Advances in Cognitive Systems X (20XX) 1-6 Submitted X/20XX; published X/20XX

Neurosymbolic AI for Situated Language Understanding

Nikhil Krishnaswamy NKRISHNA@COLOSTATE.EDU
Department of Computer Science, Colorado State University, Fort Collins, CO USA
Department of Computer Science, Brandeis University, Waltham, MA USA

James Pustejovsky JAMESP@BRANDEIS.EDU
Department of Computer Science, Brandeis University, Waltham, MA USA

Abstract

In recent years, data-intensive AI, particularly the domain of natural language processing and understanding, has seen significant progress driven by the advent of large datasets and deep neural networks that have sidelined more classic AI approaches to the field. These systems can apparently demonstrate sophisticated linguistic understanding or generation capabilities, but often fail to transfer their skills to situations they have not encountered before. We argue that computational *situated grounding* provides a solution to some of these learning challenges by creating situational representations that both serve as a formal model of the salient phenomena, and contain rich amounts of exploitable, task-appropriate data for training new, flexible computational models. Our model reincorporates some ideas of classic AI into a framework of *neurosymbolic intelligence*, using multimodal contextual modeling of interactive situations, events, and object properties. We discuss how situated grounding provides diverse data and multiple levels of modeling for a variety of AI learning challenges, including learning how to interact with object affordances, learning semantics for novel structures and configurations, and transferring such learned knowledge to new objects and situations.

1. Introduction

Over the past fifteen to twenty years, AI has seen remarkable growth, from a bust beset by disillusionment with unmet expectations to a behemoth at the center of modern computer science, and a maturing set of technologies to match (Menzies, 2003; McCarthy, 2007; Liu et al., 2018). A significant proportion of this growth has been driven by advances in natural language processing (NLP), previously a difficult problem with brittle solutions, and now a mainstay of technologies in everyday use. Developers without substantial prior knowledge of AI or linguistics can now use robust pipelines for natural language tasks such as tokenization, parsing, or speech recognition. Within the last decade, the 2010s, NLP progress was kicked into overdrive, largely due to the developments in deep learning and the concurrent emergence of large datasets and affordable GPUs for processing them. Deep learning has been applied to tasks such as question answering (Sultana & Badugu, 2020), dialogue systems (Zaib et al., 2020), and text generation (Iqbal & Qureshi, 2020).

Many of the biggest successes in NLP have been driven by large, pre-trained, task-agnostic language models, such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), and the GPT

family (Radford et al., 2019). These language models lend themselves well to transfer learning with task-specific fine tuning, and facilitate the generation of text that is grammatical, largely coherent, and usually on-topic given an initial prompt. They are also simple to deploy and well-pipelined for general use in larger applications or just as a demonstration of capability. An example is given below, generated using the GPT-2 language model, a transformer based language model (Vaswani et al., 2017) trained over 40GB of text from the internet. The input is the first sentence of the Wikipedia article on NLP, given in bold, and continued by the language model.

> ***Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.*** *NLP encompasses methods to manipulate and categorize textual data and information in ways similar to biological organisms. As such, NLP is an excellent use case for learning intelligent bots, as you can teach a computer how to process natural language data in order to explore, build, test, validate, and apply advanced learning methods in response to the data [...]*

The generated text is largely syntactically and semantically correct and mostly on topic, with a few instances of questionable phrasing (e.g., "learning intelligent bots") or possibly off-topic content or factual inaccuracies (e.g., references to categorizing language like biological information). The system is talking about NLP as one might expect a non-expert to, the errors it makes are similar to those a human layperson might make, it incorporates a significant amount of contextual information in the generated sample, and it is clear that NLP systems have come a long way on this task from the days of finite state automata and simple N-gram models.

Nonetheless, despite the apparent success of language models in NLP tasks, there are a number of ways in which they fail to demonstrate intelligence or understanding as commonly defined, in particular on tasks that would typically be trivial for a human. In previous work (Krishnaswamy & Pustejovsky, 2019b; McNeely-White et al., 2019), we have given the example of asking a computer or smartphone the simple (for a human) question "What am I pointing at?" Put simply, an NLP system such as one embedded in a smartphone or smart-speaker lacks the machinery to answer the question, being unable to see you or the surrounding context you and the device inhabit, and so tends to dodge the question[1]. Large predictive language models fare little better, as shown below.

> ***Q: What am I pointing at?***
>
> *A: There was nothing I could detect at the start of the game but not many areas. It took some training to find my way around but by the end of the game I had discovered the best way to move and how to take advantage of my surroundings [...]*

Although trained on a enormous amount of text, these models lack knowledge of the current situational context, because that context is supplied using non-textual modalities, and so the main advertised advantage of a pre-trained language model—the ability to transfer knowledge learned from observed text to previously unencountered text—disappears. We now have many usable interactive systems, such as smartphones and the whole internet-of-things, but the large datasets and

[1]Try it with your device! Answers may include "Interesting question" (Siri) or "You're a great person to chat with!" (Google Mini).

compute power that facilitate high-performing NLP fail in many contexts in which we might wish to use these devices, and might expect them to function as if they truly understand us. Put simply, the current state of the technology runs up against a wall because these systems exist in a situated context (a home, an office, a car, a pocket, etc.), but lack the ability to validate information across the different modalities of description that might be implicated in all these situations. They also lack background knowledge about other entities present in the situation. Therefore, how can we expect to interface with these devices when something so basic to a human—like "What am I pointing at?"—fails?

In this paper, we will discuss our *situated grounding* approach to multimodally encoding context, and our platform, *VoxWorld*, which demonstrates real-time modeling of context through multimodal grounding of object and event properties in a simulation environment, and the *common ground* that arises between interlocutors in the course of an interaction. We will demonstrate how situated grounding methods within VoxWorld provide diverse types of data suited to a number of different learning challenges within AI, and how deploying this data and their associated models within a *neurosymbolic* intelligence framework addresses three novel challenges in AI: affordance learning, structure/configuration learning, and transfer learning.

2. Multimodal Communication in Context

As sophisticated as current task-based AI systems are and as intelligent as they can behave in their domains, they often fail in understanding and communicating crucial information about their situations. Robust communicative interaction between humans and computers requires that:

1. All parties must be able to recognize input and generate output within multiple modalities appropriate to the context (e.g., language, gesture, images, actions, etc.);

2. All parties must demonstrate understanding of contextual grounding and the space in which the conversation takes place (e.g., co-situated in the same space, mediated through an interface, entirely disconnected, etc.);

3. All parties must appreciate the consequences of actions taken throughout the dialogue.

Central to all these is the notion of *semantically grounding* a concept to a situation. Importantly, certain modalities are better suited to grounding certain kinds of information than others (for example, deictic gesture—i.e., pointing—grounds naturally to locations, while language may be better at grounding concept labels or attribute descriptions). Nonetheless, "grounding" in currently-practiced NLP typically refers to kinds of multimodal *linking*, such as semantic roles to entities in an image (Yatskar et al., 2016), or joint linguistic-visual attention between a caption and an image (Li et al., 2019).

This type of annotation and training on large quantities of multimodal data resembles a cross-modal linking equivalent of the traits that have made large language models successful at their tasks: they are annotated, they contain structured data, and they contain mechanisms for extracting many sophisticated linguistic features, even if they have no built-in understanding of linguistic structure. In fact, they resemble the classic NLP pipeline but on a much larger scale (Tenney et al., 2019).

