The Game Engineer's Challenge: Generalizing Language Abstractly And Realizing It Concretely

Catherine Wong
MIT
catwong@mit.edu

Joshua B. Tenenbaum
MIT
jbt@mit.edu

Human language describes what we do in the world, as well as why and how we do it, with remarkable richness and flexibility. Building computational systems that actually leverage language as we do – capable of effortlessly understanding existing words in the context of both new sentences, and new states of the world – poses a fundamental challenge for formally representing linguistic meaning: how is that we can both *generalize the meanings of words so abstractly*, across such a wide range of different environmental and linguistic contexts, and yet also *understand language so concretely*, so that we can make subtle, grounded distinctions contingent on how particular sentences refer to particular worlds? Consider a significant subset of the language used in most text games, which we reuse to describe many arcade games and even highly realistic 2D and 3D simulators: the apparently concrete space of language that describes *physical actions and goals*, including action verbs like *walk*, *run*, *jump*, or simply *move*. Children learn motion verbs like *jump* pretty early on, as verb learning goes [10]. But like other motion verbs, we can reuse a verb like *jump* to refer, or *extend*, freely and abstractly across an enormous set of different and specific motions, spanning a wide range of actors, real and imagined environments, and goals. A competent language user can use *jump* in reference to motions performed by people, four-legged cats, single-legged robots, and likely, to aliens with wildly different bodies that we had never seen jump in the world before; at the same time, we can imagine the quirks of particular *jumps*, the differences between an elephant *jumping,* or a kid on a pogo stick, or a skier. We know how to extend physical action language we used in our own, three-dimensional world to describe motions in environments that do not look or behave physically like this one, or even obey continuity in space and time–after all, we can talk about *jumping* in 2D video games with the barest form of physics, like Frostbite and Qbert, or in games like Super Mario, where it is possible to jump off of mid-air; and in a purely text-based game, we might not think twice about *jumping* from room to room, and then from planet to planet. By the time we reach adulthood, what we know about the meanings of physical actions allows us to extend the same meanings to even highly abstract contexts–once we could describe being *in* a Zoom room, we could talk about jumping out of them. The meaning of *jump* extends to unite them all.

Distributional neural models of word meaning, by contrast, have certainly enabled remarkable advances across many disparate tasks, some of which relate linguistic meaning to the world: we can now predict token sequences across long contexts and even across languages [3]; we can produce linguistic descriptions of certain aspects of perceptual inputs [12]; and we can even follow linguistic instructions, allowing for some compositionality, in specific grounded domains [4, 14, 9, 6]. But where human language is remarkable for its domain generality–what we know about words extends across all of these different environments, and the many concrete action instances and goals within them–nearly all recent approaches to modeling meaning support only extremely domain-limited reasoning: understanding grounded language often means solving particular tasks in the context of a specific model of the world. Even for a single verb like *jump*, on the other hand, the "grounding" challenge that even children can solve permits:

1. Understanding the same word across *different environments* that can vary dramatically in their world state abstraction (consider the differences between *jumping* realized in a text game like Zork, a stylized Atari game like Qbert, a 2D physics-based platformer like Mario, and highly realistic legged robotic simulators); real or imagined contexts with varying

Wordplay: When Language Meets Games Workshop @ NeurIPS 2020,
https://wordplay-workshop.github.io/

underlying dynamics (we can imagine *jumping* over buildings in a single bound); and arbitrary bodies with different motor constraints (we point out spiders *jumping*, as well as knights in a game of chess); and

2. Understanding the same word in the context of a remarkable range of arbitrary and novel linguistic *goals* (consider reasoning about *jumping* to a location above me, over something tall, off of or onto a shaky platform, or to launch a water balloon off of a seesaw);

in order to support a rich range of grounded inferences that include, but are not limited to: (i) *recognizing* instances of the verb across different and entirely new situations, as well as judging how good an example of the action they are; (ii) *imagining and reasoning* about relevant aspects of the verb, conditioned on the linguistic and environmental context; (iii) *executing* instructions containing the verb; and (iv) *planning* with the verb quickly towards arbitrary goals, which require reasoning about different aspects of meaning that themselves vary in their grounding from context to context.

