instructions
Stage I: Hiring	256	854	n\a
Stage II: Collection & Verification	86	20630	14062
Stage III: Verification	86	n\a	3073

Table 10: Statistics for data collection stages. Stage I is for hiring annotators. Stage II is for collecting and verifying the instructions. Last stage is further verifying hard instances that are not verified II.

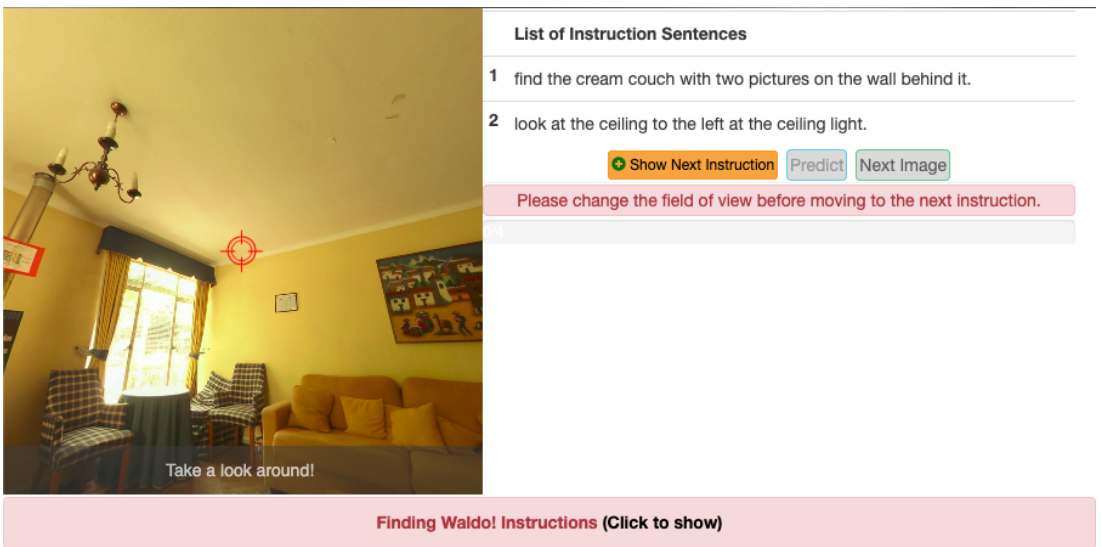


Figure 4: Screenshot of Amazon Mechanical Turk interface for finding task. We ask annotators to complete each instruction before moving to the next one. To do so change the bullseye where they think the instruction is describing.

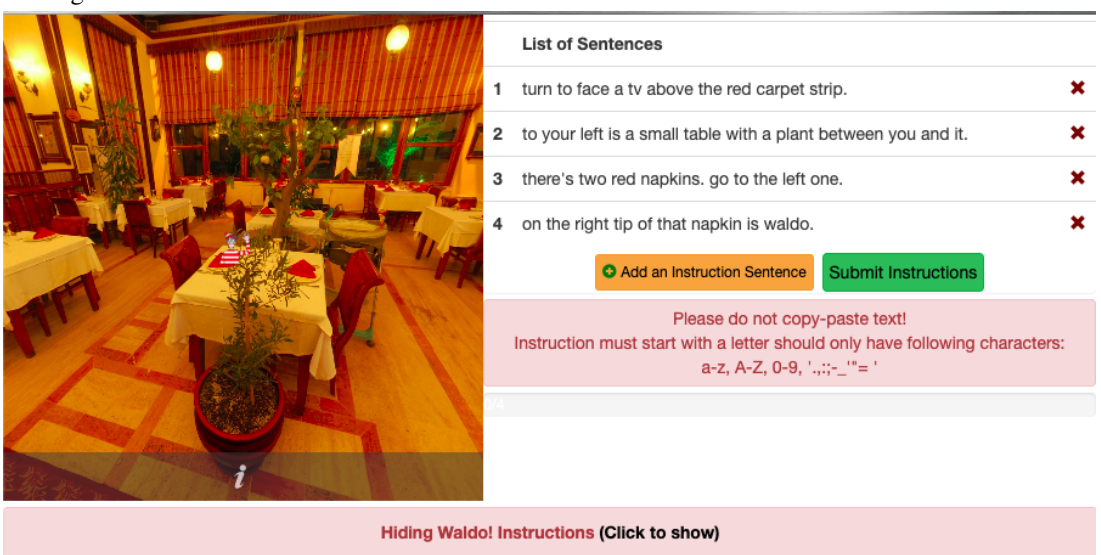


Figure 5: Screenshot of Amazon Mechanical Turk interface for describing task. We ask annotators to first find Waldo themselves, then give detailed insturctions one by one so that anyone starting from a random field-of-view find it.

Figure 6: The most frequently detected objects in Touchdown-SDR and Refer360° .

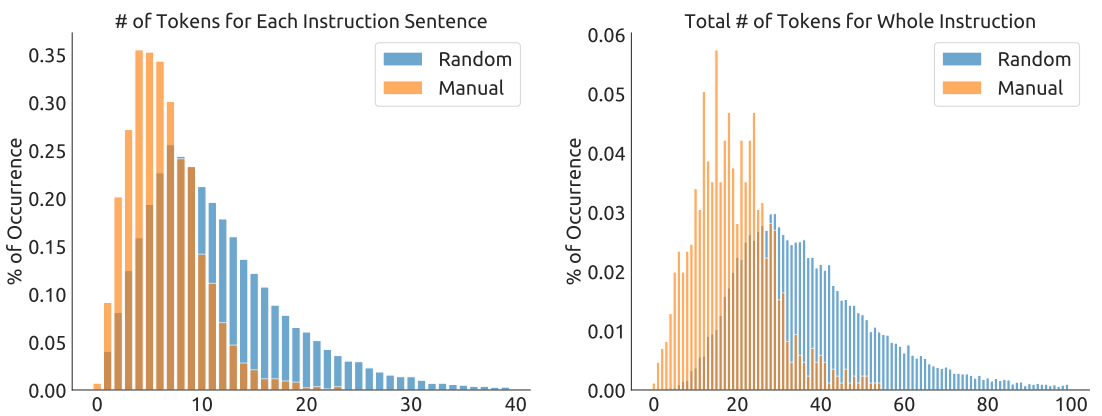


Figure 7: Text length for different placement methods for single instruction and instruction sequences. Manual means annotators pick the target location, random means we randomly pick the target location in the scene.