The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence
Symbiotic Cognitive Systems: Technical Report WS-16-14

Towards a Dataset for Human Computer Communication via Grounded Language Acquisition

Yonatan Bisk, Daniel Marcu and William Wong
Information Sciences Institute
Department of Computer Science
University of Southern California
{ybisk,marcu}@isi.edu, wjwong@gmail.com

Abstract

The Natural Language Processing, Artificial Intelligence, and Robotics fields have made significant progress towards developing robust component technologies (speech recognition/synthesis, machine translation, image recognition); advanced inference mechanisms that accommodate uncertainty and noise; and autonomous driving systems that operate seamlessly on our roads. In spite of this, we do not yet know how to talk to the machines we build or have them speak to us in natural language; how to make them smarter via simple, natural language instructions; how to understand what they are about to do; or how to work with them collaboratively towards accomplishing some joint goal.

In this paper, we discuss our work towards building a dataset that enables an empirical approach to studying the relation between natural language, actions, and plans; and introduce a problem formulation that allows us to take meaningful steps towards addressing the open problems listed above.

Context for our work

Establishing a bidirectional connection between natural language and actions & goals in the physical world requires mechanisms that will enable the joint acquisition of language and learning of planning policies; learning of mappings between natural language utterances and action sequences (understanding); and learning of mappings between action sequences and natural language (generation). Several research areas have addressed aspects of our requirements.

In natural language understanding and language grounding, NLP researchers (Zettlemoyer and Collins 2005; Kwiatkowski et al. 2013; Reddy, Lapata, and Steedman 2014; Berant et al. 2013; Roth et al. 2014) have produced systems for extracting logical representations from text and for linking concepts and entity mentions to relations and entities in large scale knowledge bases or Wikipedia entries. In this body of work, the main focus is on establishing a relation between language, objects, and properties of objects.

Within robotics, there is a long history of work on Human-Robot Interaction (Klingspor, Demiris, and Kaiser 1997; Goodrich and Schultz 2007; Mavridis 2015) which take commands and generate responses from a fixed vocabulary and grammar. Planning advances that increase robot autonomy do not correlate to free-er form language or higher level abstractions as should be the case when acquiring language. Progress on language has largely focused on grounding visual attributes (Kollar, Krishnamurthy, and Strimel 2013; Matuszek et al. 2014) and on learning spatial relations and actions for small vocabularies with hard-coded abstract concepts (Steels and Vogt 1997; Roy 2002; Guadarrama et al. 2013). Language is sometimes grounded into simple actions (Yu and Siskind 2013) but the data, while multimodal, is formulaic, the vocabularies are small, and the grammar is constrained. Recently the connection between less formulaic language and simple actions has been explored successfully in the context of simulated worlds (Branavan et al. 2009; Goldwasser and Roth 2011; Branavan, Silver, and Barzilay 2011; Artzi and Zettlemoyer 2013) and videos (Malmaud et al. 2015; Venugopalan et al. 2015).

To our knowledge, there is no body of work on understanding the relation between natural language and complex actions and goals or from sequences of actions to natural language. As observed by Klingspor (1997), there is a big gap between the formulaic interactions that are typical of state-of-the-art human-robot communications and human-human interactions, which are more abstract.

Proposed Problem-Solution Sequences Data

An intuitive characterization

We have built a large corpus of Problem-Solution Sequences (PSS) (see Figure 1 for an example). A PSS consists of a sequence of images/frames that encode what a robot sees as it goes about accomplishing a goal. To instantiate our framework, we focus first on PSSs specific to a simple world that has blocks stacked on a table and a robot that can visually inspect the table and manipulate the blocks on it.

