AISB 2011

Human Memory for Artificial Agents

Editors:
Dimitar Kazakov &
George Tsoulas

THE UNIVERSITY of York

Foreword from the Convention Chairs

The AISB'11 call for symposium proposals particularly encouraged events drawing more strongly on the cognitive science aspect of the AISB remit. The result is a coherent programme with a very strong interdisciplinary character, which is also matched in the choice of plenary speakers. The three symposia looking at the interaction between Computing and Philosophy, the prospect of machine consciousness and the quest for a new, comprehensive intelligence test, form a coherent unit where the eternal questions of who we are and what makes us so are asked from a dual Human-Machine perspective. The Symposia on Active Vision, Computational Models of Cognitive Development and Human Memory for Artificial Agents demonstrate how better understanding of the nature and basis of cognitive processes can advance work on Artificial Intelligence and, inversely, how computational models of these processes can help better to understand them. The prominent multi-agent design and modelling paradigm links the Symposium on Social Networks and Multi-agent Systems with the one on AI and Games. Finally, the Symposium on Learning Language Models from Multilingual Corpora, which brings together some of the first attempts in this area, can also be seen through the prism of such a general notion in Philosophy and Linguistics as semiosis, and the dual role of sign and interpretant that text plays in translations.

We are delighted that after another ten successful years in its long history, the AISB convention is returning to the University of York. The 2011 convention takes place on the brand-new Heslington East campus, the result of a multi-million pound expansion that is now the new home of the Department of Computer Science, and hosts the Excellence Hub for Yorkshire and Humber, a new incubator for interdisciplinary research and interaction between academia and industry. The last few years have seen a strong involvement of the Computer Science Department in such interdisciplinary collaboration through the York Centre for Complex Systems Analysis (YCCSA), and we hope that this convention will provide a boost for more synergy between York departments, with other institutions conducting AI-related research in the region, and beyond. As the programme shows, we have also made an effort to promote cooperation with industry and use the convention to support school outreach. The convention format makes it perfect for establishing dialogue and collaboration in new areas of research, as well as across disciplines, and we hope that this year, it will play again this role to the full. We want to thank everyone who has contributed to it or otherwise made this event possible and wish all participants a fruitful and enjoyable time in York.

Dimitar Kazakov and George Tsoulas

Proceeding of The 2nd Symposium on Human Memory for Artificial Agents in AISB'11
Wan Ching Ho, Mei Yii Lim and Cyril Brom (eds.)

ISBN 1 978-1-908187-04-8

Published by the Society for the Study of Artificial Intelligence and the Simulation of Behaviour (AISB)
Printed by the University of York, UK

Table of Contents

Introduction: The Second Symposium on Human Memory for Artificial Agents

Wan Ching Ho[1], Mei Yii Lim[2] and Cyril Brom[3]

INTRODUCTION

Back in AISB 2010, the 1st Symposium on Human Memory for Artificial Agents brought together researchers from the fields of cognitive science, artificial intelligence, and the social sciences to discuss important aspects of human memory suitable to be modelled in artificial intelligent agents. Through papers presented in the symposium and various interesting discussions, it revealed the potential of human memories in modelling artificial cognition and social processes. Therefore we hope more scientific contributions can be made to investigate the interactions between its essential components such as short-term, working memories, semantic knowledge and episodic experiences among others. Therefore this symposium aims to gather interdisciplinary perspectives on the above issues and review work done so far to achieve a better understanding of which, when and how human-like memory can contribute to artificial agents modelling.

Since decades ago the idea of creating computational representation of experience for agents has been mentioned often in cognitive modelling literatures. On one hand, researchers in the early 90's argued, in the context of the Turing Test, that it is questionable whether any computer in the future can pass the Test without the ability to experience. On the other hand, embodied AI emphasizes the on-going interaction between agents and their environment, in which object representation evolves from the experience of the agents with these objects. Here the term "experience" is not as defined in machine learning, but as similar to the whole cognitive concept of human "organic" memory, e.g. events attributed with "emotion" and "meaning". It includes a range of cognitive processes that our memory operates effortlessly: perceiving, encoding, storing, retrieving, generalising and forgetting of events.

Up to date, various research projects have attempted to create agents that are more natural, believable and behave in human plausible ways; however, memory components in these models are rather static and loosely connected to each other. Another direction which has captured a lot of attention is the influence of emotion in long-term episodic memory. It is important to identify the possible ways of integrating various known emotion models to artificial agents with computational human memory, particularly those designed for social interactions with human users. Additionally, many existing models do not take into consideration the bio-mechanisms of human memory operations such as those involved in retrieval and forgetting processes.

Some recent research shows that artificial agents equipped with a subset of the above listed human memory processes are perceived as more natural and have the potential of improving human-agent interaction. Consistent with these findings, we envision that the existence of more comprehensive human-like memory processes will allow artificial agents to maintain behaviour coherence and plausibility, thus may lead to the establishment of longer term interaction/relationship with humans.

This year 5 short-papers (the length of which has been extended up to 5 pages) and 3 long-papers (8 pages) have been accepted for presentations in the symposium. Here the nature of long-paper is to allow the completeness of research concept to be conveyed in the paper. Each contribution received three reviews which were made by the following program committee members (listed by surname):

- Heuvelink Annerieke, TNO Defence
- Cyril Brom, Charles University Prague (co-chair)
- Joanna Bryson, University of Bath
- Nate Derbinsky, University of Michigan
- Sibylle Enz, University of Bamberg
- Stan Franklin, University of Memphis
- Wan Ching Ho, University of Hertfordshire (co-chair)
- Mei Yii Lim, Heriot-Watt University (co-chair)
- Nikolaos Mavridis, United Arab Emirates University
- Andrew Nuxoll, University of Portland
- Christopher Peters, Coventry University
- Debbie Richards, Macquarie University
- Alexei Samsonovich, George Mason University
- Holger Schultheis, University of Bremen
- Dan Tecuci, University of Texas

ACKNOWLEDGEMENT

This symposium was partially supported by the European Commission (EC) and is currently funded by the EU FP7 ICT-215554 project LIREC (Living with Robots and Interactive Companions).

[1] Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK. Email: W.C. Ho@herts.ac.uk

[2] Computer Science, Heriot-Watt University, Riccarton, EH14 4AS, UK. Email: M.Lim@hw.ac.uk

[3] Software and Computer Science Education, Charles University in Prague. Email: brom@ksvi.mff.cuni.cz

A Memory Structure that Gives Meaning to the Notions of Knowledge and Belief

José Ferreira de Castro [1]

Abstract. Beliefs are usually represented with modal operators in rule-based systems, or with probabilities in probabilistic frameworks. This paper explains how the concepts of knowledge and belief emerge from a simple memory architecture of cinematic records.

1 INTRODUCTION

Since 2000 I have been working on a machine for a unified theory of cognition[13, 12], called the M-Logic Machine. I have not published much about it [3, 4]. Among many other interesting aspects, the machine implements two simple but effective ideas: cinematic memories and dominant thoughts.

Consider birds and penguins. If we are asked about birds in general, the first image coming to our minds may be a flock of flying birds. Therefore birds fly. If we are asked about penguins, we shall recall an image of penguins - birds that do not fly. Dominant thoughts - the first images coming up to our minds - defy strict logic coherence. This is not easily captured by rule-based systems. We need to explicitly indicate all possible exceptions.

Probabilistic frameworks do better. Dominance should favour predictive accuracy, meaning giving correct answers as often as possible. I looked for an alternative approach avoiding the mysterious notion of probability. The basic idea is easy to grasp. Imagine we have a tray with a pile of documents. When we look for an answer, we simply go through the documents, starting from the document at the top of the pile. When we find a document that gives us a satisfactory answer, we put it back on top of the pile. If we find no answers after going through the pile, we leave the pile as it is. It's easy to see that the documents that were most often found satisfactory will tend to be found first, dominating (hiding) the documents that were seldom satisfactory. There is no need to resort to the notion of probability or keep statistics of any sort. The M-Logic Machine bets on dominant thoughts. The dominance hierarchy reflects the relative frequency of satisfaction. Notice that this approach is quite different from case-based learning [1]. Case-based learning uses a similarity function to generate predictions based on *all* the previously recorded cases. The M-Logic Machine chooses the dominant case. It's much simpler, and faster.

Regarding cinematic memories, it is obvious that the documents we handle in the tray are not like photographs, but rather like short films. For survival purposes, knowing how things evolve is much more important than knowing how things are. The movies in our heads allow us to imagine dominant continuations for the current situation. Because the current situation is specified by a sequence of events, the number of state variables used for prediction can be small. As the machine learns, successful motor-sensory sequences will tend to be replayed, without the need to understand causal relations. This gives a straightforward explanation to the origin of superstitious policies [16].

In this setting, where all the machine's information about its environment is recorded from its life experiences, the role of hardwired reflexes is essential to provide basic survival skills. Avoiding a deadly cliff cannot be learned with in-life experience.

An influential paper from Elman [6] discarded the use of cinematic memories, arguing they were not a good idea to model cognition. The AI mainstream research followed other directions. In this article I will show that in a non-verbal sensory-motor approach the use of cinematic memories is worth considering.

Among many other things, the M-Logic Machine architecture gives semantics to the notions of knowledge and belief. These notions emerge from specific memory configurations of the machine. In rule-based systems these notions are usually handled with modal operators that satisfy some axioms. The creator of a rule database for a given domain must indicate from the start what is to be considered knowledge or belief. In Bayesian probabilistic approaches (a recent example is Mavridis' proposal of Grounded Situation Models [10]) the initial credences must also be defined by the end-user. In both cases the environment must be previously modelled. Those solutions clearly lack autonomy. We shall now see in some detail the M-Logic Machine fully autonomous solution. A generic mathematical description is presented in section 2, explaining the basic MLM functional structure, while section 3 presents a simple example that details some of the MLM processes.

In what follows, it is important to remember that the MLM approach does not try to build a machine that talks to humans or emulates human cognition. It tackles the problem from the other end: If we start wiring neurons one to another, what are the minimal structures needed, from where notions like knowledge and belief can emerge? Just as Turing machines clarified the vague idea of *algorithm*, the MLM allows a clarification of *knowledge* and *belief* concepts.

2 AGENTS THAT KNOW AND BELIEVE

Let us start from the usual definition of an **agent** found, for instance, in [14, p. 32]: *an agent is an entity that perceives its environment through sensors and acts upon it through actuators*. A possible representation for this understanding [5, pp. 39-40] is to consider a set of environment states $E = \{e_1, e_2 \ldots\}$, a set of the agent's perceptive states $P = \{p_1, p_2 \ldots\}$ and a set of the agent's action states $A = \{a_1, a_2 \ldots\}$. The set of all possible sequences of environment states is noted E^*, and the set of all possible sequences of perceptive

[1] CENTRIA, FCT-UNL Portugal, email: CastroJFGF@gmail.com

states is noted P^*. We can define an agent g as a function:

$$g : E^* \to A \tag{1}$$

In a learning agent the g function will evolve in time as the agent learns more. The agent has no direct access to the environment states. The environment is known by means of its perceptive abilities, and the actions are based on these perceptions. The g function can be rewritten to include explicitly the function composition $g = perception * action$[2].

$$perception \quad : \quad E \to P \tag{2}$$

$$action \quad : \quad P^* \to A \tag{3}$$

The effect of the agent's actions on the environment (from the agent g point of view) is given mathematically by:

$$environment_g : E \times A \to \mathcal{P}(E) \tag{4}$$

where $\mathcal{P}(E)$ denotes the power set of E. Although only one state is reached at each time step, at different time steps different states can result from the same environment state and agent action. This mathematical formulation expresses the environment non-determinism, as seen by the agent. In time there is a unique sequence of environment states and actions, called the agent's *objective* history of action-effect states:

$$\mathbf{h}_g : e0 \to^{a0} e1 \to^{a1} e2 \to \dots \tag{5}$$

The M-Logic Machine makes some ontological assumptions regarding the agent and its environment:

- All meaningful thoughts are linked to measurements. For instance, consider two instruments that identify squares and circles. Since there is no instrument to identify a "square circle", a "square circle" is a *verbal fiction*. Also, a born blind human is not expected to dream visual images [9].
- There is no reality beyond measurements. In simple terms, we do not measure things. The measurement is the thing. The set of measurement states resulting from a given set I of measurement instruments is noted M. All agent's perceptions are built from the agent's set of measurement states M.
  For definiteness, all measurement instruments are assumed to be **classification instruments**. This simply means there is a set of distinct possible outputs $O = \{o_{\mathrm{j}} : \mathrm{j} \in \{1 \dots \mathrm{k}\}\}$ for each measurement instrument. All output pulses are assumed to be similar, and no temporal meaning is assigned to the frequency of pulses. This requires different measurement outputs to be sent to distinct output points. With this understanding, *deep belief networks* [2] are instances of measurement instruments.

With these assumptions we do without the set E, and work only with M and P. The environment function (4) is discarded. Environments and sensors are not explicitly modelled. The agent's perception function becomes:

$$perception : M \to P \tag{6}$$

The transient outputs recorded inside a given sensory-motor time interval are called simultaneous. Perceptions are records obtained from the simultaneous transient measurement outputs of the agent.

The $perception$ function therefore plays two important roles:

- It binds several events as "simultaneous". Measurement output pulses from different instruments are seldom synchronous. On the contrary, different instruments usually take different times to complete their measurements.
- It maps measurement output points to memory locations.

Perceptions can be ordered in space to reflect the order in time of measurement outputs. These spatially ordered structures are called **cinematic memories**. A **binding memory** is the place where a set of compatible measurements is integrated to produce elements of P. It is used to compose a present-moment crisp image.

The agent detects its own actions through a specific subset[3] of measurement instruments $I.a \subset I$ defining a set of measurement states for actions $M.a = \{m.a_1, m.a_2 \dots\} \subset M$. The corresponding perceptions are given by

$$perception : M.a \to P.a \tag{7}$$

As a general principle, the signals that trigger the action are not measured, but rather some physical manifestation of the action. For instance, human voluntary muscles are crossed by nerves that detect the muscle change in length, and the rate of change. The nervous pulses that generate the muscle stretching are not measured. This means that actions are perceived (and therefore known) after they are performed [8]. Similarly, environment ($P.e$) perceptions are generated at each agent's history step. The agent's *objective* history in (5) thus becomes a perceptive history:

$$\mathbf{h}_P : \ p.e0 \to^{p.a0} p.e1 \to^{p.a1} p.e2 \to \dots \tag{8}$$

This agent's history is now fully subjective. Objective environment states and actions are no longer included. Each sensory-motor step $p.eN \to^{p.aN}$ of the perceptive history is called a **frame** and is noted f. The set of all frames is noted F. This set is agent-specific. It relies on the sensory abilities of the agent.

Since the M-Logic Machine works with sequences of perceptive states, the **cinematic records**, some notation for state sequences is needed. The set of frame sequences[4] is noted F^*. $[F^*]$ indicates the set of *continuous* sequences bounded in time. Continuous and bounded sequences of frames are called **scenes**. Scenes can be built as list structures, adding frames to the top of the list.

A maximum number n of frames in the $[F^*]$ set of scenes is indicated with $[F^{\mathrm{n}}]$. This means no new frame will be added to the top of the list when the size limit is reached. Otherwise the size of scenes in $[F^*]$ is allowed to grow indefinitely. Alternatively, a new frame added to the top of the list will push out the frame at the bottom when the size limit is reached. This process is indicated with $[F^{\vec{\mathrm{n}}}]$, and these scenes are called **scrolling scene memories**.

$|[F^*]$ indicates that the most recent frame of each scene in $|[F^*]$ refers to the present moment. The set of present moment frames is noted F_π (a present moment individual frame is noted f_π). The sets of upcoming and prior frames are noted F_π^+ and F_π^-, respectively (the corresponding individual frames are noted f_π^+ and f_π^-).

A specific subset of measurement instruments was considered in (7) to identify actions. In general, any subset of measurement instruments $I.\mu \subset I$ generates a reduced set of measurement states $M.\mu = \{(m.\mu)_1, (m.\mu)_2 \dots\} \subset M$. This reduction is controlled by the $orientation$ function of the machine:

$$orientation : |[F^{\vec{2}}] \to \{\mu\} \tag{9}$$

[2] The usual mathematical notation is rather $g = action \circ perception$

[3] Suffixes are used to discriminate instrument subsets.

[4] In what follows, the same notation conventions apply to any reduced perceptive history $F.\mu^*$

where $|[F^{\vec{2}}]$ refers to a binding memory and $\{\mu\}$ is the set of sensory modes μ (the specifier μ is called a **sensory mode**). $|[F^{\vec{2}}]$ is the minimal structure that allows the MLM to detect the rise of important signals in any of its sensory dimensions.

The corresponding perceptions are given by

$$perception : M.\mu \rightarrow P.\mu \tag{10}$$

The $orientation$ and $perception$ functions together define the measurements that are included in the frames and their specific location inside the frames. A sequence of reduced frames recorded from a reduced set $M.\mu$ of measurement states is noted $F.\mu^*$.

In a sensory-motor learning setting, the agent sensory mode should include instruments to perceive relevant environment features and its own actions. In pure reflex settings, perception of motor actions is irrelevant.

Records for two different sensory modes, say $P.\mu_1$ and $P.\mu_1$ are not distinguished by their memory content. The distinction is made using two distinct brain regions to place these records.

The $perception$ and $orientation$ functions are vital aspects of any autonomous agent. On this subject, the work of Sokolov is worth reading [18].

Both binding memories and short-term memories are scrolling scene memories. But a binding memory is only two frames deep $|[F^{\vec{2}}]$, while a short-term memory is recorded after the $orientation$ function is applied, and is therefore of type $|[F.\mu^{\vec{n}}]$.