Solving tasks in text games alone may be challenging enough for many distributional and statistical approaches to meaning – much like the challenge of learning representations suitable for recognizing and understanding language used in the context of particular grid worlds, or arcade games [2], or 3D physical simulations [15]. But the rich variance in world state abstraction and dynamics *across* these very different games suggests that a more appropriate grounded language challenge–if we wish to capture the simultaneous cross-domain generalizability and contextual specificity of human language–should look more like emulating the natural program synthesis challenge faced by the game *engineer* attempting to implement a meaningful version of any given verb within a given game: how does the game engineer represent their general knowledge of words (like *jump*) that allows them to abstractly and generally transfer its meaning across a wide variety of existing and entirely new games; and how does that general knowledge relate to the particular implementation they design for a given actor, environment, and constrained set of game goals – a representation that by construction must allow other humans to effectively recognize, plan around, and execute the action in a game world? We propose a roadmap for a grounded language challenge domain, and a research program, motivated by this radical, distinctly language-based, and more human form of generalization:

1. *Developing a dataset that requires understanding words across very different environments, in sentences that make very different contextual demands on word meaning.* Ideally, we suggest a dataset comprised of a diverse set of game environments explicitly designed to span grains of spatiotemporal state abstraction (text games, stylized turn-taking games, 2D platformers, and physically realistic simulators), but that expose a unified interface over the world state for introducing code that implements new 'action' representations, allowing for the same synthesis challenge faced by the game engineer. Further, we suggest developing a challenge set of linguistic *planning queries*, containing verbs that can be feasibly implemented across game environment used in the context of a diverse range of linguistic goals; and ideally, a gold set of *demonstrations*, produced by expert implementations of each action, for verbs in each game. While this ideal challenge would require a significant engineering effort–both to construct the environments, and to engineer ground truth demonstrations–we suggest that many of these features could be captured in domain with a smaller scope but similar breadth: one that requires reinterpreting the same physical action verbs as enacted by agents that move in concretely different ways, depending on their skeletal structures and the underlying physical dynamics of their environment. A labeled dataset of motion verbs instantiated across simulated agents with a diverse set of skeletal morphologies (for instance, the differing robot morphologies, like quadrupeds and humanoid bipeds, in the MuJoCo simulation framework, as well as more 'abstract' morphologies like the cube-shaped figures in Minecraft) and enacted under varying physical dynamics (for example, by modifying the underlying physics engine to reparameterize the gravity) would offer a difficult generalization domain for representations of physical verb meaning that are either entirely abstract–divorced from grounding, or only defined in relation to other words–or entirely concrete, coupled directly to specific motion instances and their environments.

2. *Developing evaluations that require both wide and abstract language generalization, and concrete and grounded realization.* We suggest two tasks based on natural forms of generalization that humans can perform when they know the meanings of words, which could be instantiated in the physical action verbs dataset we have described: *label extension*, which asks whether a particular verb label should be applied to a new motion performed

by an unseen actor (and which can be evaluated as a recognition problem, against human judgments based on the verb label and their observation of grounded motion examples); and *grounded action transfer*, which asks how a given verb label can be performed by an unseen actor (and which can be evaluated by comparing to gold actor-specific motion policies, as well as against human judgements about whether the verb label should be extended to a machine action).

3. *Developing semantic representations with the explicit aim of explaining both the abstract and grounded functions of verb meaning, and how they relate to each other.* We believe that an engineering-oriented approach to interpreting the language of physical actions and goals should draw inspiration from, build on, and integrate advances in representations designed for planning, simulating, and executing actions in real and simulated worlds: in particular, representations for forward execution (e.g. animation, game actions, and robotics control) and planning (e.g. classical planning and hybrd task and motion planning) that are explicitly hierarchical, and that mix abstract semantic representations with physical geometric, kinematic, and dynamic constraints [8, 11, 16, 7]. We do not believe that these representations should be at odds with advances in distributional semantics. Instead, we believe that some of the most exciting developments in both grounded language interpretation [1, 13] and non-linguistic approaches seeking to achieve language-like generalizability [5, 7] have been driven explicitly by the charge of making action representations more *language-like*: capable of capturing the aspects of meaning reused across contexts as different as text games and physically realistic simulations, while efficiently adapting to the specifics of each.

