On the Relationship Between Structure in Natural Language and Models of Sequential Decision Processes

Rafael Rodriguez-Sanchez [*1] **Roma Patel** [*1] **George Konidaris** [1]

Abstract

Human language is distinguished by powerful semantics, rich structure, and incredible flexibility. It enables us to communicate with each other, thereby affecting the decisions we make and actions we take. While Artificial Intelligence (AI) has made great advances both in sequential decision-making using Markov Decision Processes (MDPs) and in Natural Language Processing (NLP), the potential of language to inform sequential decision-making is still unrealized. We explore how the different functional elements of natural language—such as verbs, nouns and adjectives—relate to decision process formalisms of varying complexity and structure. We attempt to determine which elements of language can be usefully grounded to a particular class of decision process and how partial observability changes the usability of language information. Our work show that more complex, structured models can capture linguistic concepts that simple MDPs cannot. We argue that the rich structure of natural language indicates that reinforcement learning should focus on richer, more highly structured models of decision-making.

1. Introduction

Artificial Intelligence (AI) is concerned with designing agents that exhibit intelligent behaviour. These agents are typically formulated as sequential decision-making processes: systems that perceive their environment via sensors, and then must select actions to maximise a utility function. Markov Decision Processes (MDPs) (Puterman, 1994) are widely used to model such tasks, and many extensions have been proposed to model more complex situations by including richer structure.

Language is considered a hallmark of human intelligence—one of the key characteristics that sets us apart from other animals species. The use of language enables the transmission, storage, and evolution of knowledge for humans, and thereby supports sequential decision-making. However, human language is vastly complex. It is marked by semantics, pragmatics, rich syntactic structure, and levels of ambiguity, and the problem of having a computer understand (or generate) it fluently is still unsolved. Therefore, the integration of natural language and decision-making models is of particular interest to researchers aiming to create integrated, general-purpose intelligent agents.

Integrating language information and decision-making agents in the context of Reinforcement Learning (RL) has resulted in marked gains in performance (Luketina et al., 2019). However, there has been relatively little investigation into how the form and structure of the decision process modeled by MDPs—of which there are several classes modeling varying structure and complexity—might be reflected in the natural language appropriate for communicating with an agent solving one. We consider the following question: what parts of speech are appropriate when communicating about what classes of MDPs? As our tasks and models grow increasingly more complex, does the spectrum of language required also broaden? Similarly, does the complexity of natural language suggest that humans represent their own decision-making processes using structure models?

2. Background

2.1. Classes of Markov Decision Processes

In its simplest form, a Markov Decision Process (MDP) (Puterman, 1994) is specified by the tuple (S, A, T, R, γ), where S denotes a set of states, A denotes a set of actions the agent can take, $T : S \times A \rightarrow \Delta(S)$ denotes a transition probability distribution that represents the probability of transitioning to state $s' \in S$ when action $a \in A$ is taken while in state $s \in S$, R denotes a task-specific reward function and γ denotes a discount factor. An agent solving an MDP is typically tasked with finding a policy π, mapping states to actions, that maximises the discounted cumulative

[*] Equal contribution [1] Department of Computer Science, Brown University, Providence, RI. Correspondence to: Rafael Rodriguez-Sanchez <rrs@brown.edu>, Roma Patel <romapatel@brown.edu>.

Accepted to the 1^{st} *Workshop on Language in Reinforcement Learning*, ICML 2020, Copyright 2020 by the author(s).

rewards obtained over time: $\sum_{t=0}^{\infty} \gamma^t r_t$, where r_t is the reward obtained at time t. In hierarchical reinforcement learning (Barto and Mahadevan, 2003), the agent is able to structure its policy to introduce higher-level, temporally abstract, actions, often called *options* (Sutton et al., 1999).

MDPs model the special case where the agent is able to perceive, at every timestep, all the information it requires to decide which action to take. In the more general case, the agent's sensors at each timestep only offer a limited view of the state of its environment. Partially Observable Markov Decision Processes (**POMDP**) (Kaelbling et al., 1998) model this case by extending the MDP tuple to be $(S, A, T, R, \Omega, O, \gamma)$ where the set Ω represents the set of observations that the agent can get and $O : S \times A \to \Delta(\Omega)$ is the observation function such that $O(s, a, z) = p(z|s, a)$, i.e. when the agent in state $s \in S$ executes action $a \in A$, $O(s, a, z)$ is the probability of getting the observation $z \in \Omega$; the agent never has direct access to S. The remaining elements are defined as in the MDP case.