When applied to multimodal data, however, and pertaining to the representation of context, the same classic NLP pipeline (or deep-learned equivalent) fails to work as well.

Multimodal tasks rely on the contexts established between and across modalities (Matuszek, 2018), and so we propose that the difficulties faced by multimodal end-to-end systems, as well as the difficulty evaluating the state of the task is largely because contextual encoding still tends to be hit-or-miss, and the nature of the analytic and structural units of context, as humans use for sensitive contextual reasoning, remain the subjects of debate.

Nevertheless, human reasoning is sensitive to contextual modeling, and methods of contextual modeling in AI have followed the field from logical-symbolic models of context ("good old-fashioned AI" or *GOFAI*) before the AI winter of the 1980s to probabilistic and deep-learned vector similarity in the 2010s. Recently, GOFAI methods have resurfaced in the increasingly machine learning-driven modern AI community as a method of reincorporating some of the structure they provide into the flexible representations provided by deep learning (e.g., Besold et al. (2017); Garcez et al. (2019); Mao et al. (2019); Marcus & Davis (2019).[2] The question of better incorporating contextual structure into deep learning necessarily raises the question of the analytic and structural units of context.

Following on Clark et al. (1983), Asher (1998), Stalnaker (2002), Tomasello & Carpenter (2007), Abbott (2008), Pustejovsky (2018), and others, we have previously proposed the notion of a *computational common ground* that emerges between interlocutors as they interact, and facilitates further communication by providing common knowledge among agents (Pustejovsky et al., 2017). Common ground is one such method of encoding and analyzing situational and conversational context.

Entity Type	**Examples**
Agents	"you," "I," "us," etc.
Beliefs, desires, intentions	**Know**$_{mother}$smile(son), etc. "Goals under Discussion"
Objects	cups, plates, knives, "it," "them," etc.
Space	$\mathcal{E}$ (Embedding space)

Figure 1: Two humans interacting in a shared task with example common ground entities.

We break down computational common ground into representations of: the agents interacting; their beliefs, desires and intentions (BDI); the objects involved in the interaction; and the minimal embedding space $\mathcal{E}$ required to execute the activities implicated during the course of the interaction. All these parameters also include the terms used to discuss them. For instance, in Fig. 1, the mother and son agree that they share a goal to, e.g., put the dishes away, empty the sink, clean the dishes, etc. (if one of them does not share this belief, this impacts the way both of them will communicate about the task and their beliefs about it). This in turn implicates the properties of the objects involved, e.g., what it means to have a clean plate vs. a dirty plate with relation to what a plate is for.

[2] As well as keynote addresses given at AAAI 2020 by David Cox of IBM, Henry Kautz of the University of Rochester, and Turing Award winners Geoffrey Hinton, Yann LeCun, and Yoshua Bengio.

Object properties are a topic of much discussion in semantics, including Generative Lexicon theory (Pustejovsky, 1995; Pustejovsky & Batiukova, 2019), and are also of interest to the robotics community (Dzifcak et al., 2009). Object properties, though important for theoretical semantics and practical applications of modern intelligent systems, pose a problem for even some of the most sophisticated task-based AI systems. A formal structure provided by the elements of common ground and situational context proposes a possible solution to these difficulties. Subsequently, we detail experiments we have been conducting in VoxWorld, the situated grounding platform based on the VoxML modeling language (Pustejovsky & Krishnaswamy, 2016). These experiments combine neural learning and symbolic reasoning approaches to address affordance learning, structure learning, and transfer learning for an intelligent agent.

3. Modeling Context

Object properties and the events they facilitate (Gibson, 1977) are a primary component of situational context. Gibson's initial formulation of affordances vaguely defines the term as what the environment "offers the animal." Gibson refers to the term as "something that refers to both the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment" (Gibson, 1979).

We use the term in our work in a way that attempts to cover the extensive ground that Gibson uses it for, while maintaining a clear relation between the environment (including object configuration as a positioning, or *habitat*), the properties of an object that allows it to be used for certain behaviors (e.g., the "graspability" of a handle), and the language used to describe these behaviors and ground them to an environment or situation, as has been explored in recent neural AI work (e.g., Hermann et al. (2017); Das et al. (2017)),

For instance, in Fig. 2, the cup on the left is in a position to be *slid* across its supporting surface while the cup on the right is in a position to be *rolled*. Executing one or the other of these actions would require the cup to be placed in a prerequisite orientation, and may result in concomitant effects, such as anything contained in the cup spilling out (or not).

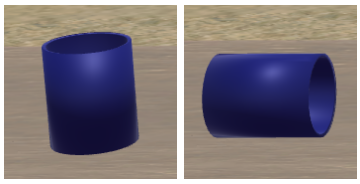

Figure 2: Cups in different habitats.

We correlate these *afforded* behaviors (a la Gibson, and Pustejovsky (1995)'s *telic* roles) with the notion of habitats (Pustejovsky, 2013; McDonald & Pustejovsky, 2013), or conditioning environments that facilitate affordances. Formally, we capture these properties in the modeling language VoxML (Pustejovsky & Krishnaswamy, 2016), that captures common object and event semantics, with a particular focus on habitats and affordances. VoxML models ontological information that is difficult to learn from corpora due to being so common that it is rarely documented and therefore not available to machine learning algorithms. VoxML provides the format for the symbolic encodings of our neurosymbolic pipeline. Each component of a VoxML encoding (e.g., object shape, event semantic class, individual habitat, affordance, etc.) can be hand-encoded, extracted from corpora, or learned, providing a way to habituate common qualitative knowledge into a structured but flexible representation. This qualitative knowledge is important to reflect human-like qualitative

reasoning capabilities in a computational context. When reasoning about a ball rolling, humans do not need to know the exact value of parameters like speed or direction of motion, but to simulate the event computationally, every variable must have a value for the program to run. VoxML provides a structured encoding of properties for these variables that allows a system to generate values when needed. Fig. 3 shows the VoxML encoding for a cup. Note the intrinsic upward orientation of the habitat $H_{[3]}$ where the cup's Y-axis is aligned with that of the world, and the afforded behaviors that may be conditioned on a particular habitat, or may be available in any habitat (denoted $H\ \rightarrow$).