The set of multiple scenes of maximum length n separated by gaps, keeping the scenes in chronological order, is written $[F^{n}\|^*$. These are called **tales**. Scenes in tales are allowed a maximum length n. Tales are generated by the $record$ function. They are built from the short-term memory and placed in the agent's long-term memory:

$$record : |[F.\mu^*] \times |[\rho^*] \rightarrow [F.\mu^{n}\|^* \tag{11}$$

where $|[\rho^*]$ represents a sequence of **record modes** $\rho = \{\text{on}, \text{off}\}$ up to the present moment. The record mode at each moment is given by:

$$record\_mode : |[F.\mu^{\vec{n}}] \rightarrow \rho \tag{12}$$

While the $orientation$ function defines what is being recorded, the $record$ function uses the short-term memory to define when the recording of a new scene takes place in the long-term memory. As the long-term memories are built up in stable environments, there is less need to record.

Each distinct sensory mode requires a distinct long-term memory (noted $LTM.\mu$). Tales can be understood as lists of scenes, with a new scene added at the top of the list. In general, the number of frames in the tale scenes is not constant. A tale built up to the latest recording session (not necessarily the present moment) is noted $|[F^{n}\|^*$.

The set of cinematic records built from tales with the scenes's chronological order possibly shuffled is indicated with $[F^{n}\|^*_{\rightleftharpoons}]$. These structures are no longer real tales, although the scenes keep their internal chronological order intact. They are called **scene dominance records**. The scene shuffling is defined by a $dominance.s$ function:

$$dominance.s : [F.\mu^{n}\|^*_{\rightleftharpoons}] \times D.\mu \times \mathcal{S} \rightarrow [F.\mu^{n}\|^*_{\rightleftharpoons}] \tag{13}$$

where D is the set of dominant scenes that can be taken from $[F^{n}\|^*_{\rightleftharpoons}]$, and $\mathcal{S}$ classifies the predictive success of D. The agent action function (3) is redefined as follows:

$$action.r \quad : \quad M \rightarrow A.r \tag{14}$$

$$action.cr \quad : \quad M.\mu_{cr} \times [F.\mu^{p}_{cr}\|^{\pm} \rightarrow A.cr \tag{15}$$

$$action.v \quad : \quad \mathcal{H} \times |[F.\mu^{\vec{m}}] \times [F.\mu^{n}\|^*_{\rightleftharpoons}] \rightarrow A.v \tag{16}$$

The functions $action.r$ and $action.cr$ are the reflex and conditioned reflex functions, respectively. In a complex agent, many reflexes coexist. $A.r$ and $A.cr$ can easily coexist because the conditioned reflex simply anticipates (triggering a little sooner) the reflex action. The relevant scenes are stored in dedicated tales $[F.\mu^{p}_{cr}\|^{\pm}$ with specific sensory modes μ_{cr}. The generation mechanism simulates Hebbian learning. Scenes are added or taken out of the tales according to their predictive success. No dominance is used. Instead, a minimal number of instances is required to trigger the conditioned reflex. The machine thus learns by repetition of a scene pattern.

The last function $action.v$ describes voluntary actions based on beliefs. $\mathcal{H}$ is the set of **decision heuristics** of the agent. A decision heuristic defines the aspects of the past and future that are being searched in the records. Heuristics are transversal to the sensory modes. It's simply a search for patterns in the record sequences, in the spirit of [7]. This requires all reduced cinematic records to have a similar frame structure. Besides, the number of used sensory modes must be small. The agent cannot afford a huge number of memory areas, one for each $LTM.\mu$. Such limiting requires a small number of classifier instruments assigned to each frame (something already noticed by Miller in [11]). There are not many effective sensory modes for a given environment. The best modes can be found with evolutionary learning.

In (16), the scenes $|[F.\mu^{\vec{m}}]$ and the tales $[F.\mu^{n}\|^*_{\rightleftharpoons}]$ are stored in the agent's short-term and long-term memories, respectively. The short-term memory tells the agent about the recent past, defining the current cinematic situation. The dominant scene in the long-term shuffled memory provides a (hopefully true) continuation for the current situation. This explains the need for both types of memory. Coexistence of $A.v$ with reflex actions will depend on the sensory mode μ_v.

The voluntary action defined in (16) can be understood as a composition of a question, a believed answer, and an action trigger:

$$action.v : question.v * answer.v.b * trigger.v \tag{17}$$

with the following mathematical definitions:

$$question.v \quad : \quad \mathcal{H} \times |[F.\mu^{\vec{m}}] \rightarrow \mathcal{Q}.\mu \tag{18}$$

$$answer.v.b \quad : \quad \mathcal{Q}.\mu \times [F.\mu^{n}\|^*_{\rightleftharpoons}] \rightarrow \mathcal{A}^{(\mathcal{B})} \tag{19}$$

$$trigger.v \quad : \quad \mathcal{A}^{(\mathcal{B})} \rightarrow A.v \tag{20}$$

where $\mathcal{Q}$ is the set of questions and $\mathcal{A}$ the set of answers. $\mathcal{A}^{(B)}$ is the same as $D.\mu$ in (13). The function $trigger.v$ triggers the expected action perceptions included in the believed answer upcoming frame f^+_π. If no action perception is included in the believed answer, no voluntary action is triggered. Ultimately, voluntary actions are triggered because the agent believes they can be done.

Propositions are questions. Questions usually include a **context** and an **interrogation**. In the Prolog implementation of the MLM [5], questions are queries. Contexts are constants and interrogations are variables in the queries. There are only constants in the consulted memories, the records of measurement outputs. An answer in $\mathcal{A}$ is the first match found in the agent's memories for a given question. For instance, if we represent cinematic memories with lists, the list is scanned from head to tail until a match is found for the context.

[5] The Prolog source code can be found in the authors homepage https://sites.google.com/site/josefgfcastro/.

The contexts are matched and the interrogation variables are unified with the corresponding constants. The first match therefore gives the **dominant answer**. Short-term predictive answers are found inside a dominant scene of $D.\mu$, the same that is used in definition (13). The heuristic that is actually used in (16) and (18) is the **dominant heuristic**, subjected at each motor-sensory step to a shuffling process defined by a $dominance.h$ function. Note that the heuristics are transversal to the sensory modes. This allows meta-learning: the heuristic dominance hierarchy will be kept when the sensory mode changes.

$\mathcal{A}^{(\mathcal{B})}$ answers are believed. This is because they are found in a non-factual (shuffled) long-term memory. Known answers $\mathcal{A}^{(\mathcal{K})}$ are given by a series of $answer.v.k$ functions:

$$answer.v.k.stm \quad : \quad \mathcal{Q}.\mu_v \times |[H.\mu_v]^{\vec{n}} \rightarrow \mathcal{A}^{(\mathcal{K})} \qquad (21)$$

$$answer.v.k.ltm \quad : \quad \mathcal{Q}.\mu_v \times |[H.\mu_v \| H.\mu_v] \rightarrow \mathcal{A}^{(\mathcal{K})} \qquad (22)$$

Knowledge answers do not trigger voluntary actions. The present and the past perceptions can no longer be acted upon. Nevertheless, it's the $answer.v.k$ process applied to the present-moment frame, trying to know the believed answer for the present-moment action, that actually triggers the voluntary action in definition (20). The two knowledge functions (21) and (22) are run in sequence. First, $answer.v.k.stm$ consults the short-term memory, and then, if needed, $answer.v.k.ltm$ consults the long-term memory. In words, we know something when the answer can be found in our memories of facts. We can only know the present and the past. We can believe the future and the past, but not the crisp present moment found in the binding memory. The crisp present moment is felt, not known. When knowledge and belief about the past overlaps, we may get confused. Specially because the scenes in the shuffled memories still appear as factual. These results are summarized in Table 1.

Table 1. The MLM epistemic states.

$\mathcal{A}^{(\mathcal{K})}$	$\mathcal{A}^{(\mathcal{B})}$	Temporal Range	Epistemic State
no	no	n.a.	none
no	yes	future, past	justified belief
yes	no	present, past	knowledge
yes	yes	past	knowledge

The aim of the $dominance$ function given in (13) is to provide the answer most often found correct for the current question. The predictive success result is given by a $success.pred$ function:

$$success.pred : (\mathcal{A}.f_\pi)^{(\mathcal{K})} \times (\mathcal{A}.f_\pi)^{(\mathcal{B})} \rightarrow \mathcal{S} \qquad (23)$$

This function compares the believed and known answers for the present-moment frame f_π. The belief answers were generated some time before the present-moment knowledge answers, and the comparison is made as soon as the relevant knowledge is generated. If the knowledge and belief answers do not match, the agent may feel surprised. This happens when the agent belief was not weakened by further scanning of the long-term memories. This belief gradation involves a simple additional memory scanning process to find a few sub-dominant belief answers.

The same evaluation $\mathcal{S}$ of the predictive success is used to update the heuristics dominance.

A M-Logic Machine memory scheme can be seen in Figure 1. In this scheme, the question (generated by the Q process) and the belief and knowledge answers (generated by the B and K processes, respectively) are gathered in a **Working Memory**. The $\mathrm{I_R}$ process provides the sub-dominant answers that generate belief gradations. Some other MLM interesting features were ignored in this article, because they are not important for our current purpose. There are, for instance, cinematic imagination processes that we not consider here. The classical work of Shepard in mental rotation [15] fits well in the cinematic imagination concept.

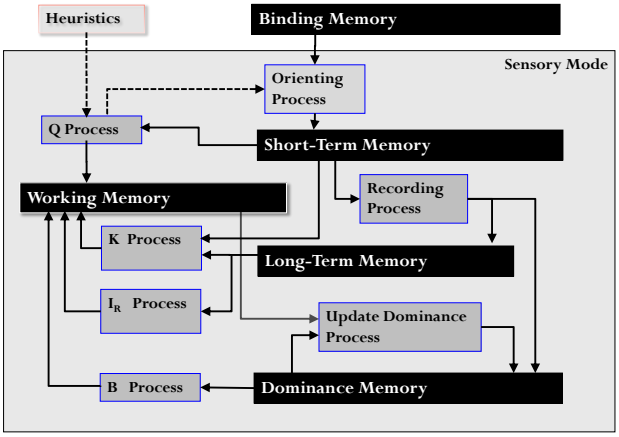

Figure 1. The MLM memory structure and processes in vigil mode

Figure 1 refers a vigil mode. There is also a sleep mode. All frames and scenes are alike in size, and therefore may contain information irrelevant to the finding of cinematic regularities. An internal cleaning process is performed during sleep mode. Memories are simplified, but become more useful for prediction.

The necessary limitations of cinematic memories regarding content and duration may be further justified by physical constraints. If we accept the hypothesis that cinematic memories are recorded in the cortical columns, the number of measurements in the scenes and their duration is necessarily limited.

3 A MATCHING-PENNIES MACHINE

Cinematic reasoning only becomes important in situations that cannot be solved directly by reflex actions. Taking advantage of cinematic regularities in incoming sequences is the most obvious example of a problem that cannot be tackled by pure reflexes. Let us therefore consider a simple version of the matching-pennies problem. The MLM plays against a player X. In each turn, both players must choose simultaneously to produce one or two coins. The MLM wins if the number of coins produced by each player is found to be equal, and loses if the number of coins is found to be different. According to the definition given in (8), the MLM perceptive history is represented with

$$\mathbf{h}_P : \; p.e0 \rightarrow^{p.a0} p.e1 \rightarrow^{p.a1} p.e2 \rightarrow \ldots$$

In order to learn the cinematic regularities, the MLM needs at least a classification instrument to detect its own relevant actions. The measurement results for producing one or two coins are written $1c$ and $2c$, respectively. This corresponds to the $p.a$ perceptions in the perceptive history. The MLM needs also to detect the two relevant environment states, i.e. the number of coins being equal or different. This is written eq and df, respectively. This corresponds to the $p.e$ perceptions. All these measurements are assigned to locations in the

frames according to the $perception$ and $orientation$ functions. Remember that different measurement results are sent to distinct output points. The assignment can be seen as the wiring between the measurement instruments and the frames. Let us assume the frame assignment $\star f$ to be:

$$\star f = \begin{bmatrix} eq & df \\ 1c & 2c \end{bmatrix} \tag{24}$$

Each line in the frame matrix is called a **frame channel**. A single instrument is assigned to each frame channel. The columns in each frame channel will hold the different measurement results of classifier instruments. In (24) the two frame channels record a sensory-motor sequence.

Now, the MLM heuristics consult the frame locations without any information about the meaning of their content. The heuristics try to bring up recorded pulses in a selected location of the frame (the pl "pleasure" location), and avoid the presence of recorded pulses in another location of the frame (the pn "pain" location). The heuristics also assume that the actions triggered by their choices are also recorded in two other locations, $a1$ and $a2$. Let us assume the heuristic frame template $\star f_{\mathcal{H}}$ to be:

$$\star f_{\mathcal{H}} = \begin{bmatrix} pl & pn \\ a1 & a2 \end{bmatrix} \tag{25}$$

Comparing (24) and (25) we see that the pleasure pl and pain pn of the MLM are situated in the game. The heuristic finds a recorded pulse in the pleasure value location when the MLM wins, and finds a recorded pulse in the pain value location when the MLM loses. This assignment is the first step to define a MLM "goal". From now on we shall refer to the eq location as pl and the df location as pn.

These remarks highlight two important aspects of the M-Logic approach:

- Different measurement outputs and records are distinguished by their different locations. For instance, the rate of action potentials is not used as a classification criterion. Ultimately, all measurement results that are used are those assigned to single neurons.
- All meaning is relational and dynamic. Each location is accessed by different processes that consult its state and manipulate its contents. It's this set of processes that gives meaning to each location. This totally eliminates the symbol grounding problem.

The notion of pleasure and pain is therefore relational and dynamic. Pain and pleasure can refer to any measurement that triggers specific motor responses in the machine *by means of a heuristic*. In the absence of a hostile environment[6], the frame assignment is irrelevant for the machine's survival. The machine could as well take pleasure in losing the game, with no practical consequence. In a hostile environment, on the contrary, the "wrong" assignments are the ones that kill the agent, while the "right" assignments are the ones that promote its survival. In this case it might seem reasonable to link nociceptors to the pain locations. But it's probably more effective to link them directly to evasive reflexes. Quite often there is no time to learn. In the MLM framework, notions like pleasure and pain, right and wrong, good and bad, goal and purpose, result from evolutionary learning.

The two matrix lines in (24) are the minimal number of frame channels required for any reinforcement learning problem. One channel (the driving channel) has two locations that records the "pleasure" and "pain" tags. A second channel (the motor channel) records the the motor choices of the MLM (i.e. the motor commands to display one or two coins), that will be confronted with the choice of player X in the next turn. Assuming reliable sensors and motors, the sensory information regarding the number of coins actually produced is redundant. This can be recorded in a third channel (the state channel) related, for instance, to visual information.

Some general considerations about the game may fit here. If an unbalanced coin is flipped and the probability is biased towards one of the possible results, say tails, the best winning strategy is to bet on the most frequent result (i.e. keep betting tails). The dominance mechanism can easily handle this problem. If, on the contrary, the probability of getting heads or tails is identical, it may still be possible to make some valuable predictions. Certain *sequences* of results may be more probable than others. A trivial example is a sequence where heads and tails always alternate. This is where cinematic records become useful. The machine can look for dominant sequences. Another valuable strategy is to find a correlation in time with some other event. For instance, if the result of a coin is always equal to the previous result of another coin, we are able to guess the result after we discovered the temporal relation among the two coins, even when the first result is unpredictable. All these strategies are implemented through the corresponding heuristics. The current MLM dominance mechanism integrates in a single procedure the search for these three types of regularities. It tends to keep betting on the one that is most frequently rewarding. This is achieved without the need to evaluate or manipulate probabilities.

Sometimes the MLM cannot generate actions based on beliefs. Whenever possible, the MLM motor actions are triggered according to the guesses of a belief generator for the immediate future, according to a given inference heuristic, say $\mathcal{H}_1$. If the belief generator fails to provide a guess for a motor action, a random generator triggers a motor action anyway. This corresponds to an exploratory instinct.

For definiteness, let us assume that X produces two coins every fifty turns, producing one coin in the remaining turns. Let us also assume that the short-term memory of the MLM is recording a single frame per turn, in two micro-steps. This will happen in a natural way if the machine sensory-motor cycle is fully dedicated to the play. In the first micro-step it gets the current situation regarding the equal or different number of coins and in the second micro-step it records the sensed force related to the motor action triggered. Let us limit to six frames the scenes in the short-term memory ($|[F.\mu^{\vec{6}}]$). After six turns, with player X always choosing to produce one coin and the MLM playing randomly, the short-term memory (STM) of the MLM could contain, for instance, the following sequence of frames:

$$\mathrm{STM}_{t_6} : \left| \begin{matrix} pn \\ \_ \end{matrix} \right| \left| \begin{matrix} pl \\ 2c \end{matrix} \right| \left| \begin{matrix} pl \\ 1c \end{matrix} \right| \left| \begin{matrix} pl \\ 1c \end{matrix} \right| \left| \begin{matrix} pn \\ 1c \end{matrix} \right| \left| \begin{matrix} pl \\ 2c \end{matrix} \right| \tag{26}$$

In each frame we only indicate the frame position tag where a positive value is recorded. Since we are assuming classification instruments, only one frame location in each frame channel is expected to present a positive value. By convention, the MLM starts with an pl perception at time t_1. The most recent frame of this scene is to the left. In the most recent frame, only the first micro-step is completed, and we have only the current equal/different situation (in this case pn) resulting from the choice $2c$ recorded in the previous frame. This means the MLM presented two coins while X presented just one coin. Remember that the rule for X is to produce two coins every fifty turns, producing one coin in the remaining turns. The latest motor action at t_6, that will have an impact on the next frame evaluation, is still undefined. Since the maximum number of frames in a

[6] Hostility is measured by the time it takes an agent to be destroyed when its actions are chosen at random.

scene has been reached, every new frame recorded in the short-term memory will push into oblivion the oldest frame. Oblivion of frames is therefore reached suddenly, not by gradual decay of their components. This aspect is in agreement with some experimental evidence [19].

The MLM searches its scene dominance records to generate beliefs. A wise criterion is needed to record scenes from the short-term memory into the **scene dominance memories** (SDM). The criterion is the $record\_mode$ function described in (12). What is "wise" or not is also a result of evolutionary learning. The recording criterion also participates in defining the machine's "goal". Let us define the $record\_mode$ function as follows:

1. The STM-to-SDM record mode ρ switches from OFF to ON when a pn frame follows two consecutive pl frames. When ρ switches from OFF to ON, the first thing recorded in the STM-to-SDM buffer is the short-term memory content, in its totality. This gives the MLM some information about the recent past, prior to the recording.
2. The record mode ρ changes from ON to OFF after a certain size limit of the STM-to-SDM buffer is reached, say ten frames. At the ON-OFF transition the buffer content is moved to the SDM.