In Figure 1, each image in the PSS corresponds to an intermediate block configuration that the robot goes through as it modifies the initial state block configuration into the final one. The PSS makes explicit the natural language instructions that a human may give to a robot in order to transform the configuration in an Image$_i$ into the configuration in an Image$_j$ - the two configurations may correspond to one robot action (for adjacent states in the sequence) or to a sequence of robot actions (for non-adjacent states). To account for lan-

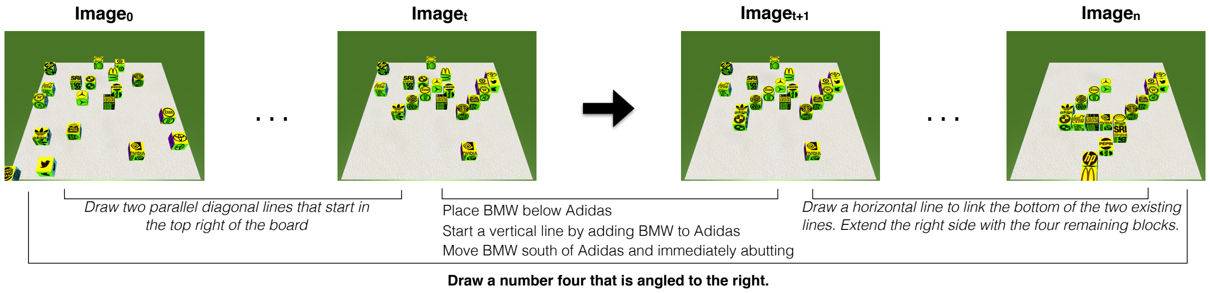

Image$_0$
Image$_t$
Image$_{t+1}$
Image$_n$
Draw two parallel diagonal lines that start in the top right of the board
Place BMW below Adidas
Start a vertical line by adding BMW to Adidas
Move BMW south of Adidas and immediately abutting
Draw a horizontal line to link the bottom of the two existing lines. Extend the right side with the four remaining blocks.
Draw a number four that is angled to the right.

Figure 1: A PSS where descriptions have been provided at differing levels of abstraction (light, *italic*, **bold**) with the eventual goal of using the blocks to build a structure that looks like the digit four.

guage variance, each simple action or sequence of actions is associated with a set of alternative natural language instructions. For example, Figure 1 shows three alternative ways of instructing a robot what to do to build Image$_{t+1}$ given the configuration in Image$_t$. The natural language instructions encode implicitly partial or full descriptions of the world ("diagonal line" in Image$_t$ or "number four" in Image$_n$).

PSS acquisition

To support a research program aimed at addressing the open questions enumerated in the Abstract, we created a virtual environment that simulates closely the physical environment of (Tamrakar 2015). We have complete control of our virtual environment: what kind of blocks it contains (size, color, shape); their locations; what actions are available to manipulate the blocks (move, lift, drop, rotate, etc); what happens when an action is taken (for example, dropping a block over another may lead to rolling under a realistic physical model); etc. We can initialize the environment with any block configuration and track changes as the robot takes actions.

Most importantly, for each state in an action sequence that we choose to sample, we have access not only to an image/frame associated with that state (which we can manipulate further via changes in perspective and angle of view) but to the ground truth associated with that image as well. For each image, we know the set of blocks in that image, their unique IDs, their characteristics (color, size, etc.), their position ((x, y, z) coordinates) and orientation (θ).

We generate PSSs in two steps: (i) we first generate coherent sequences of actions using our simulated environment (ii) subsequently, we decorate PSSs with meaning equivalent natural language instructions that we collect for simple and complex actions in the PSS via Amazon Mechanical Turk.

To date, we experimented with two approaches to generating coherent sequences of actions. (i) First, we generated an initial state by randomly choosing and placing a block from a predefined set. We then perform random actions while taking a snapshot of the current state after each action. (ii) In the second approach, we start with meaningful block structures - letters, digits, objects, etc. At each step, we move a random block to an arbitrary location. Depending on the complexity of the initial structure, within 10-20 steps, we end up with a random looking configuration. By reversing the order of this sequence, we obtain a sequence of actions that start from a random configuration and end up in a meaningful structure.