$$
\left[\begin{array}{l}
\textbf{cup} \\
\text{LEX} = \left[\begin{array}{l} \text{PRED} = \textbf{cup} \\ \text{TYPE} = \textbf{physobj•artifact} \end{array}\right] \\
\text{TYPE} = \left[\begin{array}{l} \text{HEAD} = \textbf{cylindroid[1]} \\ \text{COMPONENTS} = \textbf{surface[1], interior[2]} \\ \text{CONCAVITY} = \textbf{concave[2]} \\ \text{ROTAT\_SYM} = \textbf{Y} \\ \text{REFL\_SYM} = \textbf{XY, YZ} \end{array}\right] \\
\text{HABITAT} = \left[\begin{array}{l} \text{INTR} = {}_{[3]}\left[\begin{array}{l} \text{UP} = align(\bar{Y}, \mathcal{E}_Y) \\ \text{TOP} = top(+Y) \end{array}\right] \\ \text{EXTR} = \left[\begin{array}{l} {}_{[4]}\left[\begin{array}{l} \text{UP} = align(\bar{Y}, \mathcal{E}_Y) \\ \text{TOP} = top(-Y) \end{array}\right] \\ {}_{[5]}\left[\text{UP} = align(\bar{Y}, \mathcal{E}_{\perp Y})\right] \end{array}\right] \end{array}\right] \\
\text{AFFORD\_STR} = \left[\begin{array}{l} \text{A}_1 = H \rightarrow [put(x, y, on([1]))]support([1], y) \\ \text{A}_2 = H_{[3]} \rightarrow [put(x, y, in([2]))]contain([2], y) \\ \text{A}_3 = H \rightarrow [grasp(x, [1])]hold(x, [1]) \\ \text{A}_4 = H \rightarrow [lift(x, [1])]hold(x, [1]) \\ \text{A}_5 = H \rightarrow [ungrasp(x, [1])]release(x, [1]) \\ \text{A}_6 = H_{[3,4]} \rightarrow [slide(x, [1])]\mathcal{R} \\ \text{A}_7 = H_{[5]} \rightarrow [roll(x, [1])]\mathcal{R} \\ \textbf{...} \end{array}\right] \\
\text{EMBODIMENT} = \left[\begin{array}{l} \text{SCALE} = <\textbf{agent} \\ \text{MOVABLE} = \textbf{true} \end{array}\right]
\end{array}\right]
$$

Figure 3: VoxML encoding for a [[CUP]] voxeme.

3.1 Multimodal Simulations

The situated, simulated environments of the VoxWorld platform bring together three notions of simulation from computer science and cognitive science (Pustejovsky & Krishnaswamy, 2019):

1. *Computational simulation modeling*. That is, variables in a model are set and the model is run, such that the consequences of all possible computable configurations become known. Examples of such simulations include models of climate change, the tensile strength of materials, models of biological pathways, and so on. The goal is to arrive at the best model by using simulation techniques.

2. *Situated embodied simulations*, where the agent is embodied with a dynamic point-of-view or avatar in a virtual or simulated world. Such simulations are used for training humans in scenarios such as flight simulators or combat situations, and of course are used in video gaming as well. In these contexts, the virtual worlds assume an embodiment of the agent in the environment, either as a first-person restricted POV or an omniscient movable embodied perspective. The goal is to simulate an agent operating within a situation.

3. *Embodied theories of mind*. Craik (1943) and, later, Johnson-Laird (1987) develop the notion that agents carry a mental model of external reality in their heads. Johnson-Laird & Byrne (2002) represent this model as a situational possibility, capturing what is common to different ways the situation may occur. Simulation Theory in philosophy of mind focuses on the role of "mind reading" in modeling the representations and communications of other agents (Gordon, 1986; Goldman, 1989; Heal, 1996; Goldman, 2006). Simulation semantics (as adopted within cognitive linguistics and practiced by Feldman (2010), Narayanan (2010), Bergen (2012), and Evans (2013)) argues that language comprehension is accomplished by such mind reading operations. There is also an established body of work within psychology arguing for *mental simulations* of future or possible outcomes, as well of perceptual input (Graesser et al., 1994; Barsalou, 1999; Zwaan & Radvansky, 1998; Zwaan & Pecher, 2012). The goal is semantic interpretation of an expression by means of a simulation, which is either mental (a la Bergen and Evans) or interpreted graphs such as Petri Nets (a la Narayanan and Feldman).

Krishnaswamy (2017) brings the model testing of (1), the situated embodiment of (2), and the modeling machinery of (3) together into Monte-Carlo visual simulation of underspecified motion predicates, which forms the backbone of a situated approach to learning and language understanding. Given a label (symbol) of a motion verb, there may be a large space of potential specific instantiations of that motion that satisfy the label. The specifics may depend on the objects involved, and may contain many underspecified variable values (e.g., speed of motion, exact path—depending on the verb, etc.).

In an interaction, as in Fig. 1, each agent maintains their own model of what the other agent knows, including respective interpretations of vocabulary items. For instance, if the mother says "pass me that plate" and the son throws it at her, it becomes clear to her that his interpretation of "pass" differs from hers. Since the computer system operationalizes all these motion predicates in terms of primitive motions like *translate* and *rotate*, it needs a model that accommodates flexible representations of these primitive motions and of their composition into more complex motions.

The Monte-Carlo simulation approach of VoxWorld provides the model on which to operationalize these complex motion predicates in ways that behave according to the preconceived notions of a typical human user. Given an input (a simple event description in English), the input is parsed and broken out into VoxML representations of the objects, events, and relations involved. These individual structured representations are then *recomposed*. From that recomposition, the variables of the composed representation that remain unassigned are extracted as the underspecified features.

The VoxML- and Unity-based VoxSim software (Krishnaswamy & Pustejovsky, 2016b) was then used to generate over 35,000 animated visualizations of a variety of common motion events (put, slide, lift, roll, lean, etc.) with a vocabulary of common objects (cups, pencils, plates, books,

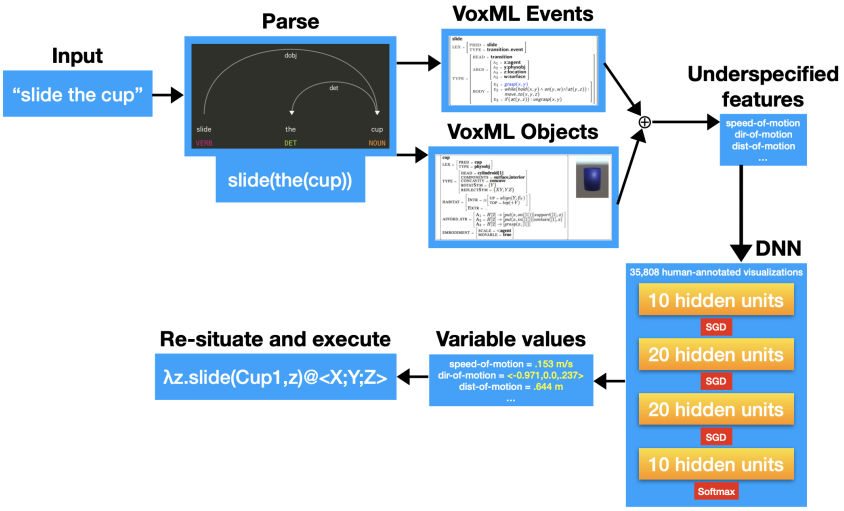

Figure 4: Neurosymbolic pipeline for generating multimodal simulations.

etc.), that displayed a wide variety of underspecified variables in their respective operationalizations. Each visualization was given to 8 annotators each, along with two other variant visualizations of the same input event, and the annotators were asked to choose the best one, as well as to choose the best event caption for each visualization[3]. We then extracted the range of values assigned to underspecified parameters in those visualizations which annotators judged appropriate, and used a feedforward deep neural network (DNN) to predict the best values for underspecified parameters given an event input in plain English. When given an input text, VoxSim runs the underspecified parameter symbols through the model, and the resultant output values are assigned to the relevant input parameters, resituated in the scene, and executed in real time to create an appropriate visualization of the input event. Fig. 5 shows the resulting state for one such visualization for "lean the cup on the book".

This pipeline is shown in Fig. 4 and serves as the basis for interactively exploring learning and reasoning through situated grounding.