Let us call this recording criterion RC_1. Looking at the STM in (26) we see that this criterion finds a match in time t_6. We therefore start placing in the buffer a copy of the STM_{t_6} scene. As new frames are recorded in the STM, they are also added to the buffer, up to ten frames. When the recording mode gets back to OFF, the buffer content is moved to the SDM. We may have got, for instance:

$$\mathrm{STM}_{\mathrm{t}_{10}}: \left|\begin{array}{c|c|c|c|c|c} pn & pl & pl & pn & pn & pl \\ \_ & 2c & 1c & 1c & 2c & 2c \end{array}\right|$$

$$\mathrm{SDM}_{\mathrm{t}_{10}}: \left|\begin{array}{c|c|c|c|c|c|c|c|c|c} pn & pl & pl & pn & pn & pl & pl & pl & pn & pl \\ \_ & 2c & 1c & 1c & 2c & 2c & 1c & 1c & 1c & 2c \end{array}\right|$$

Now $\mathrm{SDM}_{\mathrm{t}_{10}}$ is available for prediction. Let us assume the following cinematic heuristic is being used by the $question.v$ function presented in (18):

$$\mathcal{H}_1: \left|\begin{array}{c|c} c(pl) & c(PP_\pi) \\ \_ & i(A) \end{array}\right| \tag{27}$$

Heuristics are general templates used by the MLM to generate questions. The $question.v$ process is quite simple: The value for the undefined context $c(PP_\pi)$ is searched in the present moment frame f_π of the STM. In this case we are looking in $STM_{t_{10}}$. Therefore $f_\pi = f_{10}$ (the leftmost frame), and we get $c(PP_\pi) = c(pn)$. The generated question is therefore

$$\mathcal{Q}: \left|\begin{array}{c|c} c(pl) & c(pn) \\ \_ & i(A_\pi) \end{array}\right| \tag{28}$$

The context of this question is a pl following a pn frame. The interrogation $i(A_\pi)$ is the action that hopefully generates that sequence. Since the past and the future cannot be acted upon, the action is to be generated in f_{10}.

Answers for the interrogation $i(A_\pi)$ are searched by three processes. The first two belong to the knowledge acquisition process $\mathcal{K}$ presented in (22). The third belongs to the belief generating process $\mathcal{B}$.

1. $answer.v.k.stm$ looks in the short-term memory, the $\mathrm{STM}_{\mathrm{t}_{10}}$. In this case it will necessarily fail, because part of the question context refers to the future. The STM does not include frames beyond f_π.
2. $answer.v.k.ltm$ looks in the long-term memory. In the current example this memory is not even considered. It is only necessary to recall episodic past information.
3. $answer.v.b$ looks in the scene dominance memory, the $\mathrm{SDM}_{\mathrm{t}_{10}}$. This process looks sequentially through the SDM. It scans the SDM scenes, following their dominance order. Each dominance scene is scanned from the most recent frame to the oldest, looking for a match to the question.

The questions and the answers are centralized in a working memory $\mathbf{W}$. At moment t_{10} there is only one scene in $\mathrm{SDM}_{\mathrm{t}_{10}}$, and the first match found provides the following $\mathbf{W}$ configuration:

$$\mathbf{W} = \left|\begin{array}{ll|l|l|} \mathcal{Q}: & c(pl) & c(pn) \\ & \_ & i(A_\pi) \\ \mathcal{K}: & \multicolumn{2}{l|}{\mathbf{nf}} \\ \mathcal{B}: & c(pl) & c(pn) \\ & \_ & i(1c) \end{array}\right|_{t_{10}} \tag{29}$$

This $\mathbf{W}$ configuration corresponds to a belief epistemic state. The knowledge acquisition process could not find an answer, and this is noted $\mathbf{nf}$.

The belief answer does not trigger directly a $1c$ action. The MLM can think of actions without actually performing them, because the belief generating process $\mathcal{B}$ is not directly linked to actuators. It just consults memories. Actuator actions can only be performed in the present moment, and only the knowledge acquisition process $\mathcal{K}$ can be directly linked to actuators. The voluntary motor action $1c$ is triggered when the $\mathcal{K}$ process looks for an answer to a question built from the belief trimmed to the present moment.

$$\mathbf{W} = \left|\begin{array}{ll|} \mathcal{Q}: & c(pn) \\ & c(1c_\pi) \\ \mathcal{K}: & \ldots \\ \mathcal{B}: & \ldots \end{array}\right|_{t_{10}^+}$$

For voluntary actions, a physical link must exist between the $\mathcal{K}$ process and the action measurements: the $\mathcal{K}$ process triggers the actuator that gives back the requested measurement. This necessary link is the most elementary form of procedural memory. The action is measured and recorded in $\mathrm{STM}_{\mathrm{t}_{10}^+}$, while nothing changes in $\mathrm{SDM}_{\mathrm{t}_{10}}$:

$$\mathrm{STM}_{\mathrm{t}_{10}^+}: \left|\begin{array}{c|c|c|c|c|c} pn & pl & pl & pn & pn & pl \\ 1c & 2c & 1c & 1c & 2c & 2c \end{array}\right|$$

$$\mathrm{SDM}_{\mathrm{t}_{10}}: \left|\begin{array}{c|c|c|c|c|c|c|c|c|c} pn & pl & pl & pn & pn & pl & pl & pl & pn & pl \\ \_ & 2c & 1c & 1c & 2c & 2c & 1c & 1c & 1c & 2c \end{array}\right|$$

This in turn provides the desired answer to the $\mathcal{K}$ process:

$$\mathbf{W} = \left|\begin{array}{ll|} \mathcal{Q}: & c(pn) \\ & c(1c_\pi) \\ \mathcal{K}: & c(pn) \\ & c(1c_\pi) \\ \mathcal{B}: & c(pn) \\ & c(1c_\pi) \end{array}\right|_{t_{10}^{++}}$$

This **W** configuration corresponds to a knowledge epistemic state (the belief is fully covered by knowledge). Besides, there is no surprise, since the belief produced is identical to the acquired knowledge.

We thus see that the MLM knows about its voluntary actions after they were triggered. Something similar to this perplexing situation is actually found in human brains [11].

We can also see that $\mathrm{SDM}_{\mathrm{t}_{10}}$ provides a micro-theory $\mathrm{MT}_{\mathrm{t}_{10}}$ for the $\mathcal{H}_1$ cinematic heuristic, giving two micro-rules (mR_1 and mR_2) for $\mathcal{H}_1$:

$$\mathrm{MT}_{\mathrm{t}_{10}} \quad \begin{array}{l} mR_1 : \left| \begin{array}{c|c} pl & pn \\ \_ & 1c \end{array} \right| \\ mR_2 : \left| \begin{array}{c|c} pl & pl \\ \_ & 1c \end{array} \right| \end{array}$$

In words, the micro-rules tell the MLM to select $1c$ whatever the present moment situation, in order to reach a pl next frame. Remember that only the first match found in $\mathrm{DDL}_{\mathrm{t}_{10}}$ (scanning scenes from the top of the scene list, and scanning scenes from the most recent to the oldest frame) is used to define the micro-rule. We can also note that the "implicit knowledge" (a list of scenes) recorded in $\mathrm{STM}_{\mathrm{t}_{10}}$ becomes "explicit knowledge" (a set of rules) when searched by a specific cinematic heuristic.

What happens next? First, the next frame is recorded in the short-term memory,

$$\mathrm{STM}_{\mathrm{t}_{11}} : \left| \begin{array}{c|c|c|c|c|c} pl & pn & pl & pl & pn & pn \\ \_ & 1c & 2c & 1c & 1c & 2c \end{array} \right| \tag{30}$$

Now the $success.pred$ function evaluates the success of the voluntary action. To do so, a query is generated where the $\mathcal{K}$ process can check the f_{11} frame against the pl prediction in (29). The corresponding working memory configuration becomes:

$$\mathbf{W} = \left| \begin{array}{ll} \mathcal{Q} : & \left| \begin{array}{c} i(pl_\pi) \\ \_ \end{array} \right| \\ \mathcal{K} : & \left| \begin{array}{c} i(pl_\pi) \\ \_ \end{array} \right| \\ \mathcal{B} : \ldots & \end{array} \right|_{t_{11}}$$

This working memory configuration corresponds to a knowledge epistemic state. The prediction was successful. The $dominance.s$ function in (13) takes the scene in $\mathrm{SDM}_{\mathrm{t}_{10}}$ that was used by the $\mathcal{B}$ process to produce the belief answer, and moves it to a higher or lower position relative to the other scenes in the SDM, according to its predictive accuracy. So far there is only one scene in the SDM, so the $dominance.s$ function does not reorder anything. Note that it is the micro-theory that is reordered, not the derived individual micro-rules.

After the checking process is completed, the full process starts again. From this moment onward, since the motor actions of the MLM are now defined by the belief generator, the MLM starts winning until the fiftieth turn arrives, and player X plays two coins. Until then, the record mode ρ stays OFF, according to the RC_1 recording criterion. Therefore the $\mathrm{SDM}_{\mathrm{t}_{10}}$ is the only source for micro-theories, and the $\mathrm{MT}_{\mathrm{t}_{10}}$ micro-theory is dominant. According to the RC_1 recording criterion, when a new pn occurs after two consecutive pl, the record mode rho is again set to ON. After four more turns it's switched back to OFF, and the buffer content is transferred to the SDM. Therefore, at time t_{54}, the short-term memory and SDM configuration become:

$$\mathrm{STM}_{\mathrm{t}_{54}} : \left| \begin{array}{c|c|c|c|c|c} pl & pl & pl & pl & pn & pl \\ \_ & 1c & 1c & 1c & 1c & 1c \end{array} \right|$$

$$\mathrm{SDM}_{\mathrm{t}_{54}} : \left\{ \begin{array}{l} \left| \begin{array}{c|c|c|c|c|c|c|c} pl & pl & pl & pl & pn & pl & pl & \ldots \\ \_ & 1c & 1c & 1c & 1c & 1c & 1c & \ldots \end{array} \right| \\ \\ \left| \begin{array}{c|c|c|c|c|c|c|c} pn & pl & pl & pn & pn & pl & pl & \ldots \\ \_ & 2c & 1c & 1c & 2c & 2c & 1c & \ldots \end{array} \right| \end{array} \right.$$

From now on the scene on top of $\mathrm{SDM}_{\mathrm{t}_{54}}$ is scanned first. It becomes the new source for a dominant micro-theory for $\mathcal{H}_1$, and the former $\mathrm{MT}_{\mathrm{t}_{10}}$ from $\mathrm{DDL}_{\mathrm{t}_{10}}$ is forgotten "by interference". But the new micro-rules provided by $\mathrm{DDL}_{\mathrm{t}_{54}}$ to the $\mathcal{H}_1$ heuristic are identical to mR_1 and mR_2, therefore the machine simply ignores the choice of X to play $2c$ from time to time. This is, by the way, the best strategy for MLM when the exact moment of the exception cannot be predicted.

At time t_{100} the scene on top of $\mathrm{SDM}_{\mathrm{t}_{54}}$ is pushed down in the list because it provided a false prediction at time t_{99}. But this will not change the micro-theory used, since the older scene (now on top) gives an identical micro-theory for the heuristic. At time t_{104} a new scene is recorded in the SDM, just like at time t_{54}, and the process goes on. Each failure starts a new recording, and more identical scenes are added to the top of the SDM. When the limit of the SDM capacity is reached, the scenes at the bottom start being pushed out to oblivion. This is the basic mechanism that implements the machine's reinforcement learning.

Let us now suppose that player X, instead of playing two coins every fifty turns, gets tired of losing and decides to invert its playing rule at time t_{100}, and starts producing two coins instead of one. From the configuration above, four turns after this new rule of player X is started, we get:

$$\mathrm{STM}_{\mathrm{t}_{104}} : \left| \begin{array}{c|c|c|c|c|c} pn & pn & pn & pn & pn & pl \\ \_ & 1c & 1c & 1c & 1c & 1c \end{array} \right|$$

$$\mathrm{SDM}_{\mathrm{t}_{104}} : \left\{ \begin{array}{l} \left| \begin{array}{c|c|c|c|c|c|c|c} pn & pn & pn & pn & pn & pl & pl & \ldots \\ \_ & 1c & 1c & 1c & 1c & 1c & 1c & \ldots \end{array} \right| \\ \\ \left| \begin{array}{c|c|c|c|c|c|c|c} pl & pl & pl & pl & pn & pl & pl & \ldots \\ \_ & 1c & 1c & 1c & 1c & 1c & 1c & \ldots \end{array} \right| \\ \\ \left| \begin{array}{c|c|c|c|c|c|c|c} pn & pl & pl & pn & pn & pl & pl & \ldots \\ \_ & 2c & 1c & 1c & 2c & 2c & 1c & \ldots \end{array} \right| \end{array} \right.$$

The new scene in the top of $\mathrm{SDM}_{\mathrm{t}_{104}}$ does not change the micro-rules for the $\mathcal{H}_1$ heuristic. It does not contain any pn to pl transition. Therefore, the two bottom scenes are used. They fail and are pushed down alternatively. This is so because we do not find in the SDM any indication that $2c$ now brings pleasure instead of pain. And nothing further will be recorded in the SDM while the MLM keeps using the RC_1 recording criterion.

This obvious interaction of all the MLM features, from clever senses to wise recording criteria, highlights the importance of **integrated intelligence**. This aspect was already noticed in the seventies by Newell [12]. It justified the search for unified theories of cognition.

To get out of the learning deadlock, the MLM needs to resume exploring. This can be obtained in several ways. The two following strategies are very simple and can be implemented simultaneously:

1. *Failure-Based Unlearning*. This can done erasing the scenes that support the failing micro-theories after a given number of consecutive failures (we can thus tune the machine's "patience in pain"). After the SDM cleaning process, we are back to random moves and the learning process starts again. Notice that, if player X takes a long time to shift its rule, many scenes like the new one in $\mathrm{DDL}_{\mathrm{t}_{54}}$ are added to the SDM, and the cleaning process takes longer. The machine sticks longer to theories that were repeatedly rewarding.
2. *Action Randomization*. Another simple strategy is to always allow some randomness in the motor actions triggered, even when a belief is successfully generated. This is what we may call the "keep-exploring principle". This idea is frequently used to escape the curse of local maximums in learning algorithms. The learning speed is increased if we add to the RC_1 recording rule another RC_2 rule that starts recording after two consecutive pn are followed by a pl. With these two rules, a $2c$ random choice soon brings to the SDM an adequate micro-theory for $\mathcal{H}_1$. Action randomization may adapt faster than failure unlearning to the new situation, but it will also bring pain to the machine.

In this learning setting, the heuristic $\mathcal{H}_1$ is good enough in two situations. First, when the rule used by player X has a periodicity of one or two turns, with a small amount of noise. Second, although the heuristic $\mathcal{H}_1$ is unable to detect periodicities over more than one or two turns, it may be still good enough when the rules with a periodicity of one or two turns are slowly alternated, giving time for the machine to unlearn and relearn good micro-rules for $\mathcal{H}_1$. Of course, the new rule must be learned fast enough to avoid compromising the machine's survival. This will depend on the initial survival assets of the MLM and how hostile the world is.

Beyond these two situations, other simple and fast heuristics (in the spirit of [7]) and recording strategies can be tried to cover as much as possible the deficiencies of $\mathcal{H}_1$. The final set of good heuristics is a product of evolutionary learning.

The current implementation of the M-Logic Machine that can be found in the authors' homepage is somewhat more complex than this very simple example just given. In it, frames use around ten channels to structure data. The searched patterns are four frames long, instead of just two. A dozen predictive heuristics are used, along with half-a-dozen recording criteria. Some heuristics look for global pleasure at scene level, searching for greater pleasure beyond immediate pain. This requires higher-level measurements to detect scene features. Most important, there are heuristics that will try to avoid pain, not just heuristics to look for pleasure. The set of predictive heuristics is subject to a dominance update mechanism similar to the one used for scenes in the SDM. When pleasure seeking heuristics fail repeatedly, pain avoidance heuristics dominate, and vice-versa.

Because of the fixed number of channels in all frames, irrelevant information is often recorded in the available channels of the cinematic memories. The MLM interrupts the vigil mode processes from time to time, shown in Figure 1, and starts a SDM cleaning process that greatly improves the inference abilities of the machine. It's an internally generated Hebbian learning process that strongly evokes dreaming. Cinematic memories become less accurate, but more useful in many situations.

Therefore, the MLM core architecture here presented easily scales out to new features. The ability for an architecture to scale out is emphasized by Sloman in [17]. As it is, the MLM can solve an interesting set of predictive inference problems. The set of problems is embedded in the public source code of the MLM.

4 CONCLUSION

The M-Logic Machine gives a concrete meaning to the notions of knowledge and belief. The meaning relies solely on the question-answer configurations found in the machine's working memory. This clarification can contribute to the progress of any research field that deals with these important concepts. Furthermore, it was shown how the MLM achieves learning in random hostile environments without resorting to statistical calculations. Only memory manipulations are involved. Most features of the MLM machinery can be randomly generated and tuned by evolutionary learning. In this sense it's a fully autonomous solution. The MLM architecture also shows the virtues of using cinematic memories as a basis for cognition. This approach is virtually absent from current AI research.

REFERENCES

[1] A. Aamodt and E. Plaza, 'Case-based reasoning: Foundational issues, methodological variations, and system approaches', *AICom - Artificial Intelligence Communications*, **7:1**, 39–59, (1994).

[2] Yoshua Bengio, 'Learning deep architectures for AI', *Foundations and Trends in Machine Learning*, **2**(1), 1–127, (2009). Also published as a book. Now Publishers, 2009.

[3] J. F. Castro, 'M-logic: Thinking with measurements and cinematic memories', in *Proceedings of the 2008 Conference on Human System Interactions*, pp. 633–638, (May 2008).

