Natural Language Instructions for Human-Robot Collaborative Manipulation

Rosario Scalise*, Shen Li*, Henny Admoni, Stephanie Rosenthal, and Siddhartha S. Srinivasa

Abstract

This paper presents a dataset of natural language instructions for object reference in manipulation scenarios. It is comprised of 1582 individual written instructions which were collected via online crowdsourcing. Each of these instructions was elicited from one of 28 included scenario images. This dataset is particularly useful for researchers who work in natural language processing, human-robot interaction, and robotic manipulation. In addition to serving as a rich corpus of domain-specific language, it provides a benchmark of image/instruction pairs to be used in system evaluations as well as uncovers inherent challenges in tabletop object specification. Example code is provided for easy access via Python.

Keywords

Natural Language, Instructions, Human-Robot Collaboration, Manipulation, Ambiguity, Perspective, Spatial Reference

Introduction

In this paper, we present a corpus of natural language instructions used to specify objects of interest, or target objects, in a collaborative tabletop manipulation setting. Understanding and generating natural language is essential for fluid human-robot collaboration (Scheutz et al. 2006). We are particularly interested in situations where people need to set apart the target object from many visually similar objects, or distractors, and verbally make reference to it. To this end, we provide a dataset of human generated instructions specifying tabletop objects in synthesized tabletop manipulation scenarios, virtual scenes containing visually similar clutter. Fig. 1 shows an example of one such scenario.

These scenarios embody the challenges inherent in object reference. Clutter is pervasive in environments found throughout our daily lives (Berenson and Srinivasa 2008) and can easily form sets of distractors, or objects that make specifying a goal more difficult. Consider tasks such as table place-setting or object retrieval from a storage room of haphazardly organized boxes. When people intend to single out an object from the rest and refer to it during both instruction comprehension and generation, they can fail due to a range of ambiguities that arise from the presence of distractors. Ambiguities like object similarity (the lack of visually distinguishing features between objects), object proximity (the spatial closeness between objects), and perspective (the establishment of a fixed visual frame of reference) can lead to the emergence of multiple candidates an instruction might refer. Accordingly, people utilize a multitude of strategies and sources of information to reduce the ambiguities (Keysar et al. 2000). Identifying these strategies is important to researchers in the fields of both Robotics and Natural Language Processing (NLP).
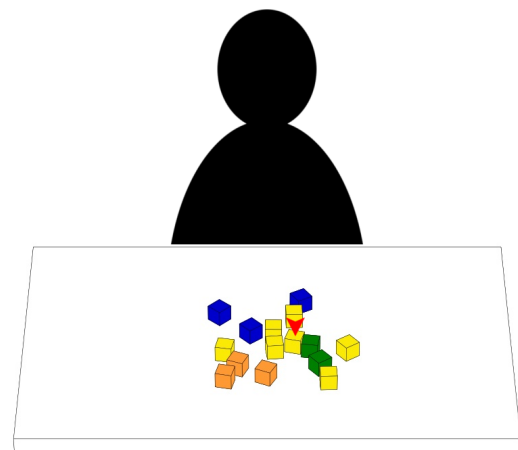

Figure 1. Example of a block configuration stimulus image, or scenario, (Image 13, Configuration Version 2). The block with a red arrow is the target block. An example of a corresponding instruction: "Pick up the yellow block which is to the right of the green block closest to you."

We provide researchers with a dataset from which these natural language strategies can be extracted. We include the images used to elicit the instructions (variants of Fig. 1) and the resulting data of human-generated typed natural language instructions describing a target object in those images. Instructions were generated by human participants

Personal Robotics Laboratory, Robotics Institute, Carnegie Mellon University, USA

Corresponding author:
Rosario Scalise & Shen Li, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213, USA
Email: {rscalise,shenl1}@andrew.cmu.edu

via Amazon's Mechanical Turk (AMT)[1]. Data coders labeled each instruction with the number of objects it could refer to and the type of perspective present in each instruction. We also provide a supplementary dataset that assesses the clarity of each human-generated instruction. We achieved this by showing each instruction to a new set of AMT participants and measuring their accuracy in selecting the correct target object for each of the original instruction/image pairs.