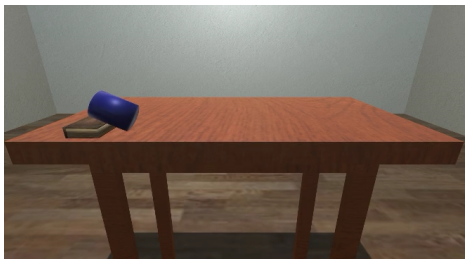

Figure 5: Visualization of "lean the cup on the book."

4. Interactive Learning of Object Affordances

Underspecified parameters in a predicate can also be inferred from the properties of objects, namely the habitats which they can occupy and the behaviors afforded by them. For instance, if a cup is both *concave* and symmet-

[3]Data is available at https://github.com/nkrishnaswamy/underspecification-tests

ric around the *Y-axis*, then there is no need to explicitly specify the orientation of the concavity; we can infer that it is aligned with the object's Y-axis, and this in turn requires that certain conditions (habitats) be enforced for certain behaviors (affordances) to be taken advantage of, such as putting something in the cup, or grasping the cup appropriately in order to drink from it (Krishnaswamy & Pustejovsky, 2016a).

Previously, we relied on hand-crafted object affordance encodings in VoxML (e.g., see Fig. 3), but this is an inefficient process and hard to scale. To automatically explore affordances such as *grasping*, a system must have an agent capable of grasping items, namely an *embodied, situated agent* that explores situated grounding from its own dynamic point of view. In Krishnaswamy et al. (2017) and Narayana et al. (2018), we examined the problem of situatedness and communication within a situated context in an encounter between two "people": an avatar modeling multimodal dialogue with a human.

Figure 6: Diana interacting with a human.

Our agent in VoxWorld, known as Diana, is situated in a virtual VoxSim environment, consumes input from 3rd party or custom speech recognition, and can see her human interlocutor's gestures with custom recognition algorithms running on deep convolutional neural networks trained on over 8 hours of annotated video and depth data from a Microsoft Kinect™. The human can gesture about objects in Diana's virtual world, making Diana an interactive collaborator.[4]

A major challenge for collaborative AI is data availability for different tasks. Nonetheless, we can use the situated grounding environment of VoxWorld and the interactive mechanisms enabled by an agent like Diana to access detailed, multimodal data for a variety of tasks. Diana's default vocabulary of 34 gestures contains a downward-opening "claw" gesture used to mean *grasp*. This gesture is sufficient to signal grasping an object like a block. However, in Diana's "kitchen world" scenario, containing common household objects, she comes across items, like plates, that cannot be grasped in this way. In that case, she must estimate positions on the object where it is graspable.

Grasp point inference uses the symmetry of objects as encoded in VoxML. Objects have rotational and reflectional symmetry, such that a cup has rotational symmetry around its Y-axis and reflectional symmetry across its XY- and YZ-planes, while a knife has only reflectional symmetry across its YZ-plane in default orientation.

[4] A full-length video demo of Diana can be viewed here.

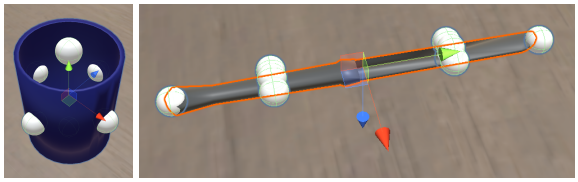

Figure 7: Grasp points on a cup and knife.

For objects with rotational symmetry, we find all points P on the surface equidistant from the extremes along the axis of symmetry, as well as the extreme points of the object along that axis. For objects without rotational symmetry, we find those points P on each component of the object that intersect the plane(s) perpendicular to the plane of reflectional symmetry (see Fig. 7). The closest one of these points to the position of the agent's hand (w) is taken to be the targeted point of interaction with the object.

From this point we calculate the maximum amount each finger (f) can bend toward the object without intersecting its bounds, take this distance of the fingers to wrist joint and add it to/subtract it from the object's extremities in both directions along all 3D major axes ($extents_{x,y,z}$).

$$\forall f \in F \underset{m\angle fwp}{\operatorname{argmin}} \mid \vec{fw} \mid \times \cos(m\angle fwp) \pm extents_{x,y,z}$$

The angle of the fingers at this point becomes a *grasp pose*, or hand position needed to activate the *grasping* affordance (that is, to exploit the *graspability*) of that object at that position. Each possible grasp pose comes with an associated affordance semantics of the form $\lambda y.grasp(obj, with(y))$ where y represents the associated iconic gesture, which is as yet unfilled. Diana then demonstrates each inferred possible grasp pose to the human and asks for a preference:

HUMAN: The plate.
DIANA: Okay. [DIANA *points to the plate.*]

[HUMAN *makes "claw down" gesture.*]
DIANA: Should I grasp it like this? [DIANA *grasps plate from beneath.*]

[HUMAN *makes "thumbs down" gesture.*]
DIANA: How about like this? [DIANA *grasps plate from side.*]
HUMAN: Yes.
DIANA: **Is there a gesture for that?**

[**HUMAN *makes "grasp plate" gesture.***]

Figure 8: Iconic *plate* gesture for "grasp the plate."

Now, with the affordance semantics available to be filled, the visual features of the novel gesture the human makes are fed into a random forest classifier trained over 2048-dimensional feature vectors derived from the annotated video data used to train the default gesture recognizer. The novel gesture is situated in the feature space of the 34 known gestures (plus any novel gestures previously learned). That new vector value is applied to the outstanding variable in the affordance semantics generated through the interaction to this

point. The result represents an operationalization of $grasp(x)$ where x is the object requiring novel exploitation of its affordances to grasp it. This operationalized predicate is then propagated down to any other events that use [[GRASP]] as a subevent over the object x. This now allows the human to instruct the agent to grasp an object using the correct pose, with a single visual cue, as in Fig. 8. Furthermore, the avatar can subsequently be instructed to perform any actions that subsume grasping a plate.

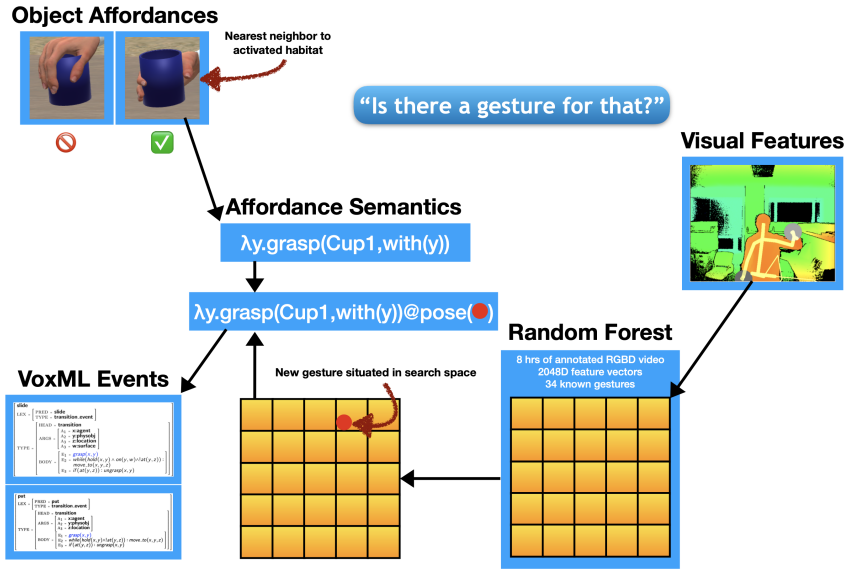

Figure 9: Neurosymbolic pipeline for learning interactions with object affordances (Sec. 4).