[4] J. F. Castro, 'Sub-rationality and cognitive driven cooperation', in *Proceedings of the 3rd International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems (ERLARS)*, eds., Josef Pauli Nils T Siebel and Yohannes Kassahun, pp. 53–57, (August 2010).

[5] E. Costa and A. Simões, *Inteligência Artificial: Fundamentos e Aplicações*, FCA, Editora de Informática, Lisboa, 2nd edn., 2008.

[6] J. L. Elman, 'Finding structure in time', *Cognitive Science*, **14**, 179–211, (1990).

[7] G. Gigerenzer, P. M. Todd, and Research A. B. C. Group, *Simple Heuristics That Make Us Smart*, Oxford University Press, New York, 1999.

[8] P. Haggard, 'Conscious intention and motor cognition', *Trends in Cognitive Sciences*, **9**(6), (2005).

[9] C. Hurovitz, S. Dunn, G. W. Domhoff, and H. Fiss, 'The dreams of blind men and women: A replication and extension of previous findings', *Dreaming*, **9**, 183–193, (1999).

[10] N. Mavridis and D. Roy, 'Grounded situation models for robots: Where words and percepts meet', in *IEEE IROS*, (2006).

[11] G. A. Miller, 'The magical number seven plus or minus two: some limits on our capacity for processing information.', *Psychol Rev*, **63**(2), 81–97, (March 1956).

[12] A. Newell, 'You can't play 20 questions with nature and win: Projective comments on the papers of this symposium', in *Visual information processing*, ed., W. G. Chase, 283–308, Academic Press, New York, (1973).

[13] A. Newell, *Unified theories of cognition*, Harvard University Press, Cambridge, MA, USA, 1990.

[14] S. Russel and P. Norvig, *Artificial Intelligence: a modern approach*, Prentice Hall Series in Artificial Intelligence, New Jersey, 1995.

[15] R. N. Shepard, 'Mental rotation of three-dimensional objects', *Science*, **171**(3972), 701–703, (1971).

[16] B. F. Skinner, ''Superstition' in the pigeon.', *Journal of Experimental Psychology*, **38**, 168–172, (September 1948).

[17] A. Sloman, 'Putting the pieces together again', in *The Cambridge Handbook of Computational Psychology*, ed., Ron Sun, 684–709, Cambridge University Press, Cambridge, (2008).

[18] N.E. Sokolov, H. Lyytinen, R. Naatanen, and J.A. Spinks, *The Orienting Response in Information Processing*, Lawrence Erlbaum Associates, New Jersey, 2002.

[19] W. Zhang and S. J. Luck, 'Do representations decay in visual working memory?', *Journal of Vision*, **4**(8), 396, (2004).

I think I know you: a sharable memory model between agent and human

Joana Campos and **Ana Paiva** [1]

Abstract. The idea of 'human-like' memory has gained importance with artificial companion systems, which attempt to "change interactions into relationships". Following that perspective, it is necessary to have memory models that not only store events, but also allow semantical integration of those events into the agent and user's lifetime. In our work, we argue that a sharable memory framework between agent and human, both in structure and content, is essential to help the agent to perform in a social environment and as such establish meaningful relationships with the user. In this paper we propose a memory framework for mapping also the user's memory into three levels of abstraction like humans do. Such structure should enhance social aspects of memory and development of social bonds between agents and human.

1 INTRODUCTION

Memory in humans is more than just a set of mechanisms for retaining mundane facts of our lives. It serves other purposes such as to define personal identity, to guide future behavior or to help people perform in a social environment. This functional approach of memory, most precisely autobiographical memory (AM), explains why people retain memories for so long [3]. Socially, those mechanisms contribute to develop intimacy and maintain relationships over time [14], as they allow us to naturally communicate with our peers.

Accordingly, 'human-like' memory architectures for *artificial companions* are recognized as an essential aspect for sustaining long-term relationships between those companions and humans [10]. Undeniably, agents endowed with such mechanisms would be able to perform in a social setting using the same base line for thought.

We are not the first to note that a *sharable* framework for representing memories or events would be an essential feature to integrate into the agents' memory [12]. In our work, we argue that such *sharable* framework not only in structure, but also in content, would support social aspects of memory and help the agent to behave properly in a verbal communication.

However, the memory architecture proposed in this paper does not try to fully reproduce the human memory. It tries to capture computationally some of its relevant aspects for achieving a more 'human-like' acceptable behaviour. We grounded our work on the assumption that conceptual autobiographical knowledge is formed from abstractions of episodic memories coupled with beliefs and attitudes of the working self [6]. In other words, AM can be seen as a 'semantic network' of events contextualized in one's life, retaining knowledge about progress of personal goals.

Therefore, we formalized a model for a companions' memory based on Conway's perspective [6], who suggests a human memory division in three levels of abstraction ([4] describes the model in detail). A more abstract level (*Lifetime Periods*) to contextualize the self in his lifetime, a middle level that accounts for the experienced events (*General Events*) and a less abstract level contextualizing the events in time and emotionally (*Memory Line*). This hierarchical conceptualization allows to the system interpret the encoded and retrieved information in a meaningful way, acting as specific views over the memory.

The underlying assumption is that such model could map not only the agent's memory, but also the companion would be able to share memory content with a human. Thus, throughout this paper, we describe the implemented memory structure and its processes of remembering and forgetting, always focusing in the social functionality of AM.

2 RELATED RESEARCH

Autobiographic memory (AM) in humans empowers the integration of the past into the future. Likewise, it has been suggested that in agents this could help them to communicate and form social relationships.

To address the question of how to include autobiographical memory mechanisms in an agent and how its own emotions can increase the believability through an interaction with a user, Ho et al. [9] defined AMIA (Autobiographical Memory for Intelligent Agents) framework - an autobiographical knowledge base of significant events sensed by the agent [9, 8]. Such framework does not try to copy an adult AM, but rather to capture essential features from some psychological models suggested by Conway [6].

Ho et al. proposed an implementable computational model (yet, not implemented), divided in Life Periods,Themes, Episodes, Events and Action, with different models that can be linked and yet evaluated separately. Events are organized by goals and they can encapsulate all necessary knowledge for a particular object or situation. Further, these highly specific experiences are a central feature of AM allowing the agents to represent their own experiences for acting in a virtual environment. The defined model, formalizes components that in fact are useful for addressing the agents' memory content. However, the model suggested does not contemplate a shared

[1] Instituto Superior Tecnico - UTL and INESC-ID, Portugal, email: joana.campos@ist.utl.pt and ana.paiva@inesc-id.pt

memory between user and agent.

Focusing on a more social memory, Mei Yii Lim et al. also present an initial prototype for a social companion generic memory. The aim is to create mechanisms reflecting human memory's characteristics and then allow companions to identify, characterize and distinguish experiences [11]. They differentiate two components of an agent's memory, Short Term Memory(STM) and Long Term Memory (LTM). The main goal is to maintain active the information relevant for the agent's actual state and at the same time ensure that the agent adapts to the situations over the long-term [7]

3 MEMORY'S ARCHITECTURE

In humans, Autobiographical memory (AM) has knowledge at three levels of specificity, which are sensitive to cues and patterns of activation. As mentioned earlier, while lifetime periods identify thematic and temporal knowledge, general events are related to actions, happenings and situations in one's life. Event Specific Knowledge (ESK) details are contextualized within a general event that in turn is associated with one or more lifetime periods, linking self autobiographical memory as a whole (fig. 1).

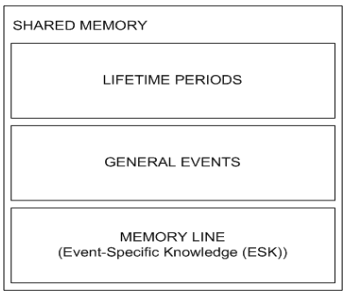

Figure 1. Knowledge base abstraction levels.

As referred by Conway [6] and other researchers[13], anything can be a cue. In our model, we will consider cues that could be represented by text due to our initial concern of developing a framework for *sharable* content between agent and human.

The underlying structure of the proposed architecture [4] is a collection of RDF (Resource Description Framework) triples consisting of $< subject, predicate, object >$. These triples are organized in graphs, one for each level of specificity, representing a simple data model for inference. The subject and object are nodes in the graph both linked by an edge. This edge represents the predicate, which establishes the relation between the two nodes.

3.1 Shared Memory

The implicit concept of shared memory (*sm*), should be clarified for a better understanding of the framework, as it is the basis for the memory's structure. Put simply, it is an experience that one had had and told to the agent. An event that occurred in the user's life. This information is defined by the tuple $< L, G, E >$, where:

- **L** – refers to one or more lifetime periods, which contextualize an event in a broad period of time.
- **G** – defines the main part of a shared memory, that is the action or event.
- **E** – specifies the details of one event.

Each one of the elements represents a level in the memory hierarchy. Besides all levels can be accessed separately, as if they were different views over the knowledge base, the 3 levels (graphs) are interconnected. This rich data integration allows us to represent knowledge in a meaningful way, while it unifies the memory as a whole.

3.1.1 Lifetime Period - L

A lifetime period (LTP) can be divided in two categories:

- **FL** – Fixed lifetime periods. Those refer to periods that are common in everyones' life. For example, the current year and the user's age.
- **SL** – When we refer to a specific time in our lives we use particular words that possibly only make sense for us. According to this idea, lifetime periods can be subjective, and cannot be defined *apriori*. Therefore, new LTPs should be created dynamically whenever it is needed.

3.1.2 General Event - G

A general event is a tuple with 6 characteristics
$< A, Wo, We, Wn, Wt, Ev, Sub >$, where:

- **A** (action) – infinitive of the main verb identified in the shared memory.
- **Wo** (who) – participants that had taken part in the event
- **We** (where) – specific place where the event occurred
- **Wn** (when) – specific time when the event occurred
- **Wt** (what) – any other complement of the shared memory that not fits in the other characteristics.
- **Ev** (event) – refers to the event itself. The event is generated by linking the action to one of the inferred characteristics: $A + \{Wo, We, Wn, Wt\}$. That link is based on the underlying semantic of the verb. For example, if the verb indicates movement, such as "go" or "go out", it links to where. Thus, the event is given by $A + We$.
- **Sub** (subevent) – link to a related G element.

3.1.3 Memory Line - E

The details of an event (G) refers to its surrounding context and the emotional details that the user may have added. This element is defined by the tuple $< T, D, Em, I, S >$, where:

- **T** (text) - sentence or set of sentences that describe the event and add personal details to it. It describes a personal view of the facts and its emotional connotation.
- **D**(date) - date object extracted from the *Wn* characteristic of the event. It corresponds to an instant (a specific day) or a interval with a settled begin and end.
- **Em** (emotion) - emotional state in T. The system is capable to work with this variable (and others that we might want to add), but at this stage of implementation the emotional stage is not inferred from T.

- **I** (image) - image that one can use to better describe the event.
- **S** (sound) - sound that could add some personal detail and sufficient to bring the event to one's mind.

3.2 Accessing agent's memory

Remembering is a complex process and is theoretically divided in three phases *Encoding*, *Storage*, *Retrieval*. We based our approach to accessing the knowledge base in this three stages, which are formalized below.

3.2.1 Encoding Process

The encoding process refers to the stage that information is registered [1] and is directly linked to how interesting some subject is. In this process, sequences of linked events are associated with different kinds of information, as formalized in the previous section. Each event is formalized according to the structure of a *sm*, formed by layer's rules.

Figure 2 depicts the encoding process for a written text as input. A main event (G1) is extracted from the set of sentences and sub-event (G2) is attached to it.

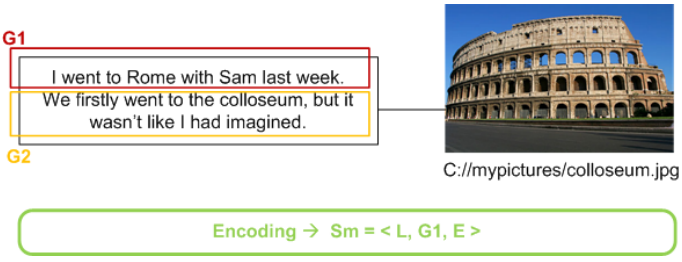

Figure 2. Encoding example.

To each one of the general events, the components of a shared memory (*sm*) are extracted and after used as cues for memory triggering.

```
L = Year 2010; Relationship with Sam

G1 = <A   = go;          G2 = <A   = go;
Wo  = I, Sam;            Wo  = I, Sam;
We  = Rome;              We  = the colosseum;
Wn  = last week;         Wn  = - ;
Wt  = - ;                Wt  = it wasn't like I
Ev  = go to Rome;        had imagined;
Sub = G2 ; >             Ev  = go to the colosseum;
                         Sub = - >

E = < T  = I went to Rome with Sam last week.
 We went to the colosseum,
 it wasn't like I had imagined.;
      D  = 2010-03-01 , 2010-03-07
      Em = - ;    S  = - ;
      I  = c://mypictures/colloseum.jpg
```

3.2.2 Storage

The *storage* or *consolidation* is the process whereby information is maintained in memory over time [1]. In human memory only relevant and important events are retained in one's memory. Thus, as we are concerned with an agent that gathers meaningful information about the user, a filter should be applied to guarantee that only relevant events are stored in memory.

A straightforward approach for filtering relevant information, can be done weighting the action present in some event. For example, actions that may change the user's state are more important than others that do not make such change. Therefore, a *sm* is only stored in the agent's memory if the event (based on its action) adds relevant information about the user.

Continuing with the previous example, the verb 'to go' indicates movement and perhaps an important change on the user's state, so that may be a good indicator that the event associated should be retained in memory. According to that decision a shared memory object is created (see fig. 3).

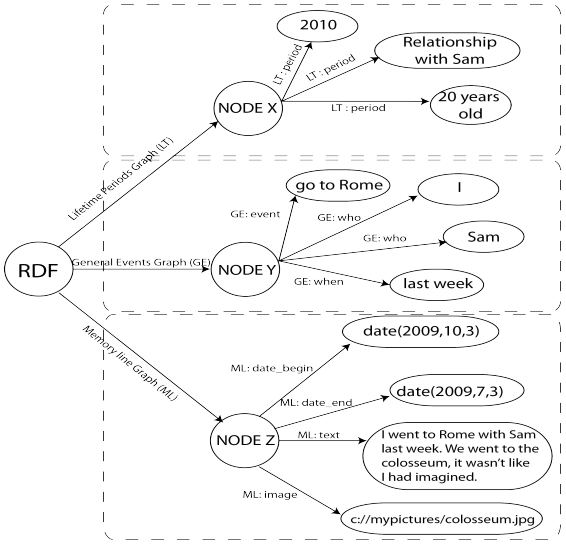

Figure 3. Graphical example of the RDF memory description.

3.2.3 Retrieval

The retrieval is a complex process for accessing information, which is possible by recognition or recall [1]. Recall is normally an intentional process but can occur when some perception - cue - triggers some experienced event. In our approach we take into account such perspective to simplify the access to the agent's memory and to take advantage of the social function of AM. Researchers have suggested that everything can be a cue for memory triggering, as long as it is linked to an event.

Therefore, any element at any level of this memory model can be retrieved, as long as, a reference to the general event is present. So to retrieve any element from memory three elements should be present $O =< G, Lv, Wr >$, where:

- **G** – reference to the specific event (SUBJECT)
- **Lv** – level in the database or graph where the search should be performed (context)
- **Wr** – what to retrieve from the performed search. Note that this element represents an edge in the graph, more precisely a relation between two nodes. (PREDICATE)
- **O** – obtained element (OBJECT)

To get any event based on information from any of the other levels, it is only necessary to perform the inverse process $G =< Wr, Lv, O >$.

4 SOCIAL MEMORY

An important function of memory in a social environment is to increase one's responsiveness, as it allows listeners to make empathetic and contextually grounded responses to what the speaker is saying [2]. To enhance this social characteristic, the described structure plays an essential role providing a very cue sensitive database, which facilitates building the social bond.

This memory model was integrated into the architecture of a companion system (MAY - my Memories are Yours) [4], which interacts with the user through dialogue. We verified that the memory content in conjunction with its structure, offers support to a more interesting interaction. Not only does this sharable framework increase the agent's responsiveness, but also its subsequent utterances are sensitive to textual cues in user's input.

4.1 Companion's overview

MAY is an agent created to assist a teenager user on self-reflection about what happens in his/her life. The communication between the agent and the user is done through dialogue by which the shared memories are collected and saved in a diary form (or timeline). The process flows as follows:

- Every sentence will be analyzed using the natural language tools, which are responsible for (1) identifying a sentence's verbal tense and to separate future from past events; (2) identifying the event (action) and its characteristics: *when* it happened, *who* participated and *where* it took place. Those components are responsible for indexing an event.

- The *shared* memory base is updated with every new relevant "sensed" event. Apart from this main module, which stores all relevant past events in user's life that the agent knows. Another one, similar in structure, accounts for events that have not happened yet.

- To produce an adequate response, the agent starts by searching its memory for anything appropriate to say. It looks for active goals, past events with some relevant information for the current situation or even go beyond the present and infer future plans. Other tools are also available for enriching the dialogue (For instance, ConceptNet[2]).

[2] http://csc.media.mit.edu/conceptnet

4.2 *Shared* structure and content

When recollecting some event from memory, humans follow a pattern that starts by establishing a broad period in their lives, which cannot always be mapped into a date ("When I was 15 I went to Brazil"). Then, they specify an event and after that they start describing its details [5]. The described memory structure tries to map this line of thought, enabling the agent to 'think' as similar as possible to the user. Allowing him/her to 'frame' the experienced events in a semantically meaningful space.

In the previously described scenario, the memory has an essential role on assisting the dialogue and again on increasing the agent's responsiveness using the previous *shared* events. This task is performed by 'sensing' if any internal stimulus lead to a pattern identified in data, either after a memory had been created or at any point of the interaction. So far, we focused on three views of data 1. *Tracking Goals*; 2. *Virtual Sensing*; and 3. *Forecast*. Also, the user is able to ask direct questions to the agent.