Researchers interested in utilizing natural language commands to perform collaborative manipulation tasks (particularly in cluttered environments) should find the presented corpus valuable in establishing a baseline of unconstrained human performance when describing objects on a tabletop subject to ambiguity. NLP researchers can utilize this as a corpus of referring expressions as well as a corpus dense with instructional language. Applications where unambiguous natural language is critical include providing robotic accessibility in the home to people with mobility disabilities, enabling mutual understanding via language in tasks requiring human-robot collaboration such as a furniture assembly, and interfacing with an industrial pick-and-place robots (such as circuit board assemblers) where discrete components can differ only slightly in appearance.

License

This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY-SA 4.0) license (Creative Commons Corporation, 2016). The Massachusetts Institute of Technology (MIT) license is used for the access and example code provided.

Related Work

In Natural Language Generation (NLG), referring expressions are used by people to specify an entity to an addressee. For example, a person who intends to refer to a specific man will say "the man in a suit" to her addressee, so that the addressee can use attributes "man" and "suit" from the expression to successfully find the referred entity. Researchers have been developing corpus-based algorithms to generate referring expressions and collecting corpora for evaluation (Krahmer and Van Deemter 2012).

General-purpose corpora have the potential to serve as resources in work elucidating referring expressions. "Pear Story of Chafe" is a corpus of narratives which has been used to study anaphora such as "he" referring to "the man" (Krahmer and Van Deemter 2012). Map Task corpus (Anderson et al. 1991) and iMap corpus (Guhe and Bard 2008) contain dialogues in navigational tasks and have been used to study initial and subsequent references (Viethen et al. 2010). Coconut corpus (Di Eugenio et al. 2000) includes goal-oriented dialogues in furniture purchasing negotiations and has been used to study intentional influence algorithms (Jordan 2000). Our dataset differs from the corpora above as it is dedicated to referring expressions in tabletop manipulation tasks, and expressions are purposed for object specification.

There are also numerous corpora solely dedicated to referring expressions. The Bishop corpus (Gorniak and Roy 2004) was collected through a task where one participant verbally described 30 randomly-positioned cones, one after another, to a partner. As compared to our corpus, the Bishop corpus involves many more unknown variables because the sequences of referred cones are decided by the participants and expressions may depend on previous dialogues. In the Drawer corpus (Viethen and Dale 2006), row and column numbers are used in referring expressions whereas in ours, row and column numbers, or x and y coordinates, are purposefully omitted. This is done to remain closely aligned with the kinds of natural language used when dealing with unorganized tabletop scenes encountered in our daily lives. Viethen and Dale (2008) collect GRE3D3, a corpus of expressions which specify one of three geometric objects. Our corpus introduces additional complexity in referring expressions as we place 15 objects in each scene. The TUNA corpus (Gatt et al. 2007) also differs from our corpus, as it contains referring expressions for 1 or 2 targets in a top-down view scene rather than a tabletop manipulation scene, and it does not involve perspective taking.

In Robotics, researchers have put considerable efforts into establishing corpora for use in navigational instructions which are inherently spatial tasks. Skubic et al. (2012) collected a set of indoor route following instructions, and MacMahon and Stankiewicz (2006) collected first-person navigational instructions in a virtual 3D environment. Unlike our dataset, they place emphasis on environmental landmarks rather than perspective or ambiguity. While these works focus on navigational trajectories, we focus on defining manipulation goals that are not necessarily constrained to particular trajectories.

Towards the goal of fluent interfacing for robotic tabletop manipulation, Bisk et al. (2016) contributed a dataset of instructions to transform one configuration of blocks on a tabletop to another in which the blocks are uniquely identified by a number or symbol on its faces. In our dataset, no numbers or symbols are available, which elicits a richer set of attributes used in block references. In addition, our instructions focus much more on object references given that the only action considered is the 'pick up' action.