Fig. 9 gives the neurosymbolic learning pipeline for object affordances and accompanying actions. Because the learned object affordance is propagated down to other events that contain the associated action, we can fill in other action sequences with this association using a continuation-passing style semantics (Krishnaswamy & Pustejovsky, 2019a). For example, extending the dynamic event structure developed in Pustejovsky & Moszkowicz (2011), the VoxML encoding of the event [[SLIDE_TO]] can be represented as in (1). This is a derived event composed from the activity [[SLIDE]] and the directional PP [[TO_LOC]] (Pustejovsky & Krishnaswamy, 2014).

(1) $grasp(e_1, \text{AG}, y)$; **while**$(hold(\text{AG}, y) \wedge on(y, \text{SURF}) \wedge \neg at(y, \text{LOC})), move\_to(e_2, \text{AG}, y, \text{LOC}))$; **if**$(at(y, \text{LOC}), ungrasp(e_3, \text{AG}, y))$

Therefore, if the agent encounters a [[SLIDE]] action with an outstanding variable ($\lambda y.slide(y, loc)$), and the human supplies a gesture denoting $grasp(plate)$, then the agent can directly lift $grasp(plate)$ to the slide action and apply the argument $plate$ to y: $\lambda y.slide(y, loc)@plate \Rightarrow slide(plate, loc)$. **while**$(C, A)$ states that an activity, A, is performed only if a constraint, C, is satisfied at the same moment. Here, should something cause the agent to drop the object or should the agent lift the object off the surface, the constraint, and therefore the overall [[SLIDE]] action will cease and remain incomplete.

5. Learning Structure and Novel Configurations

Configurations can be instantiated by exploiting an object's affordances to create relations between it and other objects. For instance, exploiting the containment affordance of a cup (i.e., the structural properties of a cup that allow it to contain other items) by putting a spoon in it results in a "spoon-in-cup" configuration that can be realized in multiple ways under constraints (e.g., handle-up or handle-down, but not typically lying flat). In this sense, complex configurations are the result of *composed affordances*.

This is a far more complex problem than learning affordances over single objects. Naively, affordance composition of k objects with m affordances each runs in $O(m^k)$ time due to the need to check every enumerated affordance of an object against every enumerated affordance of every other object in an Allen-style composition table (Allen, 1983), so for training and testing a model we back off to a simple Blocks World domain in order to reuse all affordances across all objects. Even so, we encounter two problems:

1. Different configurations imply different sets of constraints. For example, when laying a conventional place setting at a table, the plate goes in the center; putting it anywhere else creates something that is not a "place setting." When filling a moving truck with boxes or building a staircase out of blocks, there is simply a space to be filled or a configuration to be created, in lieu of very specific object placements.

2. Representative data is required for training. The one-shot gesture learning featured in Sec. 4 uses random forests over 2048-dimensional feature vectors trained over annotated RGBD video data, but no such equivalent data exists for this affordance composition/configuration problem, forcing reliance on smaller data sources and concomitant algorithms.

Krishnaswamy & Pustejovsky (2018) presents results of a user study wherein 20 naive users interacted with the Diana system. They were instructed to build a three-step staircase out of six equal-sized, uniquely-colored blocks, and told that the system could understand gestures and spoken English, but were not given a specific vocabulary. When the user was satisfied with the results, the task was considered complete. Three users failed to complete the task in the allotted time, so the study produced 17 sample staircases of diverse configurations wherein the placements of individual blocks varied, stacks of blocks were not perfectly aligned, and staircases pointed in either applicable direction (see Fig. 10). Therefore, any learning algorithm must be able to infer commonalities across this small and noisy sample. This learning procedure is detailed in Krishnaswamy et al. (2019) and can be analogized to common parts of a neural NLP pipeline using symbolic representations extracted from the situated simulated environment.

The data we gathered is sparse enough that machine learning over the raw coordinates of blocks will fail to converge. Therefore we made use of situated grounding to extract *qualitative* relations between objects. We used subsets of the Region Connection Calculus RCC8 (Randell et al., 1992), and the Ternary Point Configuration Calculus (Moratz et al., 2002) with

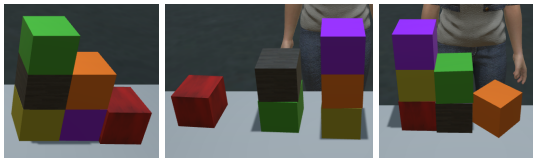

Figure 10: Krishnaswamy & Pustejovsky (2018): sample user-constructed staircases.

implementations taken from QSRLib (Gatsoulis et al., 2016), which allow grounded, denser representations of the relation sets that make up a structure or configuration.

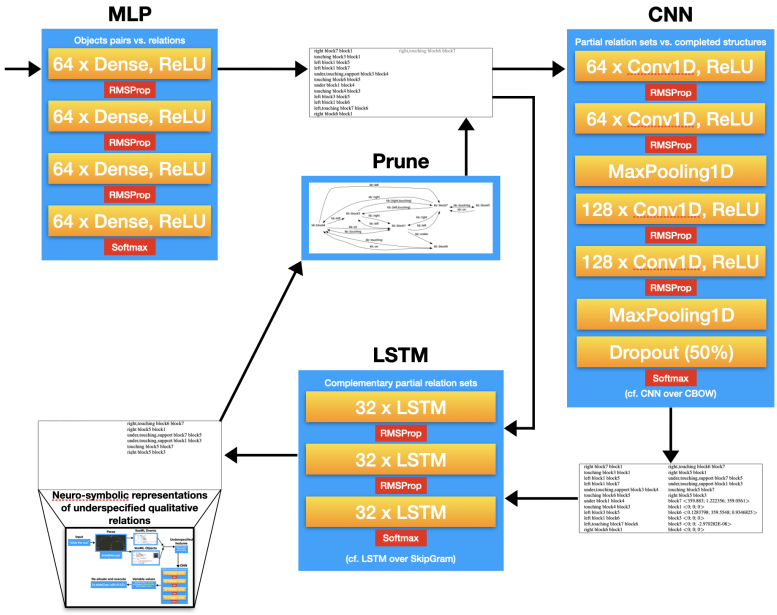

Figure 11: Neurosymbolic pipeline for learning structural configurations.

Fig. 11 shows the neurosymbolic learning pipeline for structures and configurations given situated grounding. To sample from the training data and to avoid artificially introducing some of the inferences the learning pipeline is intended to pick up itself, the first move is generated using a multi-layer perceptron that takes in a random pair of objects and outputs a relation to create between them. This generates a partial relation set that is fed into the relation-classifying CNN. The CNN should output a known, completed structure that contains that partial relation set. This can be analogized to a continuous bag-of-words model over a vocabulary of relations. This output and the partial relation set representing the current state are fed into the LSTM network, which outputs a set of move options that would complement the current partial relation set and be most likely to complete the structure predicted by the CNN. This is similar to a sequence construction task or an LSTM trained over a Skip-Gram model. Each move option presented by the LSTM is a symbolic representation of a qualitative spatial relation to be created between two blocks. As such, the specifics of where to place the blocks in Cartesian coordinate space must be filled in somehow, and so the operationalization can then use the pipeline for generating multimodal simulations (Fig. 4) to complete this process. The move options are then pruned to select a single best move that is then enacted, adding relations to the current state's partial set. This process repeats until the objects are used up.