The basic MDP and POMDP formalisms are essentially *unstructured*: while they specify the form of the decision process, they do not impose any further structure or complexity on the form of the states, actions, and observations available to the agent. These formalisms have been extended to describe more structured decision processes. We consider the following types of structure in this paper:

Factored Factored MDPs (Guestrin et al., 2003; Koller and Parr, 2000) and Factored POMDPs (Williams et al., 2005; Katt et al., 2019) structure the state as a vector of state variables $s = \{s_1, s_2, ..., s_n\}$. These variables can typically be partitioned in *factors* $s_{z_j} \subset \{s_1, s_2, ..., s_n\}$. This factorisation can be found naturally when modeling natural systems where state variables have clear semantics. This factored representation is also reflected in the transition function that can be written as the product of such factors z_j, which satisfy the conditional independence property, i.e. $P_{z_j}(s'|s, a) = P_{z_j}(s'|s_{z_j}, a)$.

Object Oriented Object-Oriented MDPs (**OO-MDP**) (Diuk et al., 2008) and POMDPs (**OO-POMDPs**) (Wandzel et al., 2019) further structure the state space by introducing the concepts of objects and object classes. Each object class is defined by a set of attributes (state variables), and each object instance has a state defined by assigning values to these attributes. The state of the entire environment is the union of the state of its constituent objects, thus allowing a more efficient and understandable representation.

Parameterised Actions Parameterised Action MDPs (**PAMDP**) (Masson et al., 2016) extend the set of actions A to be parameterised by a vector $x \in \mathbb{R}^{m_a}$. An action selection by the agent is then a pair (a, x) specifying the discrete actions as well as its parametrisation (for example, to kick a ball with a certain amount of force, or to move at a certain velocity). The extension of the model to the POMDP case is straightforward, and text-based games are an existing use of PA-POMDPs (Narasimhan et al., 2015).

Decentralised Decentralised MDPs (**Dec-MDP**) (Bernstein et al., 2002) and POMDPs (**Dec-POMDP**) (Nair et al., 2003) extend the MDP and POMDP formalisms to model the multi-agent case, in environments consisting of multiple agents—that collectively maximize the same reward function—selecting actions in a decentralised manner.

Each of these formalisms are obtained by assuming more structure about the basic MDP or POMDP formalism. That structure adds complexity and narrows the set of tasks to which the model is applicable, but at the same time gives the agent the opportunity to exploit the additional structure during learning or planning.

2.2. Syntactic Categories in Language

Syntactic categories, also known as parts-of-speech (POS), are classes of words that have semantic tendencies—for example, nouns describe *objects* while adjectives refer to *properties*. They are broadly categorised into closed class (e.g., determiners such as *"a, the"* or prepositions such as *"on, at"*, that are rarely coined or expanded as times change) and open class (e.g., verbs such as *"zoom, fax"* or nouns like *"Macbook, Roomba"* that are continually created as needed). There are four main open classes i.e., **nouns, verbs, adjectives, adverbs** each of which are subcategorised. We refer readers to the work of Marcus et al. (1993) for a full overview of parts-of-speech and the 45 categories annotated by the Penn Tree Bank. POS categories are important for several language understanding tasks, since they reveal important information about properties of the word, as well as its context. For example, for words that have different POS tags in different concepts (e.g., *"'dash"* as a verb versus *"dash"* as a noun), knowing their POS tags could help resolve ambiguity to understand the meaning of the sentence.

2.3. Different Syntax for Different Languages

We should note that, although the four main POS categories seem like fundamental syntactic constructs, some languages (like Riau Indonesian or Tongan) do not even make a distinction between nouns and verbs (Broschart, 1997). There also exist languages devoid of a certain class e.g., adjectives in Korean, where words that would normally be adjectives in English translate to a subclass of verbs in Korean (*"beau-*

Tag	Description	Example
CC	coordinating conjunction	*and, or, if*
V	verb	*move, push, pull*
IN	preposition	*above, below, on*
NN	common noun	*wall, location*
NNP	proper noun	*Taxi, Agent*
ADJ	adjective	*blue, round, small*
EX	existential 'there'	*there*
MD	modal	*can, should*
ADV	adverb	*quickly, slowly*
PRP	personal pronoun	*your, their*

Table 1. Parts of speech (as tagged by the PTB) for word classes important to decision making.

tiful"→ *"to be beautiful"*). Thus, although the different languages vary in the syntactic categories they cover, their functionality can be replicated, albeit at a cost (e.g., larger number of words).