Recently, there has been an increase in work on inferring groundings of natural language instructions via probabilistic graph in navigational tasks (Tellex et al. 2011; Howard et al. 2014; Boularias et al. 2015) and manipulation tasks (Paul et al. 2016), parsing natural language commands to robot control system (Matuszek et al. 2013), and reasoning about commands using cognitive architectures (Oh et al. 2015).

The dataset we present draws from a closely related line of work, with a greater emphasis on the ambiguities often encountered in real-world settings when issuing manipulation instructions. For example, in manipulating objects amongst clutters, it is common to inadvertently give instructions which refer to not only the target object, but also distractors. This raises the issues of resolving spatial references, understanding visual feature landmarks, and utilizing perspective in an effective manner. Li et al. (2016) uses this dataset to present initial results that explore these issues.

Data Collection Methods

Participants and Demographics

All participants were recruited through AMT. 120 participants were involved in the first study and 356 participants were involved in the second study. These two groups of participants were mutually exclusive.

Demographics were collected for each user. We asked participants to report their Age, Gender, Occupation, Computer Usage, Hand Dominance, English as Primary Language, and Experience with Robots, RC Cars, First-Person Shooter (FPS) Video Games, and Real-Time Strategy (RTS) Video Games.

We took care to ensure study designs avoided any confounding participant specific variables such as color-blindness. For example, red blocks were not used because red-green color blindness is the significantly predominant form of the deficiency (Judd 1943).

Study 1: Collecting Natural Language Instruction Corpus

The purpose of the first study was to collect a corpus of natural language used to instruct a partner in picking up blocks from a cluttered table. Participants gave instructions with respect to scenarios where it was often challenging to uniquely identify blocks. Thus, the use of spatial references or other visually apparent features were necessary for reliable target specification.

We created a set of stimulus images and presented each participant with a randomly selected subset. An example of a stimulus image is shown in Figure 1. Each image in the set consists of 15 randomly-spaced, randomly-colored blocks (orange, yellow, green, or blue) placed on a table with a silhouetted figure behind the table. The silhouetted figure represents a partner that the participant is to interact with. We synthesized 14 block configuration images and created 2 versions for each configuration by selecting differing target blocks indicated by a red arrow in both versions. In total, there are 28 unique scenarios in the set of stimuli Fig. 2.

We presented each participant with 14 randomly selected stimuli from the full set of 28. For each stimulus, we asked the participant to instruct the silhouetted figure to pick up the block indicated by the arrow. We randomly assigned each participant to one of two partner conditions: in the human-partner condition, participants were told the silhouette was another person; in the robot-partner condition, they were told the silhouette was a robot. The image of the silhouette stayed constant. The only difference across the two conditions was the word used to refer to the partner in the instructions. The participant entered their response into a textbox in typed natural language. We also asked the participant to subjectively rate the difficulty of creating the instruction for each stimulus on the Likert scale (1 (easy) to 5 (difficult)). At the end of the sequence of stimuli, we asked each participant 1) if they employed any particular strategy in completing the task, 2) how challenging they found the overall task, and 3) for their general comments.

We collected 1582 instructions in total. For each instruction, we also captured meta-data such as the total time it took the participant to write the instruction after being shown the stimulus.

Study 2: Evaluating Corpus

In order to evaluate the clarity of a given instruction, we defined a performance metric based upon object identification accuracy. For each instruction, we collected responses from a new set of participants indicating which block the participant believed the instruction specified within the corresponding stimulus scenario. The corresponding stimulus scenario refers to the image originally used to elicit an instruction with the red arrow removed. We showed the participant the stimulus/instruction pair with an interactive selection interface where participants were allowed to select the target blocks and change their selections before proceeding to the next scenario.

From the original set of 1582 instructions, we removed invalid responses, randomly sampled a subset of 1400 instructions, and collected 10 final answers per stimulus/instruction pair across all the new participants. Moreover, we also recorded the number of blocks each new participant clicked on before proceeding to the next scenario and computed the average accuracy for each original instruction.