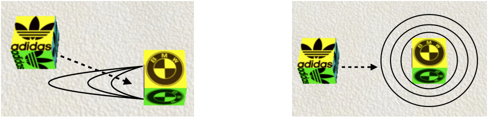
For the second approach to PSS generation, we experimented with block configurations that resemble digits (zero through nine). We took the MNIST corpus of hand-written digits (LeCun et al. 1998) and used 10 examples of each digit to serve as goal configurations. We sharpened and downsampled each image until it had at most 20 active pixels which we replaced with numbered blocks that were uploaded into our virtual world. For each goal configuration, we took up to 20 random actions, to unrecognizably scramble the blocks. Via this process, we obtained 100 initial PSS sequences (10 digits x 10 examples/digit). Figure 1 shows the initial, final, and two other states in one of our PSS sequences.

Our initial release of the *ISI Language Grounding Data Set* is available at http://nlg.isi.edu/language-grounding/. The 50 MNIST sequences drawn with logo decorated blocks yield 8,514 annotations of individual actions (and 996 corresponding scenes). The complete dataset also contains blocks decorated with digits, completely random sequences, and abstract commands over sequences.

We believe our approach to creating PSS representations is highly scalable both in terms of the types of objects/structures and the task complexity. For example, one can imagine replicating the same approach we advocate here when modeling a prototypical office or home environment with robotic actions and objects that are typical of those environments.

PSS – a formal characterization

A PSS state S_t at time t is a tuple of an image/frame IM_i (what the robot sees) and a list of objects with their characteristics, positions, and orientations associated with that image $[\ \mathrm{ID}_i, \vec{c_i}, \vec{x_i}, \vec{\theta_i}\ ]_t^{i:1\ldots n}$. A direct action arc between two adjacent states S_t and S_{t+1} is labeled with an action a_t from a set of actions, A, the robot may perform. A valid PSS connects all states from the initial state S_0 to the final state S_N via action arcs. Two states S_i and S_j, where $i < j$, can also be connected via a text instruction arc $u_{i,j}$, which stores a set of context-dependent natural language instructions $\vec{U}_{i,j}$. Given a state S_i, if the robot correctly executes the instructions expressed in $\vec{U}_{i,j}$, it should end up in state S_j.



(a) Move the Adidas block to the left of the BMW block.

(b) Move Adidas adjacent to the BMW block.

Figure 2: Two move commands with inexact goal locations.

Utility of the PSS representations

Once we generate tens of thousands of PSSs, we believe that we can use machine learning techniques, automatic evaluation metrics, and a fast generate-test idea development cycle to make rapid progress on a wide range of problems.

Context-dependent language understanding and grounding of simple actions

We first introduce a simple representation that mitigates the gap between natural language instructions, PSS states, PSS actions, and the continuous nature of the physical environment. We assume a simple semantic representation with only one predicate *move* that takes three arguments: the ID of the block to move, the target location, and the target orientation: $move(ID, \Psi_k(\vec{x}), \rho_k(\vec{\theta}))$. Because physical environments are continuous, the target location and orientation are not exact, but density functions defined over the space of the table and space of 360 degree possible orientations respectively. Figure 2, visualizes density functions (meanings) of the phrases (a) ”to the left of the BMW block” and (b) ”adjacent to the BMW block”. These are not “precise” or exact locations, rather they define a region around the BMW block.

In this context, understanding and grounding a natural language instruction amounts to mapping it to an appropriate triple: *move(ID*, $\Psi_k(\vec{x}), \rho_k(\vec{\theta})$). Table 1 lists a set of simple instructions with segments marked that should lead to the identification of the predicate *move*, the ID of the object to be moved, and its target location $\Psi_k(\vec{x})$; in these examples, we assume a default orientation $\rho_k(\vec{0})$.

Given large PSS datasets, one can develop a wide range of techniques for interpreting natural language instructions in context. The performance of these understanding and grounding algorithms can be tested automatically by measuring the degree to which the action inferred matches the action stored in the PSS representation.

