Forgetful Multi-store Memory System for a Cognitive Assistive Robot

Angel Fernando García Contreras[1], Seiya Kawano[1], Yasutomo Kawanishi[1],
Yutaka Nakamura[1], Satoru Saito[2], Koichiro Yoshino[1]
[1]Guardian Robot Project (GRP), RIKEN, Kyoto, Japan
[2]Graduate School of Education, Kyoto University, Kyoto, Japan
{angel.garciacontreras, seiya.kawano, yasutomo.kawanishi,
yutaka.nakamura, koichiro.yoshino}@riken.jp
saito.satoru.2z@kyoto-u.ac.jp

Abstract

Domestic robots are intended to coexist with humans, providing assistance and companionship. A key element to realize these interactions is life-long learning from their environment. Even with promising progress, the state of the art remains far from realizing life-long robotic cognition, complicated by the complexity of multi-modal, ambiguous real-world data and the limits of storage capacity. The Guardian Robot Project at RIKEN is developing Indy, an autonomous helper robot. In this work we present the first step towards Indy's long-term memory system, inspired by two concepts from cognitive psychology: a three-store memory model, and forgetting heuristics that help retain useful information and discard irrelevant sensor data. We outline the general framework of this system, and evaluate the impact these techniques have on the total size of stored memory using metrics to estimate the space growth of memory and its accuracy.

1 Introduction

The design and development of interactive cognitive robots that can coexist with humans in their daily lives is challenging. One of the biggest challenges is long-term functioning, as the robot would need to gather and store large amounts of continuous multi-modal data over long periods of time (preferably years) in order to make decisions that are relevant to its function. Thus, for long-term functioning, we need to consider the design of memory management; or, how data is represented, prioritized, accessed and maintained.

At RIKEN Guardian Robot Project, we are working on an autonomous robot with aims of being a long-term companion to humans. This includes the design and development of a memory system capable of functioning effectively on the long-term, by using multi-store and forgetfulness concepts inspired by cognitive psychology. In this paper we introduce this memory system, and show the results of an initial evaluation on the effect of the implemented heuristics on the total size of the memory in some interactive scenarios.

2 Autonomous Robot Indy

Indy, shown in Figure 1, is being built as an autonomous robot capable of coexisting with people in spaces built for humans. It has sensors that give it the following capabilities:

- *Object Recognition* using a YOLOv8 [1] model enhanced with SORT-based [2] tracking.
- *Person Recognition* derived from its object recognition module, with additional pose keypoint tracking via OpenPifPaf [3].
- *Scene Graph Generation* through a model trained on the Visual Genome [4] dataset.
- *Speech Recognition* detects speakers using LIDAR-enhanced location detection and noise reduction [5] and transcribes the isolated sound using a fine-tuned model based on Whisper-large-v2 [6].
- Chit-chat Responses using the Japanese Language Transformer model [7] and text-to-speech synthesis with a custom voice by ReadSpeaker[1).

1) https://readspeaker.jp/

- *Data storage*: Indy uses an on-hardware MongoDB[2] database to store persistent memory.

Indy's design is modular, with many of these modules in active development. These recognition systems will be improved and enjoined with other systems. Thus, the memory system is designed with modularity and interchangeability in mind.

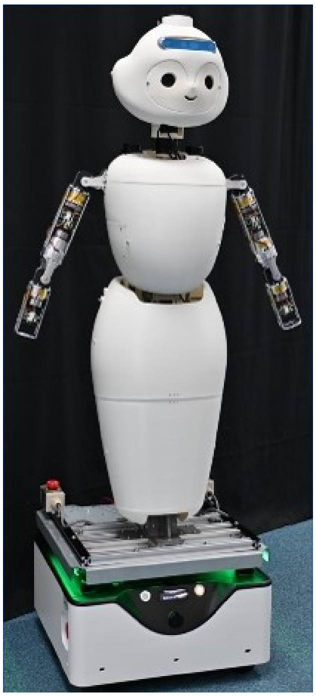

Figure 1 Autonomous Robot Indy

3 Memory System

We started designing Indy's memory system with the aim of making efficient use of memory space, minimizing the growth over long periods of time. To that end, we took inspiration from the most efficient memory usage we know: the human brain. In particular, we looked at concepts of cognitive psychology to develop systems aimed at reducing memory usage growth: multi-store memory, and forgetting curves.