Data Coding

We coded the corpus of instructions for two distinct purposes. The first purpose was to determine the ambiguity present within an instruction. To address this, we defined a set of criteria (Table 1) based upon whether an explicit perspective was used, and if so which perspective was used: *partner*, *participant*, or *neither*. We labeled each instruction with one of these perspective or the ambiguous *unknown* perspective which indicates a perspective was used, but not explicitly. In addition, we labeled each instruction with the number of blocks that could possibly be inferred from the instruction without making assumptions. For example, if there were 3 blocks in the stimulus image which fit the description 'red block near you', then the block ambiguity for the instruction 'Pick up the red block near you.' would be 3.

Perspective and block ambiguity was manually coded by four raters. In order to establish inter-rater reliability, we required each of the four coders to code the same 10% of the full dataset. We corrected for any discrepancies in coding until a high inter-rater reliability was achieved. This was confirmed by conducting pairwise Cohen's κ tests (Cohen 1968) and averaging the results. For coding perspective, the κ value was .85 which indicates a very high inter-rater reliability. For coding block ambiguity, the κ value was .68 which indicates a high inter-rater reliability. Once the reliabilities were established, each of the four coders processed one quarter of the data and the results were merged.

Dataset Details

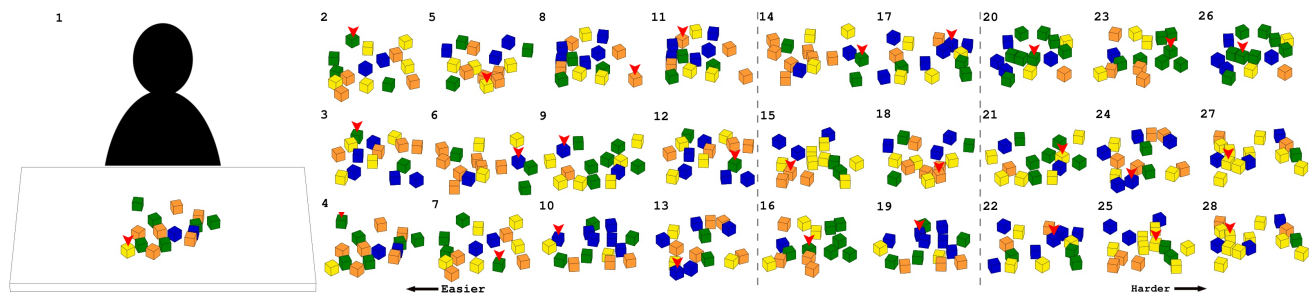The dataset can be found at the accompanying website[2].

Figure 2. All 28 table-top block configuration stimuli arranged by participant subjective difficulty rating

Type	P1	P2	Definition or Examples	
Participant Perspective	+	-	Referring to the speakers (egocentric)	
			Pick up the block that is to **my** rightest.	Pick up **my** leftmost blue block.
Partner Perspective	-	+	Referring to the addressees (addressee-centered)	
			Pick up the block second from **your** right.	Pick up the block on **your** left.
Neither Perspective	-	-	Does not take any perspectives	
			Pick up the red block in a triangle formation.	Pick up the block closest to you.
Unknown Perspective	?	?	Fails to state perspectives explicitly	
			Pick up the block to the **left** of the yellow block.	Pick up the block that is on the far **right**.

Table 1. Possible perspectives (P1-Participant, P2-Partner)

Instruction Corpus Table

The file named `NLICorpusData.csv` which contains the initially collected corpus of natural language instructions is formatted such that each line corresponds to an individual instruction (line 0 is a header which contains the field names). Note that there are multiple instructions which correspond to a given image. Here, we briefly describe each field's relationship to its corresponding instruction.

The *Instruction* field contains the string entered by the participant along with an *Index* for ease of reference. Each instruction corresponds to a particular *Scenario* image which is named by the originating image (synthesized block configuration) and its variant (v1 or v2). An example of an image name in this field is `Configuration_12_v2.png`. The *AgentType* indicates whether the instruction was given to a human or robot.