Learning unknown language primitives from PSS

Initial approaches, as is common, may assume that the mappings between words and semantic primitives are known. Our goal is to enable the learning of unknown mappings from scratch by exploiting the size of the data we collect. For example, one may not know how to interpret “next to the corner of X”; but after many examples, the density function $\Psi_{\textit{next to the corner of}} : \vec{x} \rightarrow P$ could be learned from data. This may lead to a set of Ψ_k such that Ψ_1 corresponds to “left”, Ψ_2 to “right”, and so forth. Given this density function, we can sample a location ($\vec{y}$) to learn new meanings.

{move}	{block 3}	{adjacent to the green block}
{place}	{the block closest to the front}	{at the back of the table}
{push}	{the block to the left of block 1}	{to the right of block 2}
{slide}	{it}	{more}

Table 1: Utterances split into predicate, ID and location.

Understanding and grounding of complex actions

PSS datasets also make explicit textual descriptions of action sequences (“build a line of four blocks” or “build a tower in the left corner of the table”) that are consistent with sequences of actions that lead from a state S_i to a state S_j. To understand and ground such complex instructions, the agent must be able to infer in state S_i a sequence of move operations in the service of an abstract goal that subsumes the image in state S_j. Fortunately, PSSs contain complete action sequences that correspond to complex instructions. PSSs enable the study of language understanding and grounding in the context of increasingly difficult problem scenarios.

In these scenarios, grounding is more ambiguous because the use of the blocks (and perhaps which blocks) is assumed and therefore not referenced in the instruction. Also, any block might serve as the first piece of the tower/line/... that provides the initial $\vec{x}$ to which Ψ_k is applied recursively. We can see this by thinking about the set of instructions/layout necessary for drawing a simple object (Figure 3).

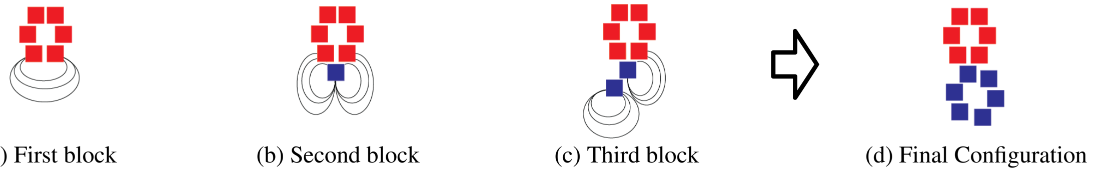
Here, we might have a function for drawing circles or figure eights which is simply conditioned on the current state of the table and can make predictions for the next action which when taken define a new state to be fed back into Ψ_{fig8} until the figure is complete. PSSs allow us to investigate how complex meanings relate to simple ones in the context of action sequences; and to evaluate the quality of our understanding and groundings objectively by comparing the states our algorithms predict with the states observed in the data.

Grounded grammar induction

Grammar induction is the task of discovering the latent grammar G for a given language. Traditionally, only utterances ($\vec{w}$) are available as input. In our framework, we can investigate grammar induction algorithms that have access not only to utterances ($\vec{w}$), but to the contexts in which they can be uttered (the states S_i) and their intent (the resulting states S_{i+1}). The initial and resulting context associated with utterances should reduce the number of nonsensical interpretations that make grammar induction so difficult.

Planning and plan understanding

Planning and plan understanding is fundamental to the task as policies must be properly learned for any particular action or abstract instruction provided. PSSs force joint learning of plan operators and natural language interpretations to determine and achieve an uncertain goal.



(a) First block

(b) Second block

(c) Third block

(d) Final Configuration

Figure 3: Possible placements of blocks when hearing *“Attach a second circle below the first one”* or *“Complete the Figure 8”*

Context-dependent language generation

Language generation is another research area that is enabled by a large PSS dataset. Our discussion has primarily focused on the use of language to learn a plan or grounded semantic representation. Of equal importance is the ability to produce natural language utterances ($\vec{w}$) given either a sequence of scenes or a list of actions that were performed. Until robots can respond to us using abstract language, we will not be fully able to treat them as collaboration partners.