The former function decides, with user acquiescence, whether to store permanently or simply eliminate some event in the active goals database. The other two functions, which we call *Virtual Sensing* and *Forecast*, extract specific generalities while focusing on specific information. These cognitive functions are supported on the fact that we nearly always interpret new events based on available knowledge about the world and about our selves. In this case, the agent interpret the past or new events based on the shared events on previous interactions.

- *Virtual Sensing* – is concerned with the agents ability to sense that something is missing in the told event. When this search in the database is performed, we compute a view of the events that match the criterion of the event introduced. We set that the agent would be confident in believing that some fact is true in proportion with the number of events in memory.

- *Forecast* – normally refers to future, but in this case it refers to anything that the agent does not actually know but still can be inferred using the data in memory. The aim of this feature is particularly useful in diversifying the conversation when the system asks about yesterday or tomorrow. In contrast with the previous feature, this takes in account the day of the week to make a prediction.

4.3 Memory access through layers

The aforementioned cognitive functions use differently the layered structure of memory. In this section we describe situations wherein the companion uses the information in memory to enrich the dialogue.

4.3.1 Virtual Sensing

In this situation the agent makes a prediction about some missing element in a sentence. To clarify this situation, consider the following sentence: "I'm going to the cinema with Lyam." . In this sentence typed by the user the element 'where' is missing, so the agent combines two things that it knows: most of the times the event 'go to the cinema'(event)

with 'Lyam'(who) happens every other 'Thursday' (when). This query occurs in two different levels of the data base: General Events, which match 'event' and 'who' and also the Memory Line, which gathers more specific details of each event (day of the week).

Figure 4 (left) depicts preliminary results of query time[3] for getting the required information. The time for query response increases in pair with the database size. At this point, the superior level would have an important and relevant role in decreasing the search space and to guarantee agent better performance. Figure 4 (right) shows the size of memory in terms of number of nodes.

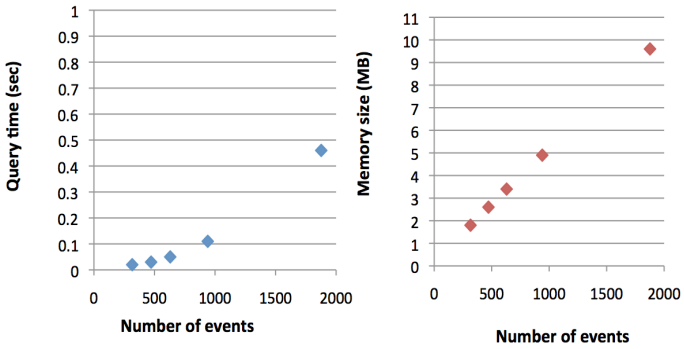

Figure 4. Storage space in function of the number of nodes

4.3.2 Forecast

When the agent tries to diversify the conversation asking about yesterday or tomorrow, a sentence can take the following structure: "Are you going jogging to the Stadium, tomorrow?" . The agent knows that several times in the past the user told it that at Saturdays he/she used to going jogging to the stadium.

5 CONCLUSION

In this paper, we described a framework which allows the representation of the agent's mental information about the user similarly to how humans do. Furthermore, this implemented model enables to share content and structure allowing the agent to behave more naturally in a social setting. In a few steps the agent is capable of recalling the exact episode in different granularities of time at any level of the RDF structure.

We consider that this structure provides an acceptable time for reaction in dialogue (< 0.5 *seconds* for large databases). However, the system would benefit if the agent encompass a lifetime period that could map the 'current period' in one's life. It would work as a 'window' over the events in memory establishing the search space constant independently of the memory size. Therefore, the memory (autobiographical memory) would be considered, as it should be, a "transitory representation " [5].

[3] The time for each query is the average of 10000 loops over the same function. We followed this procedure due to sightly variations on the Python processor and background processes during a normal interaction.

As mentioned in the opening section the social feature is only one of the characteristics of memory and this framework offers a base line for exploring the remaining functions enabling the agent to operate in situations and relations that are not present to the senses [15].

ACKNOWLEDGEMENTS. This work is partially supported by the European Community (EC) and is currently funded by the EU FP7 ICT-215554 project LIREC (LIving with Robots and IntEractive Companions), and FCT (INESC-ID multiannual funding) through the PIDDAC Program funds. The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

REFERENCES

[1] A. D. Baddeley, 'The psychology of memory', *in The Handbook of Memory Disorders*, (Apr 2002).
[2] S. Bluck, 'Autobiographical memory: Exploring its funtions in everydaylife', *Memory*, **11:2**, 113–123, (2003).
[3] S. Bluck, N. Alea, T. Habermas, and D. C. Rubin, 'A tale of three functions: The self-reported uses of autobiographical memory', *SOCIAL COGNITION*, **23(1)**, 91–117, (2005).
[4] J. Campos and A. Paiva, 'May: My memories are yours', in *Proceedings of the 10th International Conference on Intelligent Virtual Agents*, volume 6356, pp. 406–412. Springer Berlin / Heidelberg, (2010).
[5] M. A. Conway, 'Sensory-perceptual episodic memory and its context: autobiographical memory', *Philosophical Transactions of the Royal Society B: Biological Sciences*, **356**, 1375–1384, (2001).
[6] M. A. Conway, 'Memory and the self', *Journal of Memory an Language*, **53**, 594–628, (2005).
[7] W. C. Ho, K. Dautenhahn, M. Y. Lim, P.A. Vargas, R. Aylett, and S. Enz, 'An initial memory model for virtual and robot companions supporting migration and long-term interaction', in *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pp. 277 –284, (272009-oct.2 2009).
[8] W. C. Ho and S. Watson, 'Autobiographic knowledge for believable virtual characters', in *Intelligent Virtual Agents*, eds., Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier, volume 4133, 383–394, Springer Berlin / Heidelberg, (2006).
[9] W. C. Ho, S. Watson, and K. Dautenhahn, 'Amia: A knowledge representation model for computational autobiographic agents', in *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pp. 247 –252, (2007).
[10] J. E. Laird and N. Derbinsky, 'A year of episodic memory', *In: Workshop on Grand Challenges for Reasoning from Experiences, 21st IJCAI (2009)*, (2009).
[11] M. Y. Lim, R. Aylett, W. C. Ho, S. Enz, and P. Vargas, 'A socially-aware memory for companion agents', in *IVA '09: Proceedings of the 9th International Conference on Intelligent Virtual Agents*, pp. 20–26, Berlin, Heidelberg, (2009). Springer-Verlag.
[12] N. Mavridis and M. Petychakis, 'Human-like memory systems for interactive robots: Desiderata and two case studies utilizing grounded situation models and online social networking', in *AISB - Symposium on Human Memory for Artificial Agents.*, (2010).
[13] A. R. Mayes and N. Roberts, 'Theories of episodic memory', *Philosophical Transactions of the Royal Society B: Biological Sciences*, **356**, 1395–1408, (2001).
[14] K. Nelson, 'The psychological and social origins of autobiographical memory', *Psychological Science*, **4(1)**, 7–14, (1993).
[15] E. Tulving, 'What is episodic memory?', *Current Directions in Psychological Science*, **2**, 67–70, (1993).

Between Downward Spirals and Habituation: Emotion Intensity in Virtual Agents' Memory Retrieval

Paulo F. Gomes and **Ana Paiva** and **Carlos Martinho**[1]

Abstract. In the present article a model for memory retrieval of personal experiences for virtual agents is presented. It builds upon previous work and focuses on the effect memory retrieval can have on the agent's emotional state. Memory retrieval is defined as an emotional re-appraisal of past experiences. The variation of intensity of such re-experience is explored by modeling two phenomena: *downward spirals* and *habituation*. Downward spirals consist of situations in which recurrent retrieval of negatively charged past memories increases the recall intensity of such memories in depressed individuals. Habituation is the reduction of recall intensity caused by re-experiencing past experiences in a context perceived by the individual as safe.

1 INTRODUCTION

Recollection of personal experiences sometimes shoots oneself through a trip of sensations. These experiences can be moments of professional achievement, amorous conquest, peer conflict, a friend departing, etc. We travel to a past time, reliving those episodes [18], with previously elicited emotions coming to mind [12]. Daniel Schacter describes how troublesome this voyage can sometimes be:

"*We've all endured difficult experiences - the death of a loved one, rejection by a lover, failure at work - that pain us mightily in the days and weeks after they occur. In the immediate aftermath, we may find ourselves reliving the painful incident to the point of distraction,* ..."[17]

Should we not expect believable agents to have some degree of this "emotional memory retrieval"? How coherent would seem the behavior of an agent, that when returning for the first time to the place where it was first kissed, displays no emotional reaction? To address this issue, in previous work we defined and implemented a model in which an agent emotionally re-appraises past events when retrieving them [6][7].

However, the evaluation of the model seemed to indicate that because an agent tended to always react emotionally in the same way when retrieving a specific past memory, in the long-term this might render its behavior as excessively repetitive and predictable [6]. In the current document we propose a mechanism to add some variability to the model, by changing the experience's intensity between different retrieval situations. Furthermore, this mechanism will be inspired in two phenomena of human memory. Such a model can ultimately contribute to the believability of virtual agents by supporting behavior coherence while maintaining variability in the behavior [15].

[1] INESC-ID and Instituto Superior Técnico, Portugal, emails: pgomes@gaips.inesc-id.pt and ana.paiva@inesc-id.pt and carlos.martinho@ist.utl.pt

2 RELATED WORK

Many researchers have already defined agent architectures in which both past memories and emotions are present. The cognitive architecture Soar [11] is an example, yet in it there is no integration between the two. In [8] a virtual agent architecture framework is proposed in which past events are used to select strategies to deal with similar current events, and the emotional state is an indicator of the discrepancy between a desired state and the perceived state. However, emotion is not part of the memory retrieval process.

In FAtiMA [4] agents emotionally appraise events, store a personal story of these events, and can textually reconstruct it. Also oriented towards enabling reconstruction of personal stories, the agent memory system described in [3] takes into account memories' emotional charge. Memories concerning events that were initially perceived as more emotionally relevant take longer to forget. In opposition, in the cognitive architecture presented in [5], memory episodes linked with emotions of higher intensity have a higher probability of being retrieved. Nevertheless, in none of these architectures are past events re-appraised emotionally.

On the other hand, in the SALT & PEPPER agent architecture [2] emotion is an integral part of the retrieval process. The authors define emotions as performance evaluators and attention shift warnings. When an emotion is generated it is matched against the header of all nodes in a memory network. Nodes have different activation levels and only the matching node with highest activation level is retrieved to working memory. This retrieval causes the node's activation level to increase, activation which in turn is spread to neighboring nodes, similarly to activation spread in ACT-R's declarative memory [1]. Although it would be possible to use SALT & PEPPER to support change in the emotional state caused by retrieval of personal experiences, few clues are given on exactly how to discriminate between past and present in their appraisal.

In [10] the authors describe an agent architecture for an interactive virtual character (Eva) for which past interactions indirectly influence the agent's emotional state. The relationship between the agent and a specific user, defined in a two-dimensional space of dominance and friendliness, depends on emotions felt by the agent in previous interactions. For instance, gratitude will increase perceived user friendliness and decrease perceived dominance towards the user. Moreover, the relationship values change the agent's mood when interacting with the user, and in turn the mood affects the intensity of emotions. Although, memories indirectly influence the emotional state, the system is unable to model retrieval of past events not entailing a relationship.

All in all, supporting the inter-connection between emotion and memory has yet to be fully investigated, particularly in what concerns

the retrieval process.

3 MEMORY RETRIEVAL MODEL

In previous work [7] a model for memory retrieval of personal experiences was defined, and implemented, drawing inspiration from Tulving's conceptualization of the process [19]. The model is divided in two main stages: *location ecphory* and *recollective experience*. In location ecphory, memories connected with the agent's current location are selected (details in [6]). The recollective experience consists of re-appraising the selected memories' associated past events according to current motives[2]. This model of memory retrieval was integrated in an agent architecture schematically represented in Figure 1.

As can be noticed, the recollective experience is essentially an appraisal process. This process follows closely FAtiMA's reactive appraisal [4] and the emotional related concepts are inspired in the OCC theory of emotions [15]. For instance, an *emotion* is defined as a valenced evaluation of an event and contains the following elements:

- *type* of the emotion according to the OCC model [15] (e.g. pity).
- *intensity* specifies the emotion's current intensity (non-negative scalar value). It decays to zero with time. When the intensity reaches a close to zero value the emotion is removed from the emotional state.
- *valence* specifies the emotion's value (positive or negative). The valence is directly dependent on the emotion type. For instance, joy emotions are positively valenced and pity emotions are negatively valenced.

Moreover, an *emotional state* is defined by the following elements:

- *active emotions* contains the set of emotions the agent is currently feeling. The *Behavior* module of the agent architecture (Figure 1) is responsible for making the agent display a facial expression corresponding to the emotion with highest intensity.
- *mood* is a bounded scalar value that represents the agent's recent overall emotional state valence. The evocation of negatively valenced emotions decreases the mood, while the evocation of positively valenced emotions increases it. These changes are proportional to the intensity of the evoked emotions. Additionally, mood decays to a neutral value with time.

The emotional state is affected by the appraisal of past events, during the recollective experience stage of memory retrieval, as well as present perceived events (regular appraisal). Events have a frame-like representation in which an event can have a sub-event. For example, a past event of witnessing another agent falling in a hole placed at bi-dimensional coordinates (200,300) could have the following representation: [Event type: retrieval sub-event:[Event type: witness sub-event:[Event type: fallHole location:(200,300)]]]. A past event is characterized by having type *retrieval*, and its sub-event parameter value is the event associated with the retrieved memory selected in the ecphory stage. Past events are stored in memory as memory traces. A memory trace is defined by the following elements:

- *event* is a representation of the event which the memory is about.
- *emotion* specifies the emotion caused by the appraisal of the event. If an event's appraisal generates more than one emotion, one memory trace is created for each one. In the remainder of this paper, memory traces with a negative emotion will be referred to as *negatively valenced*, and memory traces with a positive emotion will be referred to as *positively valenced*.
- *time stamp* is a meta-field indicating when the event started or when the memory trace was retrieved for the last time.

In the current system only if an event's appraisal causes a change in the emotional state is the event stored. This approach tries to simulate the effect emotion content has on promoting attention focus during encoding [16] and on enhancing elaborative rehearsal. Instead of enforcing a forgetting mechanism such the one in [3], the events are filtered during storage. However, we have yet to empirically compare these two methods.

When a memory trace is selected by ecphory, a past event is created and fed to the appraisal system for recollective experience. This past event has type retrieval and has as its sub-event the event parameter of the memory trace. In the model, a past event appraisal consists of appraising its sub-event. However, appraisal may differ from the original one regarding the following perspectives:

- The mood influences the intensity of generated emotions: a positive mood enhances positively valenced emotions' intensity, and reduces the intensity of negatively valenced emotions. As mood may change, so can the intensity of generated emotions. Furthermore, if the re-calculated intensity is small enough, the emotion will not even change the emotional state. The mood influences appraisal by reflecting the recent past emotional experience.
- The agent's motives may change, and hence the appraisal and consequent generated emotions may differ from the originally created.

Finally, guided by the assumption that re-living a past experience will typically be less intense as the original experience [12], the final intensity of a generated emotion due to the appraisal of a past event is reduced by a parameterizable positive factor smaller than one (*memory retrieval intensity bias*):

$$intensity = intensity \times memory\ retrieval\ intensity\ bias$$

4 DOWNWARD SPIRALS AND HABITUATION

It is debatable to assume that re-living a past experience will always be a less intense experience than actually perceiving the experience first-hand, as the just shown expression implies. The two presented factors of emotion intensity variability have limitations: due to time decay, the mood only encodes the agent's recent past emotional context, and not an overall process of dealing with the past experience; changing motives arbitrarily can harm an agent's believability by preventing an individual evaluating the agent's behavior from creating a mental model of these motives [14]. Therefore, we propose adding to the existing model a mechanism for tuning emotion intensity variability during memory retrieval that takes into account the agent's overall experience with a specific memory.

Daniel Schacter discussed how retrieval of a certain memory can influence future recall of that same memory [17]. He states that continual reminding of a personal memory can sometimes strengthen recall. Moreover, that depressed patients better encode negative experiences and have a greater tendency for a phenomenon he names as memory *persistence*. Persistence of memories consists on an individual recurrently retrieving a memory when he, or she, does not wish to do so. An accountant not being able to concentrate on doing a company's tax declaration due to the constant retrieval of a car accident in which he was hurt, is an example of memory persistence.

[2] The term motive is used as an abstraction over goals or desires.

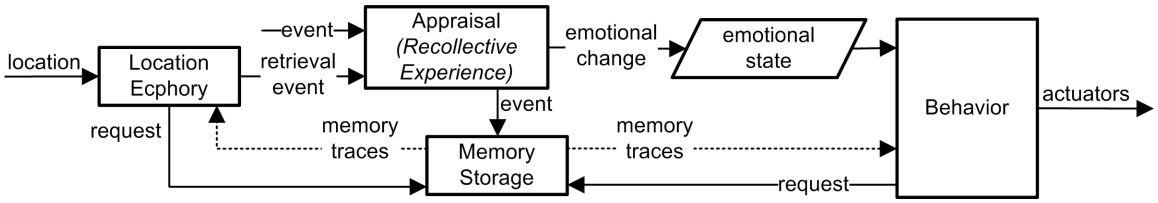

Figure 1. Agent Architecture

Depressed individuals are more prone to enter an emotional *downward spiral* connected with this phenomenon: they have a greater tendency to retrieve negative experiences, which can negatively influence their mood, consequently increasing their susceptibility to encode more negatives events (that can the experience of retrieval itself), that in turn will have a greater tendency to be retrieved. Schacter proposes that this loop may internally be enforced by an integration of negative past experiences into the self-schema (the individual's conception of itself) in which the former is linked with generic negative concepts such as "I do not do anything right" or "bad things always happen to me".

Despite their possibly devastating effects, Schacter argues that downward spirals caused by memory persistence may not be unavoidable. Telling others troubling experiences has been shown to have positive effects. Furthermore, re-experiencing a past traumatic experience in a safe context may result in a better integration with the self-schema and a less intense recollection further on. Abstractly, we can interpret this process as an *habituation*: reduced physiological response to repeated stimuli.