Each instruction was also given a *Difficulty* rating by the participant which indicates how challenging the they thought a particular prompt was when they generated the instruction. Additionally, the *TimeToComplete* for each instruction was recorded. The duration was started immediately following the current prompt loading and ended when the participant clicked next after entering their instruction and difficulty rating for the prompt.

At the end of the study, an individual participant is asked a series of questions to collect additional information and demographics. Note the participant is only asked once, however, for each instruction generated by the same participant, the corresponding answers are repeated in the table for ease of reference. These questions include *Strategy* which asked the user to enter their general comments on the strategies they might have used to generate instructions throughout the study, *Challenging* which asked how challenging the participant found the overall study, *GeneralComments* which asked for the participant's general comments on the study, the participant's *Age*, *Gender*,

Occupation, the *ComputerUsage* habits of the participant indicated by the number of hours per week, whether the participant has *EnglishAsFirst* language, and finally the participant's experiences with robots, RC cars, first-person shooter video games, and real-time strategy video games indicated by a number between 1-5 according to the Likert scale (where 1 is strongly disagree and 5 is strongly agree). Additionally, we assigned a unique user ID to each participant that is distinct from the Amazon Mechanical Turk ID.

A summary of all fields is shown in Table 2.

Corpus Evaluation Table

The file named `evaluationData.csv` summarizes the results from the evaluation study. Each line (aside from the header on line 0) corresponds to an instance of an instruction given to a participant. Note that there are 10 responses for each instruction and thus 14000 (1400*10) in total. Many of the fields are the same as those included in the instruction corpus table. We will briefly describe the non-overlapping fields found in the corpus evaluation table.

The *NumOfWords* field specifies the number of words in the corresponding instruction. *TargetBlockId* is the annotated block number that was specified when the instruction prompt was given (i.e. the correct target block). *ClickedBlockId* is the annotated block number that the participant clicked as their final decision before pressing the next button. *Correctness* is a boolean number indicating whether the clicked block was the intended target block. In other words, Correctness is true if and only if the participant made a final selection on the block that matches the one shown in the prompt during the first study when the instruction was generated.

Aside from these fields, there are a few which contain comments that correspond with a particular participant who took the study rather than an individual instruction. These include *DiffcultyComm* which asks for the participant's

comments on the overall study difficulty, *ObsHardComm* which asks for the participant's observations on what made instructions hard to understand, *ObsEasyComm* which asks for the participant's observations on what made instructions easy to understand, and *AddiComm* which asks for any additional comments the participant may have.

An additional file we include named `evaluationDataAvg.csv` contains the averaged responses from 10 participants for each instruction. Similarly to the instruction corpus table, the averaged evaluation table only contains one line per instruction where the corresponding fields reside on the same line. *ClickedBlockIdList* is a list of 10 annotated block numbers that correspond to each final block selection made across 10 participants that were assigned the particular corresponding instruction. It is composed of 10 of the previously mentioned ClickedBlockId fields in the evaluationData.csv file. *InternalIDList* is a list of the 10 participants respectively. In a similar fashion, *AccuracyAvg* indicates the average correctness for the corresponding instruction and *TimeToCompleteAvg* indicates the average time to complete the evaluation prompt for the corresponding instruction. Finally, the averaged evaluation table also includes the two manually coded fields *Ambiguity* which indicates the number of possible blocks a given instruction could possibly refer to if no assumptions are made and *Perspective* which indicates the frame of reference the instruction is given with respect to as defined in Table 1.

A summary of all fields is shown in Table 3. The fields below the dividing line in this table are only available in the averaged evaluation data.

Accessing the Dataset

Natural Language Instruction Corpus

We provide simple Python code to initialize data structures which allow for versatile access of the dataset. The file `access_NLICorpusData_CSV.py` loads `NLICorpusData.csv` and creates two data structures in Python for people to access the data (`dictionary` and `list` are Python data structures):

1. a `dictionary` of `list`: Each `list` stores the values of one particular field across all the participants, such as *NumOfWords*.
2. a `list` of `dictionary`: Each `dictionary` stores the values of all the fields for only one particular participant, such as *NumOfWords*.