   One promising approach we suggest in this realm is adopting a representational formalism more like the framework of *abstract interfaces* and their *concrete implementations* used in object-oriented programming: interfaces declare the existence of placeholder functions and are specified only based on the relations between them, providing an abstraction barrier between instances of concrete, executable code that implement them. In a framework like this one, the ability to extend physical verb labels so freely to new and different motion instances falls out of the representational flexibility that is explicitly afforded by an interface definition: interfaces are designed to provide representational independence, by only specifying placeholder names and relations between them, without committing the programmer to how those placeholders should be actually concretely realized.
   At the same time, our ability to reason about specific, grounded instances of how verbs should look and behave in the world comes from the way that interfaces are realized into particular concrete implementations: once a programmer maps placeholder names to specific code that interprets them, the result is an executable piece of code, such as a concrete motion objective. This framework also specifies how these two levels should relate to each other in the generalization tasks we described: verb extension becomes the problem of deciding that a concrete instance implements a declared interface; action production involves interpreting the named placeholders in the interface into some concrete realization, such that executing it in fact satisfies the relational specification.

   Indeed, approaches that can be understood through this framework have been described in at least two research areas interested in modeling generalizable motor actions. In animations for games and graphics, the idea of *semantic motion retargeting* [8] asks animators to describe abstract, relational action specifications like *punching* using a vocabulary of semantic body capabilities (like *grasper*) and prepositional relations (like *front* or back) that could be concretized into grounded motion objectives by allowing users to map these semantic names onto concrete character joints; in robotics, the *category-level task manipulation* approach in [7] similarly asks engineers to specify naturalistic goals like *put the mug upright on the table* in terms of semantic names for keypoints, like *center of handle*, that can be grounded concretely to produce grasping objectives across a wide variety of different object topologies. We suggest that a representation for many verbs describing physical action language could draw on this framework in order to meet both computational criteria–wide and abstract verb extension across many action instances, and grounded action transfer sufficient to simulate concrete actions in each world context.

References

[1] J. Andreas, D. Klein, and S. Levine. Modular multitask reinforcement learning with policy sketches. In *ICML*, 2017.

[2] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[4] M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. H. Nguyen, and Y. Bengio. Babyai: First steps towards grounded language learning with a human in the loop. *arXiv preprint arXiv:1810.08272*, 2018.

[5] R. Chitnis, L. P. Kaelbling, and T. Lozano-Pérez. Learning quickly to plan quickly using modular meta-learning. In *ICRA*. IEEE, 2019.

[6] M.-A. Côté, Á. Kádár, et al. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*. Springer, 2018.

[7] W. Gao and R. Tedrake. kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion. *arXiv preprint arXiv:1909.06980*, 2019.

[8] C. Hecker, B. Raabe, et al. Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics*, 2008.

[9] F. Hill, O. Tieleman, T. von Glehn, N. Wong, H. Merzic, and S. Clark. Grounded language learning fast and slow. *arXiv preprint arXiv:2009.01719*, 2020.

[10] J. Huttenlocher, P. Smiley, and R. Charney. Emergence of action categories in the child: Evidence from verb meanings. *Psychological Review*, 90(1):72, 1983.

[11] L. P. Kaelbling and T. Lozano-Pérez. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*.

[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

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

[14] L. Ruis, J. Andreas, M. Baroni, D. Bouchacourt, and B. M. Lake. A benchmark for systematic generalization in grounded language understanding. *arXiv preprint arXiv:2003.05161*, 2020.

[15] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9339–9347, 2019.

[16] M. A. Toussaint, K. R. Allen, et al. Differentiable physics and stable modes for tool-use and manipulation planning. 2018.