5 DYNAMIC EMOTION INTENSITY IN MEMORY RETRIEVAL

We propose modifying the memory retrieval model presented in Section 3 to take into account the phenomena of downward spiral and habituation. Each memory trace would have its own retrieval intensity bias value, that we rename as *persistence*, and this value could change when the memory trace was retrieved: if habituation took place it would decrease; if the downward spiral phenomenon took place it would increase; if neither phenomena were active the value would remain the same. Consequently, we need to define the conditions in which habituation and downward spirals take place.

Schacter presents depression as being a factor for an increased downward spiral effect [17]. Although depressed individuals are more prone to downward spirals, that does not rule out that the phenomenon, at least to some degree, can take place in non-depressed individuals. In a first approach, we propose mapping a state leaning towards potential depression in humans, to the agent having an extremely low mood value. Therefore, we propose that for the downward spiral phenomenon to take place, the mood must be lower than a *downward spiral threshold*. For instance, we could consider cases in which the mood is in the lower quarter of the mood range. Note that *downward spiral threshold* should be lower than the defined neutral value (see Section 3).

Turning to habituation, Schacter indicates the feeling of safety as promoting this phenomenon [17]. We propose mapping the feeling of safety to the absence in the emotional state of an emotion of type "fear" according to the OCC model [15]. Hence, for habituation to take place, there can not be an emotion of type "fear" in the emotional state.

Additionally, as habituation and downward spirals have opposite effects on subsequent recollective experiences of a memory trace, decreasing the emotion intensity and increasing it, we propose that they should be mutually exclusive. Therefore, we define that downward spirals will only take place if there is an emotion of type "fear" in the emotional state, and that habituation will only take place if the mood is higher than the downward spiral threshold. Summarizing the phenomena conditions:

- **downward spiral**: mood is lower than downward spiral threshold, and an emotion of type "fear" is present in the emotional state.
- **habituation**: mood is higher than downward spiral threshold, and no emotion of type "fear" is present in the emotional state.

Finally, considering the change in persistence (formerly named as memory retrieval bias), we propose that when a memory trace is retrieved in the downward spiral's conditions, the persistence is increased by the parameter value *persistence increase*. However, if with this increase the persistence would be greater than a parameter *max persistence*, then it would be reset to max persistence[3]. Analogously, when the habituation conditions are verified, the persistence could decrease to a minimum of *min persistence*. As the emotion intensity is a non-negative scalar, and the recalculated intensity is $intensity \times persistence$, min persistence must not be negative.

6 APPLICATION

The variation of the memory retrieval's emotion intensity will only positively affect believability if it is reflected on the agent's Behavior. The model presented in Section 3 has been integrated into a game prototype ("Meemos' Rescue") in which that happens [6]. In "Meemos' Rescue" the player controls a character (meemo captain) and through it can issue commands to several non-player characters (meemo minions). The objective is to lead the meemo minions to an exit point. The meemo minion's expressive behavior greatly depends on the architecture of Figure 1, and we will refer to them as agents for the remainder of this section.

An agent's mood is graphically represented by its color saturation. The lower the saturation of a color, the closer it will be to a gray tone. The word "gray" can be used to classify a mood [9] describing it as negative. In fact, gray tones are sometimes associated with a negative state of mind. With this motivation, when the mood is below its neutral value, the lower it is, the lower its color saturation is. Considering that the saturation percentage varies between 0 (minimum saturation)

[3] The value 1 is a potentially interesting candidate for the max persistence: on one hand the retrieval of past experiences will typically be less intense than the actual experience; on the other, if the mood is much lower than in the original situation, the resulting emotion can in fact be more intense than the original one.

and 1 (normal saturation), and that negative moods belong to the interval $[minimum\ mood; neutral\ mood]$ the saturation percentage is calculated by the following expression:

$$saturation\ percentage = \frac{mood - minimum\ mood}{neutral\ mood - minimum\ mood}$$

Mood is influenced by the evocation of emotions, as mentioned in Section 3. Consequently, the retrieval of negatively valenced memory traces indirectly decreases the agent's color saturation. Moreover, the decrease in color saturation will be proportional to the emotion intensity of the memory retrieval.

Still considering emotion expression, agents express the most intense emotion through a facial expression (see Figure 2). The current implementation supports four expressions: *neutral* (that acts as a baseline), *sadness*, *happiness* and *anger*. The choice of the three last expressions was motivated by the fact that they are part of the group of six universally recognized facial expressions (anger, disgust, fear, happiness, sadness and surprise) [13].

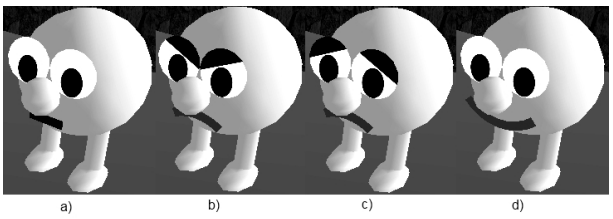

Figure 2. Meemo's face expressions: a) neutral b) anger c) sadness (Distress/Pity) d) happiness (Joy/HappyFor)

We mapped emotion types to the available emotion expressions. Note that each emotion type from OCC [15] represents a family of emotions. The emotion type "joy", for instance, represents emotional states such as happy, glad, delighted, pleased, etc. On the other hand, "distress" represents emotional states such as sad, unhappy, feeling bad, displeased, dissatisfied, etc. With this in mind, we mapped the emotion type "joy" to the *happiness* expression and the emotion type "distress" to the *sadness* expression. If there are no emotions in the active emotions, the *neutral* expression is displayed (Figure 2a). Unfortunately, emotions of type "angry" are not supported by our current model implementation, hence *angry* expression is never selected.

Turning to other supported emotion types, "happy-for" represents emotional states happy-for, delighted-for, pleased-for, etc. One can notice that these emotional states are quite similar to the ones presented for "joy". Thus, "happy-for" was also mapped to the *happiness* expression. "Pity" represents emotional states compassion, sympathy, sad-for, sorry-for, etc. It has been claimed that emotions such as compassion and sympathy have a different facial display pattern than distress [13]. However, this pattern seems to include oblique eyebrows, which are part of the *sadness* expression. Consequently, "pity" was also mapped to *sadness* expression. Lastly, the remaining emotion types supported by the implementation do not have a mapped facial expression.

In addition, if the agent's facial expression changes due to the retrieval of a past event, a thought balloon is presented (see Figure 3) for short period of time (parameterizable in a configuration file). An image is displayed on the thought balloon representing the remembered event. In Figure 3 the retrieved event was witnessing another agent falling into a trapdoor.

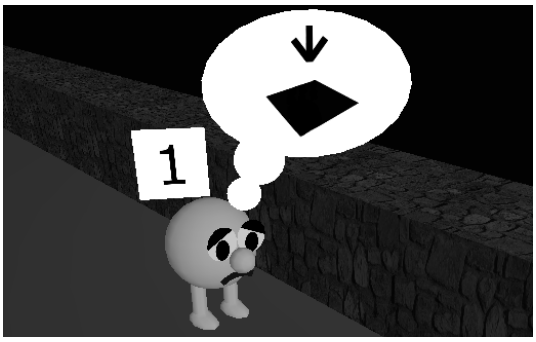

Figure 3. Meemo's thought balloon

Finally, the agent's behavior also directly considers the memories. The path planning tends to avoid locations where negative events have occurred and favoring paths where positive events have occurred [6].

7 EVALUATION

We believe that such an application can be used in a user study to evaluate how variating the emotion intensity of the recollective experience, as described in Section 5, can improve perceived believability. We propose a scenario in which a character negatively appraises an event taking place at a certain location, and returns to that same location several times, always remembering the past event (recall 1, 2 and 3). We can separate the evaluation in two sub-scenarios, one to evaluate the habituation effect (storyboard in Figure 4), another to evaluate the downward spiral effect (storyboards in Figure 5).

In the former, there will be two test conditions: *with habituation* (with H) and *without habituation* (without H). In test condition *with habituation*, when returning to the mentioned location, and remembering the past event (symbolized by the appearance of a though balloon with an iconic representation of the event), the character will progressively express a less intense expression. In test condition *without habituation* the reaction to the memory retrieval will always be the same.

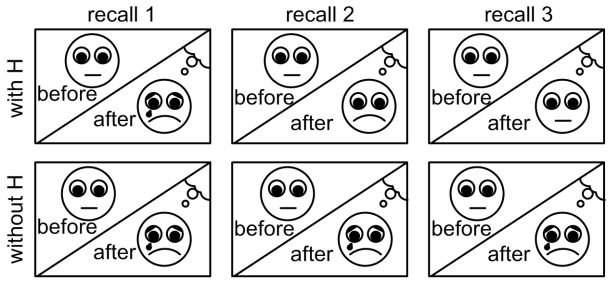

Figure 4. Habituation evaluation storyboard

When considering the effect of downward spirals, there will also be two test conditions: *with downward spiral* (with DS) and *without downward spiral* (without DS). In both test conditions, when returning to the mentioned location for the second time, the character displays a fearful expression. In test condition *with downward spiral*

the emotional reaction of the character will progressively be more intense. In test condition *without downward spiral*, the reaction will always have the same intensity.

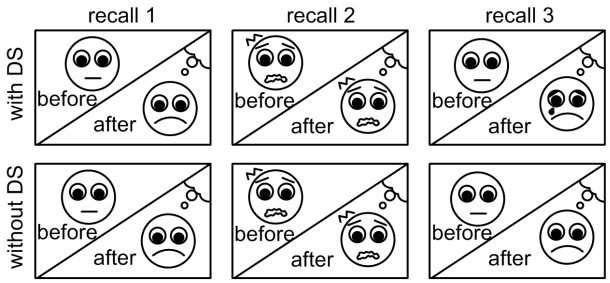

Figure 5. Downward spiral evaluation storyboard

As proposed in [14], believable character's behavior should be perceived as coherent, but at the same time not be perceived as excessively predictable. We propose that test participants, after being exposed to a test condition, would have to rate perceived coherence and predictability similarly to how it was performed in [6][7]. We expect that: test condition *with habituation* will present lower scores for predictability, and similar scores for coherence, when compared to *without habituation*; test condition *with downward spiral* will present lower scores for predictability, and similar scores for coherence, when compared to *without downward spiral*.

8 CONCLUDING REMARKS

In the present article a model for memory retrieval of personal experiences was presented. It builds upon previous work and focuses on the effect memory retrieval can have on the emotional state. Memory retrieval is defined as an emotional re-appraisal of past experiences. The variation of intensity of such experience is explored by modeling the phenomena of downward spirals and habituation. The modifications needed to take into account these phenomena have yet to be implemented and evaluated.

Although the modifications may account for a possibly greater variation of emotion intensity between retrievals, they unfortunately introduce four new parameters to the model: persistence increase, max persistence, persistence decrease and min persistence. Furthermore, it can be argued that the persistence increase should depend on the intensity of the active fear emotion or on the mood value. A potential approach for setting these parameters, as well as the initial persistence value, would be to map the differences in retrieval results between normal individuals and depressed individuals, to the differences in persistence between agents in conditions of the downward spiral phenomenon and agents not in these conditions.

If the "Meemos' Rescue" application is to be used to create the evaluation scenario, support for expressive behavior when the agent is feeling "fear" needs to be added (e.g. having the character's body tremble). It would also be an advantage to have different facial expressions for high and low intensity emotions.

Finally, as the model becomes increasingly more complex, the need for mechanisms such as forgetting, activation spreading, temporal organization between memory traces, and network organization of memory traces, also increases. Future work will probably entail integrating it into another memory system that already supports some of these mechanisms.

ACKNOWLEDGEMENTS

This work is partially supported by the European Community (EC) and is currently funded by the EU FP7 ICT-215554 project LIREC (LIving with Robots and IntEractive Companions), and FCT (INESC-ID multiannual funding) through the PIDDAC Program funds. The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein. I would like to thank the anonymous reviewers for all the comments and suggestions.

REFERENCES

[1] John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin, 'An integrated theory of the mind', *Psychological Review*, **111**(4), 1036–1060, (2004).

[2] Luís M. Botelho and Helder Coelho, 'Machinery for artificial emotions', *Cybernetics and Systems*, **32**(5), 465–506, (2001).

[3] Cyril Brom, Klára Pešková, and Jiří Lukavsky, 'What does your actor remember? towards characters with a full episodic memory', in *ICVS'07: Proceedings of the 4th international conference on Virtual storytelling*, pp. 89–101, Berlin, Heidelberg, (2007). Springer–Verlag.

[4] João Dias, Wan C. Ho, Thurid Vogt, Nathalie Beeckman, Ana Paiva, and Elisabeth André, 'I know what i did last summer: Autobiographic memory in synthetic characters', in *Affective Computing and Intelligent Interaction*, pp. 606–617, (2007).

[5] Will Dodd and Ridelto Gutierrez, 'The role of episodic memory and emotion in a cognitive robot', in *IEEE International Workshop on Robot and Human Interactive Communication*, pp. 692–697, (2005).

[6] Paulo F. Gomes, *MEEMOs: Believable Agents with Episodic Memory Retrieval*, Master's thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, 2010.

[7] Paulo F. Gomes, Carlos Martinho, and Ana Paiva, 'I've been here before! location and appraisal in memory', in *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, (2011). To appear.

[8] Wan C. Ho and Scott Watson, *Intelligent Virtual Agents*, chapter Autobiographic Knowledge for Believable Virtual Characters, 383–394, Springer Berlin / Heidelberg, 2006.

[9] *The American Heritage Dictionary of the English Language*, Houghton Mifflin Company, fourth edn., 2009.

[10] Zerrin Kasap, Maher B. Moussa, Parag Chaudhuri, and Nadia Magnenat-Thalmann, 'Making them remember - emotional virtual characters with memory', *IEEE Computer Graphics and Applications*, **29**(2), 20–29, (2009).

[11] John E. Laird, 'Extending the soar cognitive architecture', in *Artificial General Intelligence*, (2008).

[12] Andrew R. Mayes and Neil Roberts, 'Theories of episodic memory', *Philosophical Transactions of the Royal Society B: Biological Sciences*, **356**(1413), 1395–1408, (2001).

[13] Keith Oatley, Dacher Keltner, and Jennifer M. Jenkins, *Understanding Emotions*, chapter Communication of Emotions, 83–114, Blackwell Publishing, 2006.

[14] Andrew Ortony, *Emotions in Humans and Artifacts*, chapter On making believable emotional agents believable, MIT Press, 2003.

[15] Andrew Ortony, Gerald L. Clore, and Allan Collins, *The Cognitive Structure of Emotions*, Published by Cambridge University Press, 1990.

[16] Elizabeth Phelps and Tali Sharot, 'How (and why) emotion enhances the subjective sense of recollection', *Current Directions in Psychological Science*, (2008).

[17] Daniel L. Schacter, *How The Mind Forgets and Remembers*, Souvenir Press, 2001.

[18] Endel Tulving, 'Episodic memory: From mind to brain', *Annual Review of Psychology*, **53**, 1–25, (2002).

[19] Endel Tulving, Martin E. Le Voi, David A. Routh, and Elizabeth Loftus, 'Ecphoric processes in episodic memory [and discussion]', *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences (1934-1990)*, **302**(1110), 361–371, (1983).

Towards modeling false memory using virtual characters: a position paper

Michal Čermák and **Rudolf Kadlec** and **Cyril Brom**[1]

Abstract. This position paper presents our approach to development a long term episodic memory model featuring the false memory effect. We will explain motivation for the model, data structures used in the model and algorithms working over these structures. Finally we will present a prototype of an agent embodied in a 3D virtual world equipped with our model.

1 INTRODUCTION

Human memory is fallible: we do not remember everything we perceive, we forget, we may fail to retrieve information. A less traditional trait of memory fallibility - related to errors of commission rather than omission - is false memory [3]. Generally, false memory refers to "circumstances in which we are possessed of positive, definite memories - although the degree of definiteness may vary - of events that did not actually happen to us" [3, pp. 5]. Examples include remembering a *gist* of experience that may actually not correspond exactly to what has happened [3, 2, 20]; implanting distressing childhood memories, either accidentally at a psychotherapy [3, ch. 8][1, pp. 150] or in a laboratory experiment [18]; enhancing memories by or blending them with post-event information [17, ch. 4]; or fabrication of non-existing details of a criminal event by an eyewitness [3, ch. 6].

False memory is characteristic of normal rather then pathological remembering [3, 23]. Yet many people have only limited knowledge of false memory or may neglect its importance [28][1, pp. 151]. This can be particularly troublesome at courts and during psychotherapies. Therefore, the research on false memory phenomena (and increasing awareness of them) is important.

Computational approaches to memory modeling have become increasingly important in the past decade [21, 9]. In silico simulations enable a researcher to specify hypothetical mechanisms in precise detail, systematically explore the model and manipulate its parameters, and generate new predictions [26, 19, 25, 9]. It is known in computational cognitive sciences for some time that computational (neuro-)psychological episodic memory models, predominantly sub-symbolic ones, can produce some false memory-like phenomena (see [21], for a review of these models). However, to our knowledge, the issue of false memory has never been studied systematically in that field. At the same time, development of mathematical models of false memory by the community studying false memory directly is in its early stages [3, pp. 426-447].

In this position paper, we present our approach to computational modeling of false memory. We have been developing for about a year a generic episodic memory model featuring false memory characteristics, a model extending our previous episodic memory models [7, 4]. Of course there are some false memory characteristics that are out of our scope. The model is intended for acquisition, retention and retrieval of complex everyday events, such as cooking dinner (as opposed to events from laboratory tasks, e.g. presentations of lists of words). The memory representation is organized around memories of single objects (but not their features, e.g. not features of faces) and hierarchically nested events/episodes lasing from seconds to hours (e.g. knocking a door, opening the door, a visit) (see [5] for details). Our present aim is to develop architecture for false memory models rather than a single model fitting data from a particular experiment. Still, we believe that in future, when the model is stable enough, it can be used for the purpose of computational cognitive sciences. Additionally, because the underlying platform on which we test the model is a virtual character inhabiting a complex 3D virtual environment (see [6] for more on using VR for development of high-level cognitive models), the model can be also used in virtual reality applications. For instance, think of a serious game explaining to jurors limitations of eyewitness testimony with respect to false memory phenomena.

