Towards Robot Knowledge Consultants Augmented with Distributed Short Term Memory

Tom Williams*, Evan Krause†, Bradley Oosterveld†,
Ravenna Thielstrom†, and Matthias Scheutz†
*MIRRORLab
Colorado School of Mines
Golden, CO, USA
twilliams@mines.edu
†Human-Robot Interaction Lab
Tufts University
Medford, MA, USA
{firstname.lastname}@tufts.edu

I. Introduction

For robots to engage in natural task-based dialogues with human teammates, they must both understand and generate natural language expressions to refer to entities in their shared environment, such as people, locations, and objects [7, 14, 19]. These tasks, *reference resolution* and *referring expression generation*, are particularly challenging in realistic robotics applications due to the realities of how knowledge is represented and distributed in modern robotic architectures.

In previous work we presented a *Consultant Framework* [15] that allows a robot's distributed sources of knowledge to be used during both reference resolution [17] and referring expression generation [18], without requiring the language processing system to have any knowledge of how knowledge in different domains is represented and accessed.

This domain independence, however, comes at increased computational cost, especially during natural language generation. Specifically, we have used this framework to modernize the classic Incremental Algorithm [4] for referring expression generation, relaxing assumptions of that original algorithm: that knowledge is certain, that knowledge is centrally stored, and that a list of all properties known to hold for each known entity is centrally available during referring expression generation. Our Consultant Framework allows these assumptions to be relaxed, producing a referring expression generation algorithm tailored to the realities of robotic architectures, but which is computationally inefficient.

The Incremental Algorithm requires iterative consideration of potential properties that could be added to the description, considering for each whether that property holds for the to-be-described target and does not hold for at least one distractor. Under the assumptions of the classic Incremental Algorithm, these considerations can be performed as simple set-membership checks on the centrally available property sets. When the assumption as to the existence of these property sets is relaxed, however, as is the case in the modified algorithm designed to leverage our Consultant Framework, these considerations must instead be made through queries to the Consultants responsible for the target and distractors. The computational complexity of referring expression generation combined with the computational cost of these queries results in a significant computational burden.

To address this computational burden, we propose an augmented Consultant Framework that includes Consultant-Specific Short-Term Memory Buffers that cache a small number of properties determined to hold for various entities.

II. Augmented Framework

In previous work [15, 18] (see also [16, 17]), we presented a framework of "Consultants" for the DIARC architecture [12] that allows information about entities to be assessed when knowledge is uncertain, heterogeneous, and distributed, in a way that facilitates the use of Incremental Algorithm-inspired approaches to referring expression generation. Specifically, each consultant c facilitates access to one Knowledge Base (KB) k, and must be capable of at least four functions:

1) providing a set c_{domain} of atomic entities from k,
2) advertising a list $c_{constraints}$ of constraints that can be assessed with respect to entities from c_{domain}, *and that is ordered by descending preference*.
3) assessing constraints from $c_{constraints}$ with respect to entities from c_{domain}, and
4) adding, removing, or imposing constraints from $c_{constraints}$ on entities from c_{domain}.

In this section, we define a *Short-Term Memory(STM)-Augmented* Framework with an additional requirement:

5) providing a list c_{STM} of properties that hold for some entity from c_{domain}.

Crucially, the properties returned through this capability do not need to be all of the properties that hold for the target entity. A consultant may have a large number of properties that it could assess for a given entity if need be, some of which might be very expensive to compute. As such, the purpose of this capability is not to request evaluation of all possible properties for the specified entity, but rather to request the contents of a small cache of properties recently determined to hold for the specified entity, if any.

Drawing on insights from cognitive psychological theories of working memory [10], the new capability required in the *STM-Augmented Consultant Framework* requires each consultant to maintain its own set of *features* currently remembered for the set of entities for which it is responsible. This serves to allow fast access to a set of entity properties likely to be relevant, in order to avoid the expensive long-term memory queries

that make processes such as referring expression generation so expensive in the current consultant framework. In the next section we describe how our newly proposed framework can be used during the course of referring expression generation.