3. Language Elements for Describing Decision Processes

We now draw connections between *which* part-of-speech elements can be related to each model class. We summarise these facts in Table 2.

Unstructured MDPs This is the base formalism for decision processes in AI and the one that includes the least structure. The state representation is completely unstructured and, hence, tying nouns to elements of the model can be difficult. However, **proper nouns** relate to states with certain characteristics such as goal states. Actions are referred to using **intransitive verbs**—as there is no concept of an object to apply the action on. Conditional statements and connecting words i.e., **conjunctions** like *"and, or"* are used to tie together specifications of parts of the MDP. **Prepositions** denoting order relations, such as *"before, after"* are used to describe ordering a sequence of actions or sub-policies. **Determiners** such as *"more, less, equal"*, **cardinal quantities** and **comparative adjectives/adverbs** are used to specify rewards.

Factored MDPs In addition to the elements above, **proper nouns** can be used to name factors. These nouns are unique in the sense that no two factors are the same. As nouns, we can qualify them by using **adjectives** that can specify properties of the factor such a particular setting of the factor. For instance, if we consider the coordinates of a robot (x, y) to be "location", "*home* location" could be the coordinates $(0, 0)$.

OO-MDPs Object classes in OO-MDPs correspond to the concept of **common nouns**. In this way, we can use **determiners** such as *"a, the"* to talk about a specific instance of an object or about any object of a class. As before, **adjectives** qualify object instances. In this way, we can map qualified nouns to instances with particular attributes—e.g. *the red ball*. With different numbers of objects, we can use **quantifiers** such as *"there is/are, all"*. Specific instances of an object can be specified by **proper nouns**. Given the existence of objects, **transitive verbs** can be mapped to actions that affect objects, in this way information about the dynamics of the world can be represented with more complex constructs as the transition function relates to the objects' state. The fact that humans have rich notions of "nouns" and concepts hints that they are likely using this sort of structure.

PAMDPs In this class of MDPs, we have actions that are parameterized, thus calling for **adverbs** as a way to qualify action (**verbs**). In this way, we can now realise actions that correspond to verb phrases such as "*to go up slowly*".

Dec-MDPs In these models, we have referential expressions, in order to denote concepts of oneself and other existing entities in the world for multi-agent environments. Therefore **pronouns** (e.g., personal, possessive) come into play.

Until now, we have described the parts-of-speech that are relevant to increasingly more structured MDPs. Analogously, we proceed with the partially observable formalisms.

Unstructured POMDPs In these models, the important difference is the presence of partial observability. Part-of-speech that relate to MDPs are still pertinent here. However, we can use POS elements that can convey facts and uncertainty about the world such as **modal verbs** like *"could, should"* that allow to specify information about the world. Facts are necessary in partially observable domains in order to reduce the uncertainty the agent has about the state of the world. This is not the case when the state is completely observable as the agent knows all relevant attributes of its environment at each timestep.

Factored POMDPs Along with the above POMDP elements, **proper nouns** are used in the same way as in Factored MDPs. Similarly, **adjectives** can specify properties of state variables, that differentiate them from other states.

OO-POMDPs In OO-POMDPs, we need to express facts about the objects. Therefore, **prepositions** that denote order such as *"in, at, on, below, above"* are necessary to state relative order among object instances' states. For instance, if we instantiate an object class *"cup"*, we can use specify facts about an object class: *the cup is* in *the kitchen*.

MDP	V	CC	IN	Quant	NN	NNP	ADJ	EX	MD	ADV	PRP
MDP	√	√	√								
Factored MDP	√	√	√	√	√						
OO-MDP	√	√	√	√	√	√	√				
Unstructured POMDP	√	√	√					√	√		
Factored POMDP	√	√	√	√	√			√	√		
OO-POMDP	√	√	√	√	√	√	√	√	√		
LTL-MDP	√	√	√							√	
Dec-MDP	√	√	√								√
Dec-POMDP	√	√	√					√	√		√
Dec-OO-POMDP	√	√	√	√	√	√	√	√	√		√

Table 2. Differences in complexities in different classes of MDPs. Each row shows a class of MDPs while columns show POS categories that can be used to ground to existing parts of each model, as enumerated in Section 3.