Note only one data structure is necessary for access, however each provides different indexing benefits. The argument for the dictionary is a key corresponding to the field in the table header while the argument for the list is the index corresponding to a particular instruction. Basic examples are included to demonstrate intended usage.

Supplementary Corpus Evaluation

Similarly, we provide Python code for accessing the corpus evaluation table. This code functions in exactly the same manner as the access code for the NLI corpus table. The file `access_evaluationData_CSV.py` loads both `evaluationData.csv` and `evaluationDataAvg.csv` and creates appropriate data structures.

For the corpus evaluation table, we include the original data collected in the JSON format as well as the easier to view CSV format. We provide access code named `access_evaluationData_JSON.py` for the files `evaluationData.json` and `evaluationDataAvg.json`. Note this data is exactly the same as the above CSV version - it is simply included for completeness.

Acknowledgements

Notes

1. https://www.mturk.com
2. https://personalrobotics.github.io/collaborative_manipulation_corpus/

References

A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, et al. The hcrc map task corpus. *Language and speech*, 34(4):351–366, 1991.

D. Berenson and S. S. Srinivasa. Grasp synthesis in cluttered environments for dexterous hands. In *Proceedings of the 8th IEEE-RAS International Conference on Humanoid Robots*, pages 189–196. IEEE, 2008.

Y. Bisk, D. Yuret, and D. Marcu. Natural language communication with robots. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, San Diego, CA, June 2016.

A. Boularias, F. Duvallet, J. Oh, and A. Stentz. Grounding spatial relations for outdoor robot navigation. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1976–1982. IEEE, 2015.

J. Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

B. Di Eugenio, P. W. Jordan, R. H. Thomason, and J. D. Moore. The agreement process: An empirical investigation of human–human computer-mediated collaborative dialogs. *International Journal of Human-Computer Studies*, 53(6):1017–1076, 2000.

A. Gatt, I. Van Der Sluis, and K. Van Deemter. Evaluating algorithms for the generation of referring expressions using

Table 2. Overview of fields in Natural Language Instruction Corpus.

Parameter	Note	Data Format
`Instruction`	Given by participant as response to prompt	String
`Index`	Corresponds to individual instruction	Integer 1–1582
`Scenario`	Configuration by image and variation	String with Configuration 1–14 and variation 1–2
`AgentType`	Agent participant is instructing	String from $\{robot, human\}$
`Difficulty`	Rating for a scenario given by participant	Integer 1–5 (5 being most difficult)
`TimeToComplete`	Duration participant took to create instruction	Time in hours:minutes:seconds
`Strategy`	Comments on strategy for used	String
`Challenging`	Comments on overall study difficulty	String
`GeneralComments`	Comments on general on overall study	String
`Age`	Age of participant	Integer in years
`Gender`	Gender of participant	String from $\{Male, Female, Prefer\ not\ to\ answer\}$
`Occupation`	Occupation of participant	String
`ComputerUsage`	Duration participant uses a computer per week	String from $\{0\text{–}5, 5\text{–}10, 10\text{–}15, 15\text{–}20, >20\}$ in hours
`DominantHand`	Dominant hand of participant	String from $\{Left, Right, Ambidextrous\}$
`EnglishAsFirst`	Participant's first language	1 if yes, 0 if no
`ExpWithRobots`	Participant experience with robots	Likert scale 1–5
`ExpWithRCCars`	Participant experience with RC cars	Likert scale 1–5
`ExpWithFPS`	Participant experience with first-person shooter video games	Likert scale 1–5
`ExpWithRTS`	Participant experience with real-time strategy video games	Likert scale 1–5
`ExpWithRobotComments`	Comments on experience with robots	String
`InternalUserID`	Participant ID unaffiliated with Amazon ID	Integer 1–600

Table 3. Overview of fields in Corpus Evaluation.