III. Algorithmic Approach

We will now motivate *SD-PIA*, an STM-Augmented and Distributed variant of the Probabilistic Incremental Algorithm (c.f. [18]). The key difference between this algorithm and our previous approach, DIST-PIA, is our leveraging of the properties stored in STM before performing LTM-Query intensive operations. We refer to the interested reader to our original paper [18] for a full walkthrough of our original algorithm. In this section, we will instead simply describe the differences between *DIST-PIA* and *SD-PIA*.

While DIST-PIA crafted sub-descriptions using a single algorithm, SD-PIA begins by crafting an initially (possibly partial) sub-description using only the properties found in Short Term Memory Buffers. If the sub-descriptions returned through this algorithm are not fully descriminating, the partial sub-description is augmented by passing the set of still-to-be-eliminated distractors to a second algorithm, which operates much the same as our original algorithm.

The other major difference between *DIST-PIA* and *SD-PIA* comes in the design of *SD-PIA*'s helper function. Instead of considering all properties advertised by the consultant responsible for the target, *SD-PIA* considers only the properties returned by querying that consultant's STM buffer, requiring a single query rather than $O(c_m^{\Lambda})$ queries. For each of these already-known-to-hold and already-bound queries, *SD-PIA* iteratively rebinds the query to refer to each distractor x rather than the target entity. For each re-bound query, *SD-PIA* calls a function *stm-apply*, which checks whether that property holds for that distractor (x), by first checking whether the property exists in the Short Term Memory Buffer maintained by Consultant c_x for x, or, if and only if this is not the case, by checking whether the property is known to hold by Consultant c_x using its' $apply$ method, as usual.

IV. Design Trade-offs

The use of Short Term Memory Buffers in this augmented algorithm involves a number of design trade-offs. The primary motivation behind this approach is increasing performance: the number of queries needed when choosing properties to use may be much lower when those properties alone are sufficiently discriminating. Similarly, when determining whether chosen properties serve to rule out distractors, there is potential for significant performance gain, as such decisions may be able to be made on the basis of set-membership checks, rather than requiring costly queries of long term memory.

Moreover, we believe the use of these buffers may facilitate *lexical entrainment* [2]: the process by which conversational partners converge on common choices of labels and properties over the course of a conversation. If a robot's Short Term Memory buffers are populated with those properties used by itself and its interlocutors, and if the properties contained in those buffers are considered before others, than the use of these buffers may directly lead to such entrainment.

The use of these buffers may, however, come with negative consequences as well. Because the robot is arbitrarily restricting itself to a subset of the properties it *could* otherwise choose to use, it may force the robot into local maxima in the landscape of possible referring expressions. Moreover, the robot runs the risk of using a property that does not actually hold if it does not appropriately handle contextual dynamics. For example, an object previously described as "on the left" may no longer be "on the left" if the object, robot, or interlocutor has moved since the object was last discussed.

There is a vast body of psychological literature that could be exploited to prevent such mistakes from being made: A context-sensitive decay-based model of working memory might prevent this by having different properties "decay" out of cache after a certain amount of time or with a certain probability, with time or probability proportional to the degree to which the property is dynamic, i.e., how likely it is to change over time [1, 13]. A resource-based model might prevent this by having a limited total buffer size, and have property dynamics factor into the decision of what to bump from memory when new things need to be inserted into an already-full buffer [3, 5, 6]. Finally, an interference-based model might prevent this by having properties added to a buffer "overwrite" the most similar property currently in the buffer [8, 9, 11]. Note that these are loose characterizations of their respective theories from cognitive psychology; a comprehensive discussion of relevant psychological theories perspective can be found in [10]. Of course, the approach taken need not be cognitively plausible. The robot could, for example, use a model of the dynamics of different properties to periodically re-sample the properties held in its buffers.