The rest of the paper is organised around the following points: 1) psychological underpinnings, 2) architecture of the model, 3) problems stemming from validating the model against human data, including human data acquisition.

2 GENERAL APPROACH

Our false memory model capitalizes on the fuzzy-trace theory [3, 12]. In a nutshell, this theory posits two parallel mechanisms that encode incoming information: *verbatim* and *gist*. While the former encodes the surface-form of the information in detail, the latter encodes the meaning in a coarse-grained way [12]. Of course, it may not be always clear what exactly a gist is. In our approach, the gist resembles the notion of a script [24], a knowledge structure about a stereotypical situation, including typical events that will occur and the most common deviations. The verbatim corresponds to a detailed log-based hierarchical representation of a particular flow of events as we used in [7]. The overall representation can be also linked to the event segmentation theory [29] and parts of the Conway's self-memory system, namely to episodic memories and general events [10].

Concerning recollection and familiarity, verbatim and gist mechanisms may operate in opposition to each other. For instance, when a memory trace for a particular detail is not strong enough, this detail may be replaced during recall by a different information "fabricated" based on the respective gist memory trace.

[1] Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic, email: mikajel@yahoo.com, rudolf.kadlec@gmail.com, brom@ksvi.mff.cuni.cz

3 ARCHITECTURE OF THE MODEL

Our cognitive architecture integrates a decision making module, a memory system, a perception module and an emotion generator. The architecture is detailed in [6]. The next section provides a brief introduction into already existing decision making module. Then the extended long term memory module of our architecture will be described.

3.1 Decision making module

Our agent is driven by the existing decision making module based on AND-OR trees [7]. Terminology used in our model largely comes out from the structures of this module. An AND-OR tree is a tree consisting of two types of nodes: AND nodes, also called actions in our model; and OR nodes, also called goals. The property of AND nodes (actions) is that in order to accomplish it, all its children must be performed. On the contrary the OR nodes (goals) can be completed by performing any of its direct children. AND nodes not containing any child can be performed directly and are also called *atomic actions*. The root of a tree is always an OR node and it is usually referred to as a *top-level goal*. All goals and actions can also have *affordance slots*, that are placeholders for objects, places, etc. that provide resources for a node's execution, i.e. they define the roles of missing objects. The term affordance [14] was coined by Gibson. The set of all AND-OR trees specifying an agent's behavior is denoted as D. The agent also has a short term memory module [7] that keeps a track of its current goals.

3.2 Elements of the episodic memory

Our memory structure for storing episodic memories is a pair $\langle C, S \rangle$ where C is a set of *chronobags* and S is a *schema bag*. A chronobag is a unit of memory representing a certain period of time, it stores the *episode structures* that model the verbatim of episodes experienced in that period. The term chronobag was first used in our paper [4]. A schema bag holds the gist of a typical episode of a certain type. The gist is represented by a statistics about co-occurrences of goals and their satisfying actions together with objects used by the agent. The following sections describe these components more closely. Figure 1 shows the structure of both C and S components of our episodic memory model.

3.2.1 Episode structures

An *episode structure* E is a tree-like structure consisting of *episodic nodes*, objects in affordance slot and time pointers. It incorporates all the actions performed while trying to satisfy one top-level goal. The root of the episode structures is always an episode node representing one top-level goal. Its children are actions that were performed in order to satisfy it.

Episodic nodes can represent either action, sub-goal or atomic action in the decision tree and the whole episode structure represents *action/goal traces* from the top-level goal to the atomic actions performed when trying to satisfy one top-level goal. If a node has more than one child, an order of execution of child nodes is stored in a *time pointer*.

When the agent performs an atomic action, the episode with the root node corresponding to the current top-level goal is located and episodic nodes reflecting the action/goal trace are added to the episode. If the agent performs the same action several times in a row, new nodes are stored in the memory only once. All objects used during execution of an action are linked with appropriate affordance slots. Instances of these object nodes are shared among all the episodes. Note that this structure is a core of our previous models [7, 6].

3.2.2 Chronobags

A chronobag is a structure for holding episodes experienced by the agent in a given time period. The memory can contain any number of chronobags, but will always contain at least one chronobag for episodes from the current day, this chronobag is called the *present chronobag*. Anytime a new episode is experienced by the agent, it will be stored in this chronobag. In all the chronobags, there is an ordered list of episode structures belonging to it. Moreover in the present chronobag, there is also a separate list for episodes that are not finished yet.

The action selection algorithm allows for temporary interruption of the top-level goal the agent is trying to accomplish. The agent can interrupt the current episode (i.e. performing actions satisfying the current top-level goal), experience another episode (accomplish another top-level goal) and return to the original episode (and original top-level goal) later. The present chronobag can therefore contain several opened episodes. Each time the top-level goal of an episode is successfully satisfied or the agent abandons its top-level goal, the particular episode is marked as finished and moved from opened episodes to finished episodes. This also happens to all opened episodes during the agent's sleep.

Chronobags are organized in a layered structure. In the lowest level there are chronobags for episodes from single days, in higher layers there are multiday chronobags that integrate episodes from lower level chronobags. The multiday chronobags hold episodes belonging to the period of time of its subordinate chronobags. Currently the model divides chronobags into four different layers, the most abstract layer incorporating episodes from 7-day period.

3.2.3 Schema bag

Specific part of our model is so called schema bag corresponding to the gist trace from the fuzzy-trace theory. It incorporates all the events the agent experienced during its existence and helps to determine how often the agent performed specific actions and how often it used specific objects. Any action, goal or atomic action from AND-OR trees experienced by the agent will have the associated node in the schema bag. These nodes are called *schema episode nodes*. Apart from these nodes, the schema bag also keeps separate nodes for each object the agent used during its lifetime. These are called *schema object nodes*. Schema bag also includes representatives of affordance slots and special nodes that connect object node with affordance slot it was used in. These special nodes are called *slot content nodes*.

Probably the most important component of the schema bag are *schema counters*. Schema counters keep track of how many times a set of schema nodes was executed/used by the agent. This set can contain schema episode nodes, slot content nodes, or both node types. The maximum set size is currently set to 3 due to combinatorial explosion problem. Schema bag not only provides information how often a specific node is executed or used, it also provides conditional probabilities $P(X|Y)$ where X and Y can be any set of schema nodes provided the combined size of sets X and Y is not larger than 3. Information deducible from the schema can

be for example: the agent visited a cinema 6 times so far; when commuting to work the agent used a bus 6 times out of 10.

Nodes in the schema bag and all the counters are updated on-line as the agent performs atomic actions.

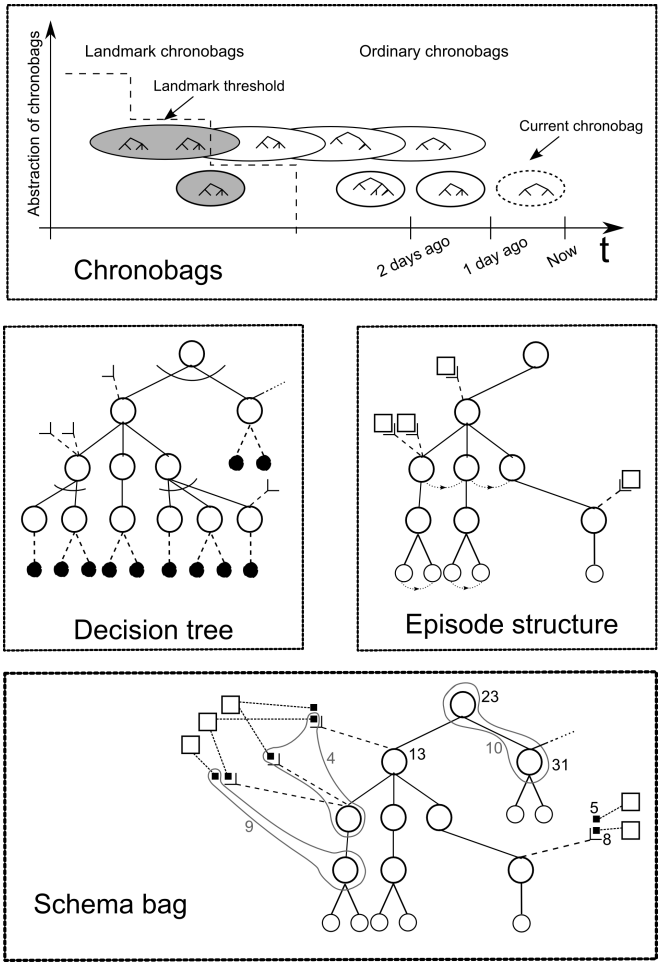

Figure 1. Different representations used by the memory model. The set of decision trees D represents procedural knowledge, it stores hierarchy of goals and actions satisfying those goals together with affordance slots (L-shaped figures) representing resources, and atomic actions (black circles) that can be performed directly in the agent's environment. The episode structure E represents actual experience. It is similar to one decision tree, but slots are already filled with object (squares). It also contains the sequence of episode nodes performed by the agent. Chronobags C hold the sets of episodes of similar age. As chronobags get older, details of episodes can be forgotten. When a chronobag gets old enough it becomes a landmark chronobag and it contains *fossilized* episodes that could not be forgotten. The boundary between ordinary chronobags and landmark chronobags is shown as a dashed line. Besides forgetting there is another process of creating more abstract chronobags for longer time periods. Details of lower level chronobags are merged into a higher level, more abstract chronobag spanning longer time period. The distinct chronobag on the right is the present chronobag used for storing current episodes. The last representation is schema bag S - schema bag is similar to the decision tree but it is extended with counts of how often each node was selected, how often each object was used in all affordance slots (black squares) and also it keeps a track of how many times different nodes and objects appeared in an episode together (there are three aggregate counts shown on the figure).

3.2.4 Example of filling the memory

For clearer conception of how new memories are added into the memory structures consider the following illustrative case. The agent starts with empty memory structures and he will try to fulfill the top-level goal *dinner* by performing following action *eating at restaurant*. To perform this action, he will have to complete the following subgoals: travel to a restaurant, order something to eat, eat it, and pay for the food.

When the agent starts following a new top-level goal, a new episode in the present chronobag will be created. The root of this episode will be the top-level goal *dinner*. This node will have one child node (*eat_at_restaurant*) and four grandchildren nodes (*travel*, *order*, *eat*, *pay*). Objects used will be also part of the episode: for example a *lobster* can be associated with the affordance slot *food* on the *eat* node. Each of these goals has to be completed by performing an action consisting of atomic actions executed by the agent in the virtual environment.

Apart from the episode structures, the schema bag is also being updated each time the agent performs an action. Imagine that the agent is sitting in the restaurant. The set of all schema nodes relevant to schema counter updating in this scene will be $S = S_{episode} \cup S_{slot\_content}$ where $S_{episode} = \{dinner, eat\_at\_restaurant, eat\}, S_{slot\_content} = \{lobster\_in\_food\_slot\}$. Then for each $X \subseteq S, |X| \leq 3$ the value of a schema counter will be increased by 1.

3.3 Processes maintaining the memory structure

One process behind maintenance of memory data structures deals with the acquisition of new information and it was explained in the previous section. Other processes described in this section are triggered during the agent's sleep and are more complicated. Some of these processes still have to be calibrated.

3.3.1 Shifting of chronobags

Shifting of chronobags simulates aging and generalization of episodic memories. There are two mechanisms working behind the chronobag shifting process each night:

1. Forgetting – as time passes chronobags are continuously being shifted back to the past. Age of chronobags is increased by one day every night. The present chronobag is moved to the set of past chronobags and new empty present chronobag is created. During every shift, some details of the episode can be gradually forgotten, as described later. This happens until the chronobag reaches age $t_l^{Landmark}$ when it becomes one of a *landmark* chronobags for the l-th level of chronobags. After this point no more details are forgotten from this specific chronobag. In literature this is referred to as a *flash bulb* memory, *flash bulb* memories are for example attacks from 9/11, birth of a child etc [8].
2. Episode merging – this process takes episodes from (non-landmark) consecutive chronobags, creates a chronobag representing union of time intervals of the chronobags being merged and copies all the contained episodes to it. This mechanism causes creation of several levels of abstraction of chronobags, with the daily chronobags being the least abstract chronobags. When the more abstract chronobag already contains a similar episode to the one being added, details of those episodes are merged, creating an "average" of the two. This is one of mechanisms for induction of false memories.

3.3.2 *Deriving an episode from the schema*

Existence of some nodes in the episode structure E can be deduced from other nodes in the episode with the use of schemas. If the conditional probability of existence of node n_1 given the existence of node n_2, that is $P(n_1|n_2)$ is close to 1, it means node n_1 does not have to be stored in episode E as long as node n_2 is not forgotten.

For example consider an agent that always goes to work by bus. Then the episode of *going to work by bus* happens every work day and it has the highest count among all ways of transport in the schema bag associated with going to work. It will be easily derivable from the schema (nodes *travel* and *work* will imply the existence of node *bus*). But when the agent oversleeps, it may use its car instead of the bus. Then this episode will not be derivable from the schema and its details should be remembered in a particular episode structure.

This mechanism helps to reduce the memory size and it would not cause any side effects if the derivability of nodes stayed constant during the existence of the agent and only derivable episodes would be forgotten. But in our model, even details that are not derivable can be forgotten, and in reality, the derivability of nodes can also change (because schemas are constantly updating). This process is another mechanism capable of inducing a false memory. Consider for example *going to work* episode mentioned above. If the node *car* is forgotten, the model will derive the node *bus* instead and the agent will not be able to distinguish this false memory from any other stored memory.

3.3.3 *Details of forgetting*

The forgetting of episode nodes is performed using the node's $score$. Each node is assigned a numeric score:

$$score = \sum_{a \epsilon Attributes} weight_a \cdot value_a \qquad (1)$$

based on the following $Attributes$ set: the user defined salience, the frequency of executing the node, the ability to derive its existence with the use of schema bag, the salience of objects attached to the node, the number of subnodes. Generally the score is higher for more interesting nodes: those more salient, less frequently executed and those that cannot be derived from the schema. Weights of all attributes will be fine-tuned during more complex testing of the model.

An important feature of the model is that the scores do not change in time. However, each chronobag has only limited capacity based on its age and the saliency of nodes in it. The capacity is currently calculated according to the formula:

$$capacity = MaxCapacity \cdot \frac{1}{a_l * t + 1} + b \qquad (2)$$

where t is the chronobag's age, a_l is a coefficient based on the chronobag's level of abstraction and b is a parameter used to increase capacity of chronobags with many salient nodes. The node scoring mechanism (Eq. 1) together with the limited capacity of chronobags (Eq. 2) should result in a believable forgetting process.

4 IMPLEMENTATION

The memory model is being developed as a standalone Java library independent of the agent's decision making system (DMS). The current implementation is divided into three separate projects:

- Bot – this library includes the DMS of the agent (in this case AND-OR trees) and it controls the agent's body through the Pogamut platform [13]. Pogamut is a tool for programming agents in virtual 3D environment.
- Memory ↔ Pogamut interface – a lightweight layer translating events originating in the agent's DMS into representation used in the episodic memory.
- Episodic memory model – a standalone library implementing the core of the model, that is: chronobags, schema bag, the chronobag shifting algorithm (see Section 3.3) and a GUI for exploring the content of the memory (see Figure 2). There is a clearly defined API used to insert information into the memory. AND-OR trees are the default formalism used by the model but any other DMS with hierarchical nature can be connected to the memory module too. The model works with the notion of more and less abstract actions (or goals), it does not matter whether those actions are implemented in the DMS as Hierarchical Finite State Machines, AND-OR trees etc.

This modular architecture makes it possible to connect our model to any other source of data without much effort in the future. For new environments, only the lightweight interface translating events to the format expected by the core memory model has to be implemented. The core episodic memory model can remain unchanged.

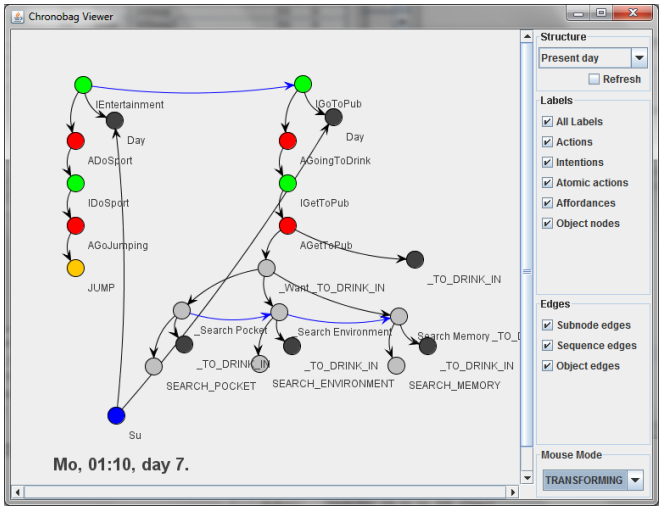

Figure 2. The GUI of the Episodic memory module showing a content of one chronobag. It shows two very simple episodes executed by the agent. The GUI is able to display contents of any chronobag, decision tree or schema structures. JUNG library [22] was used for visualization of the graph.

5 VALIDATION OF THE MODEL

Natural question is how the model will be validated and parameterized. Research on human memory provides only limited data about function of memory outside psychological laboratory. There are many experiments with memorizing lists of words, non-sense syllables and figures, but fewer results about working of memory in daily life on a scale of months, years or an entire life. For purposes of episodic memory modeling it would be best to have data with:

1. Input of the memory - e.g. all events, objects and other actors that the subject was exposed to.

2. Content of the memory - which of the inputs were remembered and the depth of detail that can be recalled.

Concerning Point 1, for longer time scales there is no such data yet, however this may change in a near future. Dawn of devices like Microsoft SenseCam [15] can generate tons of data about inputs of memory that we are currently lacking. Considering already existing published data, Wagenaar's six year diary study [27] and video based study of real life events [11] seems to be the closest matches to our requirements.

