SE-Assignment

Victor Aregbede

August 2023

1 Introduction

The symbol grounding problem describes the ability to map words in the language to aspects of the external world. However, the true meaning of language can not be gained purely by computational symbol manipulation. Embodied agents must, therefore, interact with the world to truly understand the meaning of symbols in order to discern impossible goals for possible ones. This is what we call affordability inference as what is discernable is determined by affordances. In cognitive science, the mechanisms by which affordability inference is possible are through a process commonly referred to as simulation semantics – the process by which we understand and reason about utterances by simulating their content. Taking inspiration from cognitive science my research explores using semantic simulation of the effect of actions on objects to facilitate the learning of object affordances, while also providing an ability to generate scenarios for symbol grounding. In particular, I investigate the relational settings between objects from an object-centric approach to infer affordances. I also explore how to transfer this learned affordance to a robot in the real world.

2 Data validation

Data plays a crucial role in my research, that is data both from a simulated environment and data from the real world when dealing with deployment to an embodied agent(a robot). To learn object affordances we need to make use of data that are multimodal in nature. The nature of the data currently is semantic for grounding affordances, visual for learning object properties for affordance and in the future we look to other forms of sensory data like sound and tactile. Considering the relation between different modalities of data to learn and infer affordance, data validation is of paramount importance as it ensures that there's no mismatch between the simulated environment and the real world or a close approximation of the real world. Any deviation could lead to the wrong understanding of affordances. In deep learning(machine learning) data validation is a process to ensure that data used by a model in an ML pipeline for training and evaluation are accurate, consistent, precise and timely(data is updated)[1]. For embodied agents, data validation starts with data from the simulated environment, data for training the model must be accurate covering different scenarios and edge cases-semantics and affordances considered are to be found in the real world. Data has to be consistent, for a case of semantic consistency, when mapping affordances to the semantics it should be logical and consistent- if a semantic tag is 'breakable' objects should have properties that are physically breakable. The Process of getting the data has to be precise, all simulations of effect-action on an object should produce the same result.

3 Boundary testing

Validating various aspects of research can be achieved through boundary value testing, which is a useful framework. For affordance inference, this type of testing can provide rigorous validation. For example, if an embodied agent has a weight limit of 10 kilograms, automated tests can challenge this upper limit. The agent's ability to accurately infer that an 11-kilogram object is un-liftable can be tested, as well as its ability to handle a 9.9-kilogram object. These boundary tests ensure reliable affordance inferences, even under edge conditions. Semantic simulation is another critical area that requires validation. Object relationships and interactions in the simulated environment must adhere to real-world constraints. Automated boundary value tests can examine the system's response to edge cases, such as the maximum distance at which the agent can still interact with an object tagged as "near." These tests establish the robustness of semantic simulation models in capturing real-world relational settings. Symbol grounding mechanisms can also be examined through boundary value testing, such as how well the system grounds symbols when objects are barely visible due to poor lighting or are at the outermost reach of the agent's sensory input range. These edge cases ensure that the system is not only functional but highly adaptive to a range of conditions. Additionally, boundary value testing can be applied to examine algorithmic stability and performance in learning affordances. This testing is crucial for ensuring that learning mechanisms are robust enough for real-world applications. Integrating boundary value testing into the data validation process achieves a higher standard of empirical results. Ultimately, boundary value testing enriches the embodied agents' ability to interact meaningfully with both virtual and physical environments.