Parameter	Note	Data Format
`Instruction`	Instruction being evaluated	String
`Index`	Corresponds to individual instruction	Integer 1–1582
`Scenario`	Configuration by image and variation	String with Configuration 1–14 and variation 1–2
`NumOfWords`	Number of words within an instruction	Positive Integer
`TargetBlockId`	Corresponds to the target block of an instruction	Integer 1–15
`ClickedBlockId`	Corresponds to the block participant finally selected	Integer 1–15
`Correctness`	Whether participant chose the correct target block	1 if right, 0 if wrong
`TimeToComplete`	Duration participant took to evaluate an instruction	Float number in seconds
`DiffcultyComm`	Comment on the general difficulty of overall task.	String
`ObsHardComm`	Observation on instructions difficult to interpret	String
`ObsEasyComm`	Observation on instructions easy to interpret	String
`AddiComm`	Comment in general on overall study	String
`ClickedBlockIdList`	Corresponds to the blocks multiple participants respectively selected	List of Integers 1–15
`AccuracyAvg`	Average correctness among multiple participants	Float 0.0–1.0
`TimeToCompleteAvg`	Average duration among multiple participants	Float number in seconds
`InternalUserIDList`	Multiple participant IDs unaffiliated with Amazon ID	Integer 1–600
`Ambiguity`	Manually coded number of potential references	Integer
`Perspective`	Manually coded perspective instruction uses	Integer

a balanced corpus. In *Proceedings of the 11th European Workshop on Natural Language Generation*, pages 49–56. Association for Computational Linguistics, 2007.

P. Gorniak and D. Roy. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470, 2004.

M. Guhe and E. G. Bard. Adapting referring expressions to the task environment. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society (CogSci)*, pages 2404–2409, 2008.

T. M. Howard, S. Tellex, and N. Roy. A natural language planner interface for mobile manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 6652–6659. IEEE, 2014.

P. W. Jordan. *Intentional influences on object redescriptions in dialogue: evidence from an empirical study*. PhD thesis, University of Pittsburgh, 2000.

D. B. Judd. Facts of color-blindness. *Journal of the Optical Society of America*, 33(6):294–307, Jun 1943. doi: 10.1364/JOSA.33.000294.

B. Keysar, D. J. Barr, J. A. Balin, and J. S. Brauner. Taking perspective in conversation: The role of mutual knowledge in comprehension. *Psychological Science*, 11(1):32–38, 2000.

E. Krahmer and K. Van Deemter. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38 (1):173–218, 2012.

S. Li, R. Scalise, H. Admoni, S. Rosenthal, and S. Srinivasa. Spatial references and perspective in natural language instructions for collaborative manipulation. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication*, 2016.

M. MacMahon and B. Stankiewicz. Human and automated indoor route instruction following. *Def*, 2(6):4, 2006.

C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *Proceedings of International Symposium on Experimental Robotics (ISER)*, pages 403–415. Springer, 2013.

J. H. Oh , A. Suppe, F. Duvallet, A. Boularias, J. Vinokurov, L. E. Navarro-Serment, O. Romero, R. Dean, C. Lebiere, M. Hebert , and A. T. Stentz . Toward mobile robots reasoning like humans. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, January 2015.

R. Paul, J. Arkin, N. Roy, and T. M. Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016. doi: 10.15607/RSS. 2016.XII.037.

M. Scheutz, P. Schermerhorn, and J. Kramer. The utility of affect expression in natural language interactions in joint human-robot tasks. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 226–233. ACM, 2006.

M. Skubic, T. Alexenko, Z. Huo, L. Carlson, and J. Miller. Investigating spatial language for robot fetch commands. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence Workshop*, 2012.

S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1507–1514, San Francisco, CA, August 2011.

J. Viethen and R. Dale. Algorithms for generating referring expressions: do they do what people do? In *Proceedings of the Fourth International Natural Language Generation Conference*, pages 63–70. Association for Computational Linguistics, 2006.

J. Viethen and R. Dale. The use of spatial relations in referring expression generation. In *Proceedings of the 5th International Natural Language Generation Conference*, pages 59–67. Association for Computational Linguistics, 2008.

J. Viethen, S. Zwarts, R. Dale, M. Guhe, et al. Dialogue reference in a visual domain. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2010.