The question of cognitive plausibility also raises a different question: how extensive should the robot's memory caches be? Should the robot keep property caches for all entities, for only those that are relevant in the current context, or for an even smaller set? And for each entity, should the robot track all relevant knowledge for so long as that entity is tracked, or should it track only a fixed, small number of properties? And should such limits be local, or global limits shared between tracked entities? These are once again questions for which candidate answers can be gleaned from the psychological literature [10]. Here, interesting tradeoffs can be made: while robots can be made to remember much more than humans, expanded memory may come at a computational cost. Moreover, choosing to remember more means increased risk of incorrect behavior due to mishandling of property dynamics.

Ultimately, experimentation will be needed to tease out different tradeoffs made by the proposed approach. We plan to explore the efficacy of this approach and its tradeoffs using a modified version of the evaluation framework proposed in our previous work [18]. In more distant future work we plan to evaluate the tradeoffs of aligning the robot's approach with that of different cognitive theories.

References

[1] Alan D Baddeley, Neil Thomson, and Mary Buchanan. Word length and the structure of short-term memory. *Journal of verbal learning and verbal behavior*, 14(6): 575–589, 1975.

[2] S E Brennan and H H Clark. Conceptual pacts and lexical choice in conversation. *Journal of experimental psychology. Learning, memory, and cognition*, 22(6): 1482–1493, 1996. ISSN 0278-7393.

[3] Robbie Case, D Midian Kurland, and Jill Goldberg. Operational efficiency and the growth of short-term memory span. *Journal of experimental child psychology*, 33(3): 386–404, 1982.

[4] Robert Dale and Ehud Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, 1995.

[5] Marcel A Just and Patricia A Carpenter. A capacity theory of comprehension: individual differences in working memory. *Psychological review*, 99(1):122, 1992.

[6] Wei Ji Ma, Masud Husain, and Paul M Bays. Changing concepts of working memory. *Nature neuroscience*, 17 (3):347, 2014.

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

[8] James S Nairne. A feature model of immediate memory. *Memory & Cognition*, 18(3):251–269, 1990.

[9] Klaus Oberauer and Reinhold Kliegl. A formal model of capacity limits in working memory. *Journal of Memory and Language*, 55(4):601–626, 2006.

[10] Klaus Oberauer, Simon Farrell, Christopher Jarrold, and Stephan Lewandowsky. What limits working memory capacity? *Psychological bulletin*, 142(7):758, 2016.

[11] Satoru Saito and Akira Miyake. On the nature of forgetting and the processing–storage relationship in reading span performance. *Journal of memory and Language*, 50(4):425–443, 2004.

[12] Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. An overview of the distributed integrated cognition affect and reflection diarc architecture. In Maria Isabel Aldinhas Ferreira, Joo S.Sequeira, and Rodrigo Ventura, editors, *Cognitive Architectures*. 2018 (in press).

[13] Richard Schweickert and Brian Boruff. Short-term memory capacity: Magic number or magic spell? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 12(3):419, 1986.

[14] Kees Van Deemter. *Computational Models of Referring: A Study in Cognitive Science*. MIT Press, Cambridge, Massachusetts, 2016.

[15] Tom Williams. A consultant framework for natural language processing in integrated robot architectures. *IEEE Intelligent Informatics Bulletin*, 2017.

[16] Tom Williams and Matthias Scheutz. POWER: A domain-independent algorithm for probabilistic, open-world entity resolution. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1230–1235, 2015.

[17] Tom Williams and Matthias Scheutz. A framework for resolving open-world referential expressions in distributed heterogeneous knowledge bases. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, pages 3598–3964, 2016.

[18] Tom Williams and Matthias Scheutz. Referring expression generation under uncertainty: Algorithm and evaluation framework. In *Proceedings of the 10th International Conference on Natural Language Generation (INLG)*, 2017.

[19] Tom Williams and Matthias Scheutz. Reference resolution in robotics: A givenness hierarchy theoretic approach. In Jeanette Gundel and Barbara Abbott, editors, *The Oxford Handbook of Reference*. Oxford University Press, 2017.