Other memory frameworks: Cognitive psychology has also inspired the development of *cognitive frameworks*, such as ACT-R [8] and SOAR [9], that simulate the way information is processed in the brain, which includes processing memory. These frameworks have been used with various degrees of success in various robotics applications, such as route mapping and control [10, 11], and human interaction [12, 13]. While the work we present here is also inspired by the same principles, the implementation we are aiming for does not seek the replicate / simulate human thought processes, but to provide a foundation to build a cognitive model that fits the needs of Indy's own development and systems.

3.1 Three-store Memory System

Atkinson-Shiffrin's Multi-Store model [14] identifies three memory "levels", based on how much of a stimuli is used and retained, and for how long. Inspired on this model, we designed a three-store memory system for Indy:

- *Sensor Memory* can be thought as a buffer, temporarily storing the output of Indy's recognition systems. This store works at the same rate as recognition, and is cleared regularly as its data is used.
- *Short-Term Memory*, also known as "working memory", digests the data from sensor memory into "chunks" representing the same entity or event. This way, a single object "detected" multiple times is recognized as the same object, and stored only once. Repeated perceptions of the same object trigger a "recall" of the same object, which increases its chunk's impression and longevity. If a chunk is not "recalled" frequently enough, it is forgotten.
- *Long-Term Memory* converts chunks that have maintained enough relevance into a different representation, building relational and temporal relationships and an internal narrative. This system is inspired by episodic memories of humans.

For this paper, we developed and evaluated the sensor and short-term memory modules.

3.2 Forgetfulness Heuristics

Forgetfulness is a mechanism that allows the human brain to maintain life-long learning within a physically limited storage. Since they were first studied by German psychologist Hermannn Ebbinghaus [15], the general behavior of forgetfulness has been codified as probability curves, expressing the decay of the likelihood of remembering a given piece of data. For Indy, we designed two types of "forgetfulness heuristics": *forgetting curves*, and *compilation rules*.

2) https://www.mongodb.com/

3.2.1 Forgetting curves

For Indy's forgetting curves, we wanted an equation whose behavior could be controlled and adapted parametrically, in order to represent different memory behaviors (i.e. different modalities). We chose a modified Weibull probability curve used in a large-scale memory and forgetting study [16]. The Weibull forgetting function in Equation (1) represents the probability that a given memory chunk is forgotten given an initial memory strength μ at a time t, with a decay parameter a, and a early-late forgetting balance parameter d.

$$p(t) = \mu\ exp\left((at/d)^d\right) \tag{1}$$

In Indy's system, we chose to make this a deterministic system: instead of treating it as a probability, a memory "chunk" whose curve value drops below a given threshold is considered safe to forget.

3.2.2 Compilation Rules

Sensor memory acts as a buffer for the recognition systems. Each recognition system produces data at regular intervals; for example, object recognition outputs results once every $1/30^{th}$ of a second. To store redundant recognition results more efficiently, we need *compilation rules* that outline how the sensor memory data should be managed: if it is too redundant, it is safe to discard; if not, then represent the perceived data into a more compact representation that also manifests how the subject of perception may have changed.

For this work, we implemented and evaluated a subset of the recognition systems, with the following compilation rules:

- *Object recognition* returns centroids and viewport coordinates; our memory uses these to reconstruct an approximation of a box that encloses the object. Every new perception of the same object can potentially update this enclosure, if a new perception of an identical object is within/near the abstract box. Additionally, we determine the object's importance based on its proximity to Indy and its prominence within the viewport.
- *Person recognition* follows similar rules to object recognition, but also computing the similarity (measured as cosine distance) in the perceived person's pose to an existing record in short-term memory.
- *Speech Recognition and Generation* sensors produce separate output, and are kept as such in sensor memory. Short-term memory assigns them to a single storage, in chronological order.