PSSs and vision

By construction, all research problems above can be tackled in the PSS context with perfect image interpretations (the characteristics, locations, and orientations of all objects in a scene). However, PSSs also provide the opportunity to assess the degree to which the performance of current vision algorithms impacts the performance of systems designed to solve the problems discussed herein. For example, PSSs allow one to work from noisy outputs produced by image understanding software, and study how the performance changes as images are made noisy, include occlusion, or as the camera offers a different perspective/angle on the scene.

Acknowledgments

This work was supported by Contract W911NF-15-1-0543 with the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO).

References

Artzi, Y., and Zettlemoyer, L. S. 2013. Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions. *Transactions of the ACL* 49–62.

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proc. of EMNLP*.

Branavan, S.; Chen, H.; Zettlemoyer, L. S.; and Barzilay, R. 2009. Reinforcement Learning for Mapping Instructions to Actions. In *Proc. of ACL-IJCNLP*.

Branavan, S. R. K.; Silver, D.; and Barzilay, R. 2011. Learning to Win by Reading Manuals in a Monte-Carlo Framework. In *Proc. of ACL*, 268–277.

Goldwasser, D., and Roth, D. 2011. Learning From Natural Instructions. In *Proc. of IJCAI*.

Goodrich, M. A., and Schultz, A. C. 2007. Human-Robot Interaction: A Survey. *Foundations and Trends in Human Computer Interaction* 1(3):203–275.

Guadarrama, S.; Riano, L.; Golland, D.; Göhring, D.; Yangqing, J.; Klein, D.; Abbeel, P.; and Darrell, T. 2013. Grounding Spatial Relations for Human-Robot Interaction . In *Proc. of IROS*, 1640–1647.

Klingspor, V.; Demiris, J.; and Kaiser, M. 1997. Human-Robot-Communication and Machine Learning. *Applied Artificial Intelligence Journal* 11:719–746.

Kollar, T.; Krishnamurthy, J.; and Strimel, G. 2013. Toward Interactive Grounded Language Acquisition. In *Robotics: Science and Systems*.

Kwiatkowski, T.; Choi, E.; Artzi, Y.; and Zettlemoyer, L. S. 2013. Scaling Semantic Parsers with On-the-Fly Ontology Matching. In *Proc. of EMNLP*, 1545–1556.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proc. of the IEEE* 86(11):2278–2324.

Malmaud, J.; Huang, J.; Rathod, V.; Johnston, N.; Rabinovich, A.; and Murphy, K. 2015. Whats cookin? interpreting cooking videos using text, speech and vision. In *Proc. of NAACL-HLT*, 143–152.

Matuszek, C.; Bo, L.; Zettlemoyer, L. S.; and Fox, D. 2014. Learning from Unscripted Deictic Gesture and Language for Human-Robot Interactions. In *Proc. of AAAI*.

Mavridis, N. 2015. A review of verbal and non-verbal human–robot interactive communication. *Robotics and Autonomous Systems* 63:22–35.

Reddy, S.; Lapata, M.; and Steedman, M. 2014. Large-scale Semantic Parsing without Question-Answer Pairs. *Transactions of the ACL* 1–16.

Roth, D.; Ji, H.; Chang, M.-W.; and Cassidy, T. 2014. Wikification and Beyond: The Challenges of Entity and Concept Grounding. Tutorial at the 52nd Meeting of the ACL.

Roy, D. K. 2002. Learning visually grounded words and syntax for a scene description task. *Computer speech & language* 16(3-4):353–385.

Steels, L., and Vogt, P. 1997. Grounding Adaptive Language Games in Robotic Agents. *Proceedings of the Fourth European Conference on Artificial Life*.

Tamrakar, A. 2015. CwC Blocks World Apparatus: API Reference Manual. Technical report, Stanford Research International.

Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R.; and Saenko, K. 2015. Translating videos to natural language using deep recurrent neural networks. In *Proc. of NAACL-HLT*, 1494–1504.

Yu, H., and Siskind, J. M. 2013. Grounded language learning from video described with sentences. In *Proc. of ACL (Volume 1: Long Papers)*, 53–63.

Zettlemoyer, L. S., and Collins, M. 2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *Proc. of UAI*.