5.1 Validation

The CNN and LSTM portions of the learning framework above rely on a notable assumption: that when only a few relations are instantiated, a known example and complementary relation set will be almost random but as more relations are instantiated, the prediction of both neural networks becomes more accurate.

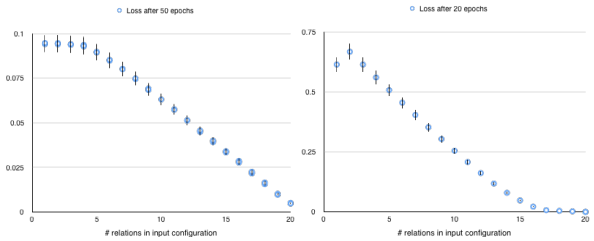

Figure 12: CNN (L)/LSTM (R) loss vs. input length.

To validate this assumption, we sampled increasingly large relation sets from the training data, fed them through both networks, and measured the cross-entropy loss after 50 epochs (for the CNN) and 20 epochs (for the LSTM). As expected (Fig. 12), the prediction of both nets becomes more accurate as relation input size increases, and approaches almost 0 once 20 relations are instantiated, representing, in most cases, a complete structure from the training data.

5.2 Results

Fig. 13 shows sample generated staircases displaying desired inferences. To prune the output of the two stacked neural nets to choose a single move to execute, we assessed 5 heuristic loss functions: random chance as a baseline, Jaccard distance to measure the shared presence or absence of a spatial relation (Jaccard, 1912), Levenshtein distance to measure the *count* of shared spatial relations (Levenshtein, 1966), a graph-matching algorithm called SPIRE (McLure et al., 2015), and a Levenshtein distance-pruned version of SPIRE. We generated 50 novel instances of staircases in total; 5 from each trained model, and had a set of evaluators rate them all from 0-10, based on the question "How much does the structure shown resemble a staircase?" Images were viewed in random order to minimize sequential biases across evaluators. SPIRE was the most successful heuristic loss function; the average evaluator score for staircases generated using SPIRE was **5.8313**, compared to **2.0375** for random chance, and **4.7188** for the next-highest-performing heuristic, the Levenshtein distance-pruned graph matcher.

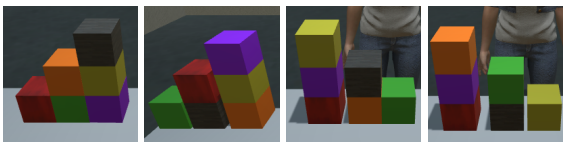

Figure 13: Sample generated staircases.

Situated grounding within a multimodal simulation provides methods for inferring the composition of structures and configurations from small sample sizes, specifically the ability to translate from specific coordinates to qualitative relations and back within a neurosymbolic model. This addresses the *generation* aspect of novel concept learning for an AI agent.

5.3 Semantic Grounding

Supplementary to generation is the ability to recognize and classify instances of novel concepts. Within VoxWorld this is addressed as a constraint satisfaction problem. Since we are primarily concerned with learning the habitats within which a novel structural configuration exists, and the

afforded behaviors that those habitats enable, we provide the system with the components that make up a novel structure, and allow it to infer what affordances of those components are being used in the novel structure, and which satisfy the constraints in the learned and generated samples.

In Krishnaswamy & Pustejovsky (2019b) this is approached from many angles, including weighted constraint satisfaction or as a partially-observable Markov decision process (cf. Lee et al. (2018)). One promising avenue, using both the advantages of qualitative representation within situated grounding and mapping cleanly to the learning pipeline described above, is a qualitative constraint network (QCN). We use a variant of the CONACQ.2 algorithm outlined in Mouhoub et al. (2018) where the language consists of spatial relations from the training data and generated examples (replacing Allen Temporal Relations (Allen, 1983)), the background knowledge is transitive closure rules over the relation vocabulary, the learning strategy is the pipeline outlined above, and the result is encoded in VoxML.

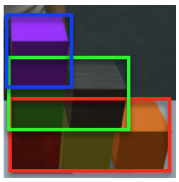

Figure 14: Staircase components.

input : Relation language $\mathcal{B}$, background knowledge K, strategy $\mathcal{L}$
output: Acquired constraint set $\mathcal{S}$
$\mathcal{S} \leftarrow \{\}$; $converged \leftarrow false$;
while $\neg converged$ **do**
  $q \leftarrow QueryGeneration(\mathcal{B}, \mathcal{S}, K, \mathcal{L})$;
  if $q = nil$ **then**
    $converged \leftarrow true$;
  else
    if $Answer(q) > threshold$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \wedge \bigwedge_{c \in K(q)} c \vdash \mathcal{S}$;
    else
      $\mathcal{S} \leftarrow \mathcal{S} \wedge (\bigvee_{c \in K(q)} c \nvdash \mathcal{S})$;
    end
  end
end
return $\mathcal{S}$;

Algorithm 1: Adapted CONACQ.2 for habitat and affordance inference.

In the query generation phase of Algorithm 1, we back off to the LSTM used in step 3 of the learning strategy $\mathcal{L}$ (Fig. 11), and make use of the attention vectors to generate the queries. Thereafter, if the answer to the query is $true$ above a given threshold in the example under examination, it is added to the constraint set, otherwise it is removed if present.

Fig. 15 shows a sample result, and that we can successfully extract certain constraints that describe not just the staircase shown, but an abstract staircase. These constraints include: the steps ascending to either the left *or* right (A_3, A_4), that placing an object on the base or step creates a new (non-base or -step) tier (A_3, A_4), or that putting something on the base makes it part of the step (A_2). This provides at least some of the components of a semantic model for the new object. When asked about a "staircase," the agent now has semantics for a decontextualized reference that can be reproduced and adjusted without having to retrain the purely numerical model. Thus situ-

$$\left[\begin{array}{l}\textbf{staircase}\\ \text{TYPE} = \left[\begin{array}{l}\text{HEAD} = \textbf{assembly[1]}\\ \text{COMPONENTS} = \textbf{base[2],step[3]*,top[4]}\end{array}\right]\\ \text{HABITAT} = \left[\begin{array}{l}\text{INTR} = {}_{[5]}\left[\begin{array}{l}\text{BASE} = align([2], \mathcal{E}_X)\\ \text{UP} = align(vec(loc([4]) - loc([2])), \mathcal{E}_Y)\end{array}\right]\end{array}\right]\\ \text{AFFORD\_STR} = \left[\begin{array}{l}\text{A}_1 = H_{[5]} \rightarrow [put(x, on([1]))]part\_of(x, [1])\\ \text{A}_2 = H_{[5]} \rightarrow [put(x, on([2]))]part\_of(x, [3])\\ \text{A}_3 = H_{[5]} \rightarrow [put(x, left \vee right \vee\\ \quad touching([2]) \wedge \neg on([2])]extend(x, [2])\\ \text{A}_4 = H_{[5]} \rightarrow [put(x, left \vee right \vee\\ \quad touching([3]) \wedge \neg on([3])]extend(x, [3])\end{array}\right]\end{array}\right]$$

Figure 15: Acquired constraints describing a staircase.

ated grounding allows probing of models in a tractable way by examining qualitative relations and acquired constraints.