4 Evaluation and Experiments

For this initial implementation, we want to look into a question: as Indy accumulates new data, how is the old data in memory changed? In particular, we want to focus on how *memory size* changes in relations to *queries* made to the memory system as *time* passes, when using a *multi-store* memory with *forgetfulness*. We prepared a set of 100 queries in natural language, representing topics that a user could reasonably ask Indy to recall. These queries involve objects and people that can be found with relative frequency in Indy's environment at the Guardian Robot Project, as well as subjects that are not available to and could not possibly be perceived by Indy. We stored all data published by Indy's sensors over a 24-hour period, during which members of Guardian Robot Project walked naturally in the office space surrounding the robot, occasionally talking with Indy and placing objects within the robot's field of vision to reinforce them in memory. The data was then moved to a stand-alone computer, where it was used to simulate two types of memory systems: a "sensor memory-only" system, and a "two-store memory with forgetting" system. During this simulation, we chose three moments in which to make the 100 queries, at $t = [300, 3600, 86400]$ seconds. For each query, we measured the following data:

- *Number of Entries* in the database that were consulted to answer the query;
- *Query Execution Time* in milliseconds, as reported by MongoDB's engine;
- *Number of Memory Records Returned* as a response to the query;
- *Accuracy*, or whether the query resulted in a record found in the database, and the record matched the original data published by the sensors.

5 Results and Analysis

The average execution results for simulating a full sensor memory system (Table 1) and a two-store memory system (Table 2) show that the second system provides a reduction on all measurements of multiple orders of magnitude, suggesting the reduction in size could also have a deeper impact in overall system performance by returning results much faster, even after a long amount of time has passed.

Accuracy, Recall [17] and F1 Score [18] are often used to measure accuracy in natural language processing systems. We are omitting the results for the full sensor memory system because they would mirror the original database, so all measures have a value of 1. For the results in the two-store memory system in Table 3, and as far as memory alone is concerned, the system does not lose any significant information, even with a significantly smaller number of database entries. After a full 24 hours, we see the effect of "forgetfulness", with a reduction of accuracy across all measures, as the queries request objects and topics that have not been frequently "rehearsed". We suspect this number is particularly high because of the limited area in which Indy operated during data collection, which caused a near-constant perception of the same objects in the environment.

Finally, we computed the ratio of the same execution measurement averages between the full sensor memory and the two-store memory in Table 4. Of note is the rate at which the ratio metrics change over time: the decrease suggests that not only are the sizes in the two-store memory smaller, but their growth is orders of magnitude slower as well.

t (secs)	N.Entries	Exec.Time	Doc.Ret.
300	2733.72	2.18	33.45
3600	23629.65	41.05	312.69
86400	411725.46	1070.68	9762.77

Table 1 Avg. exec. data for full sensor memory queries

t (secs)	N.Entries	Exec.Time	Doc.Ret.
300	24.31	0.74	0.43
3600	144.40	0.24	2.30
86400	2185.88	2.09	34.23

Table 2 Avg. exec. data for two-store forgetful memory queries

t (secs)	Accuracy	Recall	F1 Score
300	1.00	1.00	1.00
3600	1.00	1.00	1.00
86400	0.78	0.70	0.83

Table 3 Accuracy measures of short-Term memory system

t (secs)	N.Entries	Exec.Time	Doc.Ret.
300	0.00889	0.33945	0.01286
3600	0.00611	0.00585	0.00736
86400	0.00531	0.00195	0.00351

Table 4 Ratio of avg. between full and two-store memory systems

6 Conclusions and Future Work

In this work, we presented the memory system framework for an autonomous robot. This memory system stores information from a robot's recognition streams and, inspired by research in cognitive psychology, select data to "compile", keep or "forget" in order to reduce its total size growth over long-term periods of time. We presented the general process that drives an initial partial set of modules for this memory system, and carried out a groundwork evaluation of these modules to determine their impact on memory size and retrieval. This evaluation shows that for the implemented modules, even a partially-implemented memory system is capable of reducing the memory size by orders of magnitude without drastic loss of data.