Dec-POMDPs Analogously to the observable case, we have that in this case we need to specify facts about other agents' state and policies. As in OO-POMDPs, **prepositions** are necessary to provide both spatial information about the agents and temporal information—e.g. *"before, after"*—when ordering and coordination among agents' policies is required.

Non-Markovian Policies Through the combination of conditional statements and **temporal prepositions** such as *"until, before, after"*, it is possible to specify instructions for a task which relates to further structure in the action space of the model. These instructions are related to (sub-)policies in any of the models we have discussed thus far. However, these may be non-Markovian given that the use of prepositions such as "until" implicitly requires the agent to record past actions and states in order to determine if a condition is satisfied. Linear Temporal Logic (LTL) has been used in several works (Ding et al., 2011; Oh et al., 2019) that use **LTL-MDPs** to handle such dependencies.

4. Discussion and Conclusions

In this work, we were primarily interested in characterising what elements of natural languages are important to different MDPs. The relation between parts-of-speech and decision making models we have laid out here are by definition true. Which means that it is the case that certain parts-of-speech may become dispensable given the type of decision process the agent must solve. Therefore, our claims are not about trying to establish an empirical performance benchmark, but instead about establishing what elements of language are necessary for the different types of decision process.

We have attempted to use insights from syntactic constructs in natural languages to characterise the differences in the forms of language useful for communicating with agents using different classes of decision-making models. We posit that the richness of language has its roots in the richness of the decision process that humans are solving. From the parallels drawn in the previous section, we can see how much of language can be (partially) related to elements of the different MDP classes, which in turn suggests the different kinds of structure and abstractions that humans might be implicitly using to tractably solve their own decision problems. This suggests that the form of language used to describe a task has the potential to aid in automatically determining the structure and abstraction necessary for an agent to solve that task.

Moreover, we wish to emphasize that our exploration shows that much of humans' use of language is for conveying factual information about objects and object classes, which strongly suggests that human decision processes are both partially observable and highly structured. However, the flexible nature of language, both in the way that new elements—such as those of the open classes of syntactic elements—can be defined based on known elements and how verbs and nouns can be qualified in new ways, could result from the flexibility with which we can generate new abstractions to handle new problems and better handle partial observability.

Current RL research does not use highly structured and partially observable models—many current efforts go into designing algorithms with as little structural bias as possible. The richness of human language—and its clear links to richer, more structured representations—suggests that RL research should perhaps focus instead on highly structured formalisms, especially for research on grounding language in RL.

References

A.G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13:41–77, 2003.

D.S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of operations research*, 27

(4):819–840, 2002.

J. Broschart. Why Tongan does it differently: Categorial distinctions in a language without nouns and verbs. *Linguistic Typology*, 1(2):123–165, 1997.

X.C. Ding, S.L. Smith, C. Belta, and D. Rus. MDP optimal control under temporal logic constraints. In *Proceedings of the 2011 IEEE Conference on Decision and Control and European Control Conference*, pages 532–538, 2011.

C. Diuk, A. Cohen, and M.L. Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 240–247, 2008.

C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

S. Katt, F.A. Oliehoek, and C. Amato. Bayesian reinforcement learning in factored POMDPs. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 7–15, 2019.

D. Koller and R. Parr. Policy iteration for factored MDPs. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000.

J. Luketina, N. Nardelli, G. Farquhar, J. Foerster, J. Andreas, E. Grefenstette, S. Whiteson, and T. Rocktäschel. A survey of reinforcement learning informed by natural language. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6309–6317, 2019.

M. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

W. Masson, P. Ranchod, and G. Konidaris. Reinforcement learning with parameterized actions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1934–1940, 2016.

R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 705–711, 2003.

K. Narasimhan, T. Kulkarni, and R. Barzilay. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*, 2015.

Y. Oh, R. Patel, T. Nguyen, B. Huang, E. Pavlick, and S. Tellex. Planning with state abstractions for non-Markovian task specifications. In *Robotics: Science and Systems*, pages 59–68, 2019.

M.L. Puterman. *Markov Decision Processes*. John Wiley & Sons, 1994.

R.S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2): 181–211, 1999.

A. Wandzel, Y. Oh, M. Fishman, N. Kumar, L.L.S. Wong, and S. Tellex. Multi-Object Search using Object-Oriented POMDPs. In *Proceedings of the 2019 IEEE International Conference on Robotics and Automation*, 2019.

J.D. Williams, P. Poupart, and S. Young. Factored partially observable Markov decision processes for dialogue management. In *Proceedings of the IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 76–82, 2005.