6. Transfer Learning of Object Properties and Linguistic Description

Through correlating cross-modal representations (e.g., the visual features of a gesture and an embodied action in 3D space, or the composition of qualitative relations and a structural label or its semantics), or correlating cached symbols with neural representations (e.g., a relation or action label and a set of specifically quantitative operationalizations), situated grounding serves as a platform for improving sample efficiency through reuse. Therefore, it should also facilitate transferring knowledge gained from solving one problem and applying it to another situation. Situatedness is particularly useful for transfer learning, because similar concepts often exist in similar situations (cf. analogical generalization, a la Forbus et al. (2017)).

Associating affordances with abstract properties or symbolic labels informs the way that the entities that possess those affordances can be discussed. In Sec. 1, we discussed the inability of unimodal language understanding systems to answer the simple question "what am I pointing at?" While situated grounding provides a solution to linking linguistic terms to entities sharing the agent's co-situated space, the agent can still only discuss these entities if she knows the appropriate terms for them. If an agent encounters a new object that she doesn't know the name of, she can discuss it in terms of "this one" or "that one," but cannot decontextualize the reference with a lexical label. Transfer learning provides a way to give the agent a way of talking about a novel item by comparing it to known items.

We focus here on transfer learning of affordances. Since similar objects typically have similar habitats and affordances (e.g., cylindrical items with concavities often serve as containers), it is worth investigating whether such properties can be transferred from known objects to novel objects that are observed to have similar associated properties. The method we use is termed *affordance embedding*. This follows an intuition similar to the Skip-Gram model in natural language processing (Mikolov et al., 2013), or the masked language model of BERT (Devlin et al., 2018), but exploits the linkage between affordances and objects present in a situated grounding model like VoxWorld.

We are experimenting with two architectures: a 7-layer MLP and a 4-layer CNN with 1D convolutions, both trained over habitat-affordance pairs taken from our vocabulary of known objects. For instance, a habitat-affordance pair for a [[CUP]] voxeme might be ($H_{[2]}$ = [UP = $align(Y, \mathcal{E}_Y)$, TOP = $top(+Y)$], $H_{[2]} \rightarrow [put(x, in(this))]contain(this, x)$) (gloss: *the cup's Y-axis is aligned upward with the Y-axis of the embedding space, and if something is put inside the cup, the cup contains that thing*). The network outputs collocation probabilities for every individual habitat-affordance pair in the vocabulary. Subsequently, for each possible action the agent may take with an object (e.g., grasp, pick up, move, slide, put on, etc.), the system queries the learned affordance embeddings, *excluding* those affordances that include the particular action in question. This restates the answer to a query, e.g., "describe the appropriate habitat for *grasping* an object" in terms of *other* actions that can be taken in that habitat, and the habitat is matched to other objects that share that habitat. This is effectively a second-order collocation.

For example, if the agent comes across an unfamiliar object that appears to share the $H_{[2]}$ = [UP = $align(Y, \mathcal{E}_Y)$, TOP = $top(+Y)$] (upward alignment) habitat of [[CUP]], she can infer that it might be grasped in a similar way. Fig. 16 shows this process enacted through dialogue. In frame 1, the human points to a new object (recognizable as a bottle, but Diana has no label associated with it). In frame 2, Diana says "I don't know"—reflecting the semantic gap in her vocabulary—"but I can grasp it like a cup"—reflecting the information about it that she is able to infer from its habitats and affordances, which gives her a way to talk about this object with her human partner. In frame 3, the human says "grab it," and Diana demonstrates her inferred method of grasping, using grasp pose calculation as described in Sec. 4.

Figure 16: 1) Human: "What is that?"; 2) Diana: "I don't know, but I can grasp it like a cup."; 3) Human: "Grab it." + resultant grasp.

7. Conclusions

In this paper, we hope to have demonstrated that the notion of situatedness goes well beyond visually grounding a text or a concept to an image or video; rather, it involves embedding the linguistic expression and its grounding within a multimodal semantics. This approach provides for environmentally-aware models that can be validated; each additional modality provides an orthogonal angle through which to validate models of other modalities. It provides many methods of encoding context both quantitatively and qualitatively, and provides a model to accommodate both neural and symbolic representations. The diverse types of data available through a situated ground-

ing platform are adaptable to different tasks with novel types of network architectures, with less data overhead than typical machine learning. As such, in an era where mainstream AI tends toward increasingly large datasets and bigger models involving more and more parameters, with concomitant costs in energy and resource usage, such a platform provides a sustainable way toward more powerful AI.

Acknowledgements

We would like to thank our collaborators at Colorado State University and the University of Florida for their longtime collaboration on developing the Diana interactive agent. We would also like to thank the reviewers for their helpful comments. This work was supported by the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO) under contract #W911NF-15-C-0238 at Brandeis University. The points of view expressed herein are solely those of the authors and do not represent the views of the Department of Defense or the United States Government. Any errors or omissions are, of course, the responsibility of the authors.

References

Abbott, B. (2008). Presuppositions and common ground. *Linguistics and Philosophy*, *31*, 523–538.

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, *26*, 832–843.

Asher, N. (1998). Common ground, corrections and coordination. *Journal of Semantics*.

Barsalou, L. W. (1999). Perceptions of perceptual symbols. *Behavioral and brain sciences*, *22*, 637–660.

Bergen, B. K. (2012). *Louder than words: The new science of how the mind makes meaning*. Basic Books.

Besold, T. R., et al. (2017). Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.

Clark, H. H., Schreuder, R., & Buttrick, S. (1983). Common ground at the understanding of demonstrative reference. *Journal of verbal learning and verbal behavior*, *22*, 245–258.

Craik, K. J. W. (1943). *The nature of explanation*. Cambridge University, Cambridge UK.

Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., & Batra, D. (2017). Embodied question answering. *arXiv preprint arXiv:1711.11543*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dzifcak, J., Scheutz, M., Baral, C., & Schermerhorn, P. (2009). What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. *International Conference on Robotics and Automation* (pp. 4163–4168). IEEE.

Evans, V. (2013). *Language and time: A cognitive linguistics approach*. Cambridge University Press.

Feldman, J. (2010). Embodied language, best-fit analysis, and formal compositionality. *Physics of life reviews*, *7*, 385–410.

Forbus, K. D., Ferguson, R. W., Lovett, A., & Gentner, D. (2017). Extending sme to handle large-scale cognitive modeling. *Cognitive Science*, *41*, 1152–1201.

Garcez, A. d., Gori, M., Lamb, L. C., Serafini, L., Spranger, M., & Tran, S. N. (2019). Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*.

Gatsoulis, Y., et al. (2016). Qsrlib: a software library for online acquisition of qualitative spatial relations from video.

Gibson, J. J. (1977). The theory of affordances. *Perceiving, Acting, and Knowing: Toward an ecological psychology*, (pp. 67–82).

Gibson, J. J. (1979). *The ecology approach to visual perception: Classic edition*. Psychology Press.

Goldman, A. I. (1989). Interpretation psychologized*. *Mind & Language*, *4*, 161–185.