In our future work, we will further develop this framework to be a full three-store memory system including long-term memory, and investigate the effectiveness of various representations of memory, such as a narrative first-person memory. In our initial implementation, we utilized re-observation as the trigger of recalling; however, using different modalities for such triggers is essential. For example, information that someone was talking about an actual object will be a trigger for recalling the observation of object recognition. We will implement such cross-modal recalling to improve our forgetting mechanism.

Acknowledgements

A part of this work is supported by JSPS Kaken Grant Number 23K19984 and 22H04873.

References

[1] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Yolo by ultralytics, October 2023.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In **2016 IEEE International Conference on Image Processing (ICIP)**, pp. 3464–3468. IEEE, sep 2016.

[3] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Openpifpaf: Composite fields for semantic keypoint detection

and spatio-temporal association. In **IEEE Transactions on Intelligent Transportation Systems**, Vol. 23, pp. 13498–13511. arXiv, 2022.

[4] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In **International Journal of Computer Vision**, Vol. 123, pp. 32–73. Springer, 2017.

[5] Carlos T. Ishi, Chaoran Liu, Jani Even, and Norihiro Hagita. Hearing support system using environment sensor network. In **2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**, pp. 1275–1280, 2016.

[6] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[7] Hiroaki Sugiyama, Masahiro Mizukami, Tsunehiro Arimoto, Hiromi Narimatsu, Yuya Chiba, Hideharu Nakajima, and Toyomi Meguro. Empirical analysis of training strategies of transformer-based japanese chit-chat systems. In **2022 IEEE Spoken Language Technology Workshop (SLT)**, pp. 685–691. arXiv, 2023.

[8] Frank E. Ritter, Farnaz Tehranchi, and Jacob D. Oury. Act-r: A cognitive architecture for modeling cognition. **WIREs Cognitive Science**, Vol. 10, No. 3, p. e1488, 2019.

[9] John E. Laird. Introduction to soar, 2022.

[10] William G. Kennedy, Magdalena D. Bugajska, Matthew Marge, William Adams, Benjamin R. Fransen, Dennis Perzanowski, Alan C. Schultz, and J. Gregory Trafton. Spatial representation and reasoning for human-robot collaboration. In **Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2**, AAAI'07, p. 1554–1559, Vancouver, British Columbia, Canada, 2007. AAAI Press.

[11] John E. Laird, Keegan R. Kinkade, Shiwali Mohan, and Joseph Zhen Ying Xu. Cognitive robotics using the soar cognitive architecture. In **CogRob@AAAI**, 2012.

[12] Adriana Bono, Agnese Augello, Giovanni Pilato, Filippo Vella, and Salvatore Gaglio. An act-r based humanoid social robot to manage storytelling activities. **Robotics**, Vol. 9, No. 2, 2020.

[13] Sonja Stange, Teena Hassan, Florian Schröder, Jacqueline Konkol, and Stefan Kopp. Self-explaining social robots: An explainable behavior generation architecture for human-robot interaction. **Frontiers in Artificial Intelligence**, Vol. 5, , 2022.

[14] Richard C. Atkinson and Richard M. Shiffrin. Human memory: A proposed system and its control processes. In Kenneth W. Spence and Janet Taylor Spence, editors, **The psychology of learning and motivation.**, Vol. 2 of **Psychology of Learning and Motivation**, pp. 89–195. Academic Press, New York, 1968.

[15] Hermann Ebbinghaus. **Memory: A contribution to experimental psychology**. No. 3. Teachers College, Columbia University, 1913. Translated by Henry A. Ruger and Clara E. Bussenius.

[16] M. Meeter, J. M. J. Murre, and S. M. J. Janssen. Remembering the news: Modeling retention data from a study with 14,000 participants. **Memory & Cognition**, Vol. 33, pp. 793–810, 2005.

[17] Kai Ming Ting. **Precision and Recall**, pp. 781–781. Springer US, Boston, MA, 2010.

[18] Claude Sammut and Geoffrey I. Webb, editors. **F1-Measure**, pp. 397–397. Springer US, Boston, MA, 2010.