4 Architecture and Design

The purpose of architecture is to meet the needs of a system. It serves as a link between the requirements of the system and how it is designed and built. Architectural design enables us to create systems that are strong and can be modified without causing disruption to the entire system[2]. When dealing with complex systems as embodied agents that are dynamic and have real-time requirements, an Event-Driven Architecture (EDA) may be preferred. EDA is an architecture type where each element of the system communicates with one another through broadcasting. EDA's real-time nature is particularly beneficial for affordance inference. For example, when an embodied agent picks up an object, an "object-picked-up" event could trigger immediate affordance inference. This enables more dynamic interactions and quick adaptations to new circumstances in the agent's environment. EDA may also facilitate a more adaptive and real-time approach to symbol grounding. Events triggered by object state changes can quickly update symbol-object mappings. Such adaptability could offer a more fluid and realistic grounding process, further resolving aspects of the symbol grounding problem. Additionally, the modular nature of EDA proves advantageous for semantic simulation. Separate event-triggered modules could handle different aspects like object relationships or potential action effects. This modular separation allows for cleaner code, easier debugging, and straightforward functionality extensions. Data validation can also be seamlessly integrated into this architecture. An event subscriber devoted to boundary value testing could automatically validate data whenever relevant events occur, contributing to system robustness and reliability. Furthermore, EDA inherently provides scalability and extensibility. New event producers and consumers can be added as the research evolves, allowing seamless integration of additional functionalities like multi-agent systems or alternative learning algorithms. The interoperability facilitated by EDA is especially beneficial for transferring learned affordances to real-world robots. A smooth transition from simulation to real becomes feasible as each system component, whether in simulation or reality simply needs to comply with event production and consumption protocols.

4.1 Automated testing

Since each system component has defined interfaces through which it interacts with other components. Automated tests can validate that these interfaces are functioning as expected [3]. For example, when the affordance inference component identifies that an object is 'climbable,' this information should correctly propagate to the semantic simulation component, triggering appropriate simulated actions. Automated interface tests can ensure that data flows seamlessly between these components, preserving the semantic integrity of the symbol-grounding process. Different components in the system often need to share or reference common data—such as object properties or environmental conditions. Inconsistencies in this shared data can lead to incorrect affordance inference or flawed semantic simulations. Automated testing can routinely check for data consistency across different modules, flagging any discrepancies for immediate correction. End-to-end automated tests simulate complete workflows that involve multiple components. For instance, a test might simulate a sequence of events starting from sensor data acquisition, moving to affordance inference, followed by semantic simulation, and finally ending in a real or simulated action by an embodied agent. These tests help ensure that the entire system works in harmony to produce expected and realistic outcomes. Automated tests can also be structured to isolate each component while it interacts with 'mock' versions of the surrounding components. This isolation makes it easier to pinpoint issues when unexpected behaviours occur. For example, if semantic simulation fails to produce realistic outcomes, tests could help discern whether the issue lies within the semantic simulation component itself or is a result of incorrect data being fed to the affordance inference component. Automated scripts can continually check for data consistency, flagging outliers or inconsistencies that could potentially disrupt the semantic simulations. This ensures that the relational settings between objects, which are crucial for affordance inference, are accurately maintained. Given the iterative nature of my research, datasets may evolve. Automated tests can ensure newer datasets maintain integrity and are compatible with existing models. If a dataset is updated to include new object types, automated tests could verify that these new types still enable accurate affordance inferences.

5 Future trends and directions of Software Engineering

It's inevitable that machine learning will be integrated into traditional software engineering practices. Machine learning models have the potential to be predictive tools for software engineers, providing insights into how changes in code or environment can affect system behaviour. My research uses machine learning for affordance inference, which naturally fits into this new paradigm and could contribute predictive models that help with more informed software development processes. As AI systems become more autonomous, it's important to consider ethical issues. I focus on meticulous data validation for internal and external validity, which is directly related to this concern. Before any AI system is deployed, it's necessary to validate its actions and decision-making processes in a simulated environment. This practice is likely to become the industry standard, ensuring that AI systems are both ethical and trustworthy. The trend towards embodied AI agents in realistic simulations is also relevant to my research. As we move forward, the ability of AI agents to understand and interact with their environment will become increasingly important, especially in areas like robotics and IoT devices. My work, which focuses on transferring learned affordances to real-world agents, could be a cornerstone in the development of intelligent systems like these.

References

[1] Eric Breck et al. "Data Validation for Machine Learning." In: *MLSys*. 2019.

[2] Alex Serban and Joost Visser. "Adapting software architectures to machine learning challenges". In: *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE. 2022, pp. 152–163.

[3] Cedric Renggli et al. "Continuous integration of machine learning models with ease. ml/ci: Towards a rigorous yet practical treatment". In: *Proceedings of Machine Learning and Systems* 1 (2019), pp. 322–333.