Goldman, A. I. (2006). *Simulating minds: The philosophy, psychology, and neuroscience of mindreading*. Oxford University Press.

Gordon, R. M. (1986). Folk psychology as simulation. *Mind & Language*, *1*, 158–171.

Graesser, A. C., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological review*, *101*, 371.

Heal, J. (1996). Simulation, theory, and content. *Theories of theories of mind*, (pp. 75–89).

Hermann, K. M., et al. (2017). Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*.

Iqbal, T., & Qureshi, S. (2020). The survey: Text generation models in deep learning. *Journal of King Saud University-Computer and Information Sciences*.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New phytologist*, *11*, 37–50.

Johnson-Laird, P. (1987). How could consciousness arise from the computations of the brain. *Mindwaves. Oxford: Basil Blackwell*, (pp. 247–257).

Johnson-Laird, P. N., & Byrne, R. M. (2002). Conditionals: a theory of meaning, pragmatics, and inference. *Psychological review*, *109*, 646.

Krishnaswamy, N. (2017). *Monte-carlo simulation generation through operationalization of spatial primitives*. Doctoral dissertation, Brandeis University.

Krishnaswamy, N., Friedman, S., & Pustejovsky, J. (2019). Combining Deep Learning and Qualitative Spatial Reasoning to Learn Complex Structures from Sparse Examples with Noise. *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI.

Krishnaswamy, N., & Pustejovsky, J. (2016a). Multimodal semantic simulations of linguistically underspecified motion events. *Spatial Cognition X: International Conference on Spatial Cogni-*

tion. Springer.

Krishnaswamy, N., & Pustejovsky, J. (2016b). VoxSim: A visual platform for modeling motion language. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. ACL.

Krishnaswamy, N., & Pustejovsky, J. (2018). An evaluation framework for multimodal interaction. *Proceedings of LREC*.

Krishnaswamy, N., & Pustejovsky, J. (2019a). Multimodal continuation-style architectures for human-robot interaction. *arXiv preprint arXiv:1909.08161*.

Krishnaswamy, N., & Pustejovsky, J. (2019b). Situated grounding facilitates multimodal concept learning for ai. *Workshop on Visually Grounded Interaction and Language*.

Krishnaswamy, N., et al. (2017). Communicating and acting: Understanding gesture in simulation semantics. *12th International Workshop on Computational Semantics*.

Lee, J., Kim, G.-H., Poupart, P., & Kim, K.-E. (2018). Monte-carlo tree search for constrained pomdps. *Advances in Neural Information Processing Systems* (pp. 7923–7932).

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady* (pp. 707–710).

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., & Chang, K.-W. (2019). Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Liu, J., Kong, X., Xia, F., Bai, X., Wang, L., Qing, Q., & Lee, I. (2018). Artificial intelligence in the 21st century. *IEEE Access*, *6*, 34403–34421.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., & Wu, J. (2019). The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

Marcus, G., & Davis, E. (2019). *Rebooting ai: building artificial intelligence we can trust*. Pantheon.

Matuszek, C. (2018). Grounded language learning: Where robotics and nlp meet. *IJCAI* (pp. 5687–5691).

McCarthy, J. (2007). From here to human-level ai. *Artificial Intelligence*, *171*, 1174–1182.

McDonald, D., & Pustejovsky, J. (2013). On the representation of inferences and their lexicalization. *Proceedings of the Second Annual Conference on Advances in Cognitive Systems ACS* (p. 152). Citeseer.

McLure, M. D., Friedman, S. E., & Forbus, K. D. (2015). Extending analogical generalization with near-misses. *AAAI* (pp. 565–571).

McNeely-White, D. G., et al. (2019). User-aware shared perception for embodied agents. *2019 IEEE International Conference on Humanized Computing and Communication (HCC)* (pp. 46–51). IEEE.

Menzies, T. (2003). Guest editor's introduction: 21st century ai–proud, not smug. *IEEE Intelligent Systems*, (pp. 18–24).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Moratz, R., Nebel, B., & Freksa, C. (2002). Qualitative spatial reasoning about relative position. *International Conference on Spatial Cognition* (pp. 385–400). Springer.

Mouhoub, M., Al Marri, H., & Alanazi, E. (2018). Learning qualitative constraint networks. *25th International Symposium on Temporal Representation and Reasoning (TIME 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Narayana, P., et al. (2018). Cooperating with avatars through gesture, language and action. *Intelligent Systems Conference (IntelliSys)*.

Narayanan, S. (2010). Mind changes: A simulation semantics account of counterfactuals. *Cognitive Science*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pustejovsky, J. (1995). *The generative lexicon*. Cambridge, MA: MIT Press.

Pustejovsky, J. (2013). Dynamic event structure and habitat theory. *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)* (pp. 1–10). ACL.

Pustejovsky, J. (2018). From actions to events: Communicating through language and gesture. *Interaction Studies*, *19*, 289–317.

Pustejovsky, J., & Batiukova, O. (2019). *The Lexicon*. Cambridge University Press.

Pustejovsky, J., & Krishnaswamy, N. (2014). Generating simulations of motion events from verbal descriptions. *Lexical and Computational Semantics (* SEM 2014)*, (p. 99).

Pustejovsky, J., & Krishnaswamy, N. (2016). VoxML: A visualization modeling language. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA).

Pustejovsky, J., & Krishnaswamy, N. (2019). Situational grounding within multimodal simulations. *arXiv preprint arXiv:1902.01886*.

Pustejovsky, J., Krishnaswamy, N., Draper, B., Narayana, P., & Bangar, R. (2017). Creating common ground through multimodal simulations. *Proceedings of the IWCS workshop on Foundations of Situated and Multimodal Communication*.

Pustejovsky, J., & Moszkowicz, J. (2011). The qualitative spatial dynamics of motion. *The Journal of Spatial Cognition and Computation*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*, 9.

Randell, D., Cui, Z., Cohn, A., Nebel, B., Rich, C., & Swartout, W. (1992). A spatial logic based on regions and connection. *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference* (pp. 165–176). San Mateo: Morgan Kaufmann.

Stalnaker, R. (2002). Common ground. *Linguistics and philosophy*, *25*, 701–721.

Sultana, T., & Badugu, S. (2020). A review on different question answering system approaches. In *Advances in decision sciences, image processing, security and computer vision*, 579–586. Springer.

Tenney, I., Das, D., & Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Tomasello, M., & Carpenter, M. (2007). Shared intentionality. *Developmental science*, *10*, 121–125.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems* (pp. 5998–6008).

Yatskar, M., Zettlemoyer, L., & Farhadi, A. (2016). Situation recognition: Visual semantic role labeling for image understanding. *In Proceedings of the Conference of Computer Vision and Pattern Recognition (CVPR)*.

Zaib, M., Sheng, Q. Z., & Emma Zhang, W. (2020). A short survey of pre-trained language models for conversational ai-a new age in nlp. *Proceedings of the Australasian Computer Science Week Multiconference* (pp. 1–4).

Zwaan, R. A., & Pecher, D. (2012). Revisiting mental simulation in language comprehension: Six replication attempts. *PloS one*, *7*, e51382.

Zwaan, R. A., & Radvansky, G. A. (1998). Situation models in language comprehension and memory. *Psychological bulletin*, *123*, 162.