Concerning Point 2, the exact content of memory remains unknown, we can study it only through recollection and recognition experiments. Our aim is to fit data from this kind of experiments with human subjects.

In our methodology we want to create simulation on the scale of several months (and later a life time simulation) of our agent to obtain inputs of the memory. In a 3D simulated world we can log every subtle detail of the environment. After we obtain this log of information, we will use it as an input of our memory model and try to fit data dealing with false memories reported in [3, 16] and the data dealing with forgetting curves and retention intervals reported in [1, 11, 17].

We plan to perform several experiments:

1. The first is to prove that the model can recall episodes that did not happen but are compliant with the schemas. To do this we will perform simulation of three weeks with one set of plans the agent will be following and then one additional week with slightly modified plans. We expect to find reasonable parameters of our model, where a false memories will appear. We will try to fit the data reported in [16].
2. The second experiment should find parameters for a model that will approximate the retention curve of remembered memories. We will try to fit the real life data for several retention periods going from one day to several weeks, as reported in [1, 11, 17].
3. In the next experiment we will try to find out if our memory model is able to support a hypothesis that memory dating errors peak at multiples of seven days, as reported in [17].
4. We also consider creating a setting for the experiment where the agent's recollections of different events and items will be ordered. We want to parameterize the model so that less errors will be made in items recollected earlier, as reported in [17].

6 CONCLUSION

We have presented our computational model of long term episodic memory that aims to model false memory effect. The model capitalizes on our previous work [7, 4] and extends it with a notion of a schema bag and a chronobag shifting algorithm (Section 3.3). The chronobag shifting algorithm combining both gradual forgetting and episode merging was briefly described. We believe that these two mechanisms together with node derivability (Section 3.3.2) can result into emergence of false memory effects well known from psychological literature. However our model is currently a work in progress, the validation of the model against data from psychology will be the next step.

ACKNOWLEDGEMENTS

This work was partially supported by the student research grants GA UK 44910 and 21809, by the grant GACR 201/09/H057 and by the research project MSM0021620838 of the Ministry of Education of the Czech Republic.

REFERENCES

[1] A.D. Baddeley, M. Eysenck, and M.C. Anderson, *Memory*, Hove: Psychology Press, 2009.
[2] F.C. Barlett, 'Remembering: A study in experimental and social psychology', *University Press, Cambridge*, (1932).
[3] C.J. Brainerd and V.F. Reyna, *The science of false memory*, Oxford University Press, 2005.
[4] C. Brom, O. Burkert, and R. Kadlec, 'Timing in Episodic Memory for Virtual Characters', in *Proceedings of CIG 2010*, (2010).
[5] C. Brom and J. Lukavský, 'Towards virtual characters with a full episodic memory II: The episodic memory strikes back', in *Proc. Empathic Agents, AAMAS workshop*, pp. 1–9, (2009).
[6] C. Brom, J. Lukavský, and R. Kadlec, 'Episodic Memory for Human-like Agents and Human-like Agents for Episodic Memory', *International Journal of Machine Consciousness*, **2**(2), 227–244, (2010).
[7] C. Brom, K. Pešková, and J. Lukavský, 'What does your actor remember? towards characters with a full episodic memory', *Virtual Storytelling. Using Virtual Reality Technologies for Storytelling*, 89–101, (2007).
[8] R. Brown and J. Kulik, 'Flashbulb memories', *Cognition*, **5**(1), 73–99, (1977).
[9] N. Burgess, 'Computational models of the spatial and mnemonic functions of the hippocampus', in *The Hippocampus Book*, Oxford University Press, (2006).
[10] M.A. Conway, 'Memory and the self', *Journal of Memory and Language*, **53**(4), 594–628, (2005).
[11] O. Furman, N. Dorfman, U. Hasson, L. Davachi, and Y. Dudai, 'They saw a movie: long-term memory for an extended audiovisual narrative', *Learning & Memory*, **14**, 457–467, (2007).
[12] D.A. Gallo, *Associative illusions of memory: False memory research in DRM and related tasks*, Psychology Press, 2006.
[13] J. Gemrot, R. Kadlec, M. Bída, O. Burkert, R. Píbil, J. Havlíček, L. Zemčák, J. Šimlovič, R. Vansa, M. Štolba, et al., 'Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents', *Agents for Games and Simulations*, 1–15, (2009).
[14] J.J. Gibson, *The ecological approach to visual perception*, Lawrence Erlbaum, 1986.
[15] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood, 'SenseCam: A retrospective memory aid', *UbiComp 2006: Ubiquitous Computing*, 177–193, (2006).
[16] J.M. Lampinen, S.M. Copeland, and J.S. Neuschatz, 'Recollections of things schematic: Room schemas revisited', *Journal of Experimental Psychology: Learning, Memory and Cognition*, **27**, 1211–1222, (2001).
[17] E. Loftus, *Eyewitness Testimony*, Harvard University Press, 1979.
[18] E. Loftus, 'Creating false memories', *Scientific American*, **277**, 70–75, (1997).
[19] S.C. Marsella and J. Gratch, 'EMA: A process model of appraisal dynamics', *Cognitive Systems Research*, **10**(1), 70–90, (2009).
[20] U. Neisser, 'John Dean's memory: A case study', *Cognition*, **9**(1), 1–22, (1981).
[21] K.A. Norman, G.J. Detre, and S.M. Polyn, 'Computational models of episodic memory', *The Cambridge handbook of computational cognitive modeling*, 189–225, (2008).
[22] J. O'Madadhain, D. Fisher, and T. Nelson. Java Universal Network/Graph Framework. `http://jung.sourceforge.net`, 2011.
[23] D.L. Schacter, *The seven sins of memory: How the mind forgets and remembers*, Mariner Books, 2002.
[24] R.C. Schank and R.P. Abelson, *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*, volume 2, Lawrence Erlbaum Associates Hillsdale, NJ, 1977.
[25] R. Sun, *The Cambridge handbook of computational psychology*, Cambridge University Press, 2008.
[26] T. Tyrrell, *Computational mechanisms for action selection*, University of Edinburgh, 1993.
[27] W.A. Wagenaar, 'My memory: A study of autobiographical memory over six years', *Cognitive psychology*, **18**(2), 225–252, (1986).
[28] R.A. Wise and M.A Safer, 'A Survey of Judges' Knowledge and Beliefs About Eyewitness Testimony', *Court review*, 6–16, (2003).
[29] J.M. Zacks and K.M. Swallow, 'Event segmentation', *Current Directions in Psychological Science*, **16**(2), (2007).

A Preliminary Functional Analysis of Memory in the Word Sense Disambiguation Task

Nate Derbinsky[1] **and John E. Laird**[1]

Abstract. We focus on the problem of efficiently retrieving knowledge from large memories given ambiguous cues. First, we analyse the word sense disambiguation task in context of memory model comparison and evaluation. Then, in this task, we demonstrate the functional benefit of two forms of memory retrieval bias, recency and frequency of memory access, and present a preliminary evaluation of heuristics to efficiently support these biases in memory systems.

1 INTRODUCTION

One advantage the human brain demonstrates over the current generation of artificially intelligent agents is its ability to extract diverse, useful experiences from its interactions with the world; store large amounts of this information in memory for long periods of time; and later retrieve this knowledge from memory when it is relevant to making decisions and taking action. There is evidence that extending agents with long-term memory supports many functional cognitive capabilities [1]; however, maintaining and querying large memories poses significant computational challenges that currently make it impossible to task these agents with real-world problems.

The focus of this paper is one specific challenge facing long-term memory: given a large store of knowledge, how should the system respond to an ambiguous cue, one that pertains to multiple previously encoded memories. Anderson and Schooler [2], positing that human memory optimally solves this problem with respect to the history of past memory access, have developed and validated memory models that are widely used in the cognitive modelling community. However, existing computational implementations of these long-term declarative memory models do not scale to large bodies of knowledge [3].

Previously [4] we developed and evaluated computational techniques to efficiently support queries of large declarative memory stores; however, this work supported only a limited class of bias in the case of ambiguous cues. In this paper, we extend our prior work and evaluate methods for efficiently incorporating recency and frequency of memory access as functional models of memory retrieval bias. We evaluate our methods in the word sense disambiguation (WSD) task, an important and well-studied problem in the Natural Language Processing (NLP) community [5]. We are not attempting to solve the word sense disambiguation problem, but instead our work is intended to provide evidence that (1) the WSD task is an appropriate benchmark when evaluating and comparing memory models and that (2) agents whose long-term memory systems incorporate historical regularities of past memory access will benefit in this task.

[1] Computer Science and Engineering Division, University of Michigan, 2260 Hayward St., Ann Arbor, MI, 48109-2121. Email: {**nlderbin**, **laird**}**@umich.edu**.

We begin with an introduction to the word sense disambiguation task, including an analysis of WordNet [6] and SemCor [7], the datasets we use in our evaluation. We then present results of how simple baseline algorithms perform on the WSD task, including the relative advantage of memory-based algorithms that incorporate the recency and frequency of memory access. Given this performance advantage, we evaluate the WSD performance of the base-level model of memory bias [8], a commonly used model based upon the rational analysis of memory [2] that combines recency and frequency of memory access. As the base-level model performs relatively well in this task and dataset, we motivate, describe, and evaluate preliminary heuristics to efficiently implement this model in a long-term memory system. Finally, we conclude with a discussion of this and future work.

2 WORD SENSE DISAMBIGUATION

Many words in the English language are *polysemous*, that is they have multiple, distinct meanings, or *senses*, which are interpreted differently based upon the context in which they occur. For instance, consider the following sentences:

a) Deposit the check at the *bank*.
b) After canoeing, they rested at the *bank*.

The occurrences of the word *bank* in the two sentences clearly denote different meanings: ‘financial institution’ and ‘side of a body of water,’ respectively. Word sense disambiguation is the ability to identify the meaning of words in context in a computational manner [5]. The task of WSD is critical to the field of NLP and has been studied for decades. There are several formulations of and many approaches to this problem.

As the focus of this work is memory, not language processing, we simplify components of the general WSD problem and adopt a variant of the *lexical sample* formulation of the problem. As input, the agent receives a sequence of sentences, each composed of a sequence of words. For simplicity, each word in the input is tagged with its appropriate part of speech (noun, verb, adjective, or adverb). Additionally, the agent has access to a machine-readable dictionary (MRD), such that each lexical word/part of speech pair in the input corresponds to a list of word senses within the MRD. For each sense, the MRD contains a definition and tag frequency from a representative corpus. Thus, for each word in each input sentence, the agent’s task is to select the most appropriate sense from the MRD.

3 TASK ANALYSIS

The data source we use is version 3 of the SemCor [9] *semantic concordance*. A semantic concordance is a textual corpus and lexicon linked such that every substantive word in the text is linked to its appropriate sense in the lexicon [7]. SemCor is the largest and most used sense-tagged corpus, which includes 352

texts from the Brown corpus [10]. We use the 186 Brown corpus files that have all content words tagged, which includes more than 185,000 sense references to version 3 of the WordNet lexicon [6]. WordNet 3, the most utilized resource for WSD in English, includes more than 212,000 word senses.

To understand the task at hand, it is useful to examine certain properties of WordNet and SemCor. We begin with aggregate sense size in WordNet, which measures the number senses per lexical word/part of speech pair, and thus the average *ambiguity* faced by an agent attempting to disambiguate an arbitrary input word. In total, the average number of senses per word is 1.33 (std. deviation 1.12), with a minimum of 1 and a maximum of 59. If we aggregate these statistics with respect to word part of speech, we see the breakdown reported in Table 1 (sorted in order of increasing maximum sense size).

Part of Speech	Min.	Max.	Avg.	Std. Dev.
Adverb	1	13	1.2453	0.7379
Adjective	1	27	1.3968	1.0731
Noun	1	33	1.2421	0.8563
Verb	1	59	2.1725	2.5128

Table 1. Part of Speech Sense Size Statistics in WordNet 3.

This summary statistic, however, refers only to WordNet, and thus does not take into account the distribution of words within the SemCor texts. Table 2 indicates this proportion of words in the SemCor data set, aggregated by part of speech. These statistics reveal that nearly 73% of the words fall into the two most ambiguous parts of speech (nouns and verbs).

Part of Speech	Proportion of SemCor
Adverb	10.2%
Adjective	17.1%
Noun	47%
Verb	25.7%

Table 2. SemCor Part of Speech Proportion.

For about 0.33% of SemCor words, multiple senses are equally appropriate within the linguistic context, as annotated by a human interpreter. Thus we introduce *effective* sense size, computed as one divided by ambiguity in a given input context. For example, consider the following sentence from SemCor:

Pansies are supposed to like it cool, but those great velvety flowers were healthy and perky in the glaring sun.

The verb "to like" has five senses in WordNet and in this context, two are considered equally appropriate ("find enjoyable or agreeable" and "want to have"). So while the sense size of this word is 5, its *effective* sense size in context is (1/[2/5]) = 2.5.

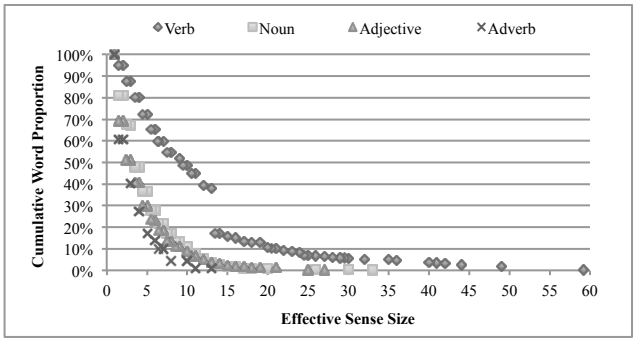

Figure 1. Cumulative SemCor Word Proportion vs. Effective Sense Size

Figure 1 synthesizes the data from Tables 1 and 2, plotting cumulative proportion of words against effective sense size, aggregated by part of speech. This chart is rich with useful trends and statistics that we consider here. First, the data exhibits a strong right skew, containing mostly words with few effective senses. Next, we draw out the proportion of words that require no disambiguation, as their effective sense size is 1, by reading the second plotted point for each part of speech. While for adverbs and adjectives this value is about 40% and 30%, respectively, for nouns and verbs it is about 20% and 5%. Next, we can assess the median effective sense size for each part of speech by reading the x-axis as each part of speech intersects 50% on the y-axis. For adverbs, adjectives, and nouns, this value is between 2 and 4, while for verbs this it is about 10. In summary, the average effective sense size in the SemCor dataset is between 2 and 3 and the expected performance on the WSD task, given a random selection strategy, is 38.73%.

	SemCor Task Performance
Random	38.73%
Static Frequency	76.39%
Lesk	63.40%
Simplified Lesk	65.52%

Table 3. Non-Memory Baseline Results.

4 NON-MEMORY BASELINE ALGORITHMS

To contextualize performance of memory-based algorithms on the WSD task, we first implemented some non-memory baseline algorithms. The results from these baselines are summarized in Table 3, including *random* selection, derived in the previous section. All algorithms we implement select a word sense for all input words, and thus precision and recall are identical for all results, so we report them jointly as "task performance." Each task performance result we report is the true accuracy of the algorithm on this dataset, as opposed to the sample average of individual probabilistic runs; consequently, even small differences in performance should be considered relevant.

The first baseline was a frequency-biased random selection strategy. As previously described, WordNet includes, for each sense, a static tag frequency from the Brown corpus. As the SemCor textual corpus is a subset of the Brown corpus, we expected this information to be highly informative during sense disambiguation, and unsurprisingly this algorithm yielded nearly twice the performance of pure random selection.

The remainder of the non-memory baselines were variants of the Lesk algorithm for word sense disambiguation [11]. The Lesk algorithm, a commonly used baseline metric [12], assumes that words in a given "neighbourhood" (such as a sentence) will tend to share a common topic, and thus biases sense selection based upon shared terms in sense definitions and context. The classic algorithm finds the maximum overlap between all definitions of all candidate senses in the neighbourhood, and is thus computationally intractable as the size of the neighbourhood grows, so it is common to introduce constant-sized neighbourhood "windows" to reduce search time. A "simplified" Lesk algorithm [13] defines word context as simply the terms in the neighbourhood, as opposed to their definitions, and thus has more tractable growth, scaling with the sense size for the word to

be disambiguated. The performance of the Lesk family of algorithms is highly sensitive to the exact wording of sense definitions, and so it is common to supplement Lesk with heuristics and additional sources of semantic meaning (ex. [14]).

For the classic algorithm, we evaluated neighbourhood windows of size 2 and 4 in each sentence. For both classic and simplified, we also evaluated four heuristics. The first was the use of a *stop* list, which excludes definition terms that are common to the target language, such as "a" and "the." The second was to exclude example sentences from sense definitions, as the example terms might pollute overlapping computations. The third was the use of the Porter Stemming [15] algorithm to strip word suffixes with the intention of facilitating overlap of words with common linguistic roots. Finally, we included a bias towards the corpus frequency information described above. We evaluated the full combinatorial set of these parameters across both algorithms. The maximal results (see Table 3) for both classic and simplified algorithms occurred using the stop list, pruned definitions, and frequency bias, but not the Stemming algorithm. For the classic algorithm, the neighbourhood size that yielded greatest task performance was 2. Again, these results are simplistic, very specific to our implementation and data sets, and are not intended for representation of or comparison to modern NLP techniques, but instead to provide a baseline for later memory-based results. These results suggest that we can apply basic techniques, which do not incorporate memory-based methods, and expect to disambiguate up to 76.39% of input words in SemCor.

5 MEMORY BASELINE ALGORITHMS

Inspired by non-memory frequency bias results (see Table 3) and the rational analysis of memory [2], this section investigates algorithms for maintaining a dynamic history, or *memory*, of information presentation and evaluates the degree to which these memories facilitate effective sense selection.

In context of the WSD task, the algorithms described below differ in what information constitutes a history of past sense assignment, and how this information is maintained over time, as well as how, when presented with a word to disambiguate, this history is resolved to select a word sense. Therefore, to evaluate these algorithms, we applied a common evaluation sequence. First, for each word in each input sentence, we performed a read-only *query*, the result of which was scored. We then *presented* to the algorithm the correct sense (or senses) that had been annotated within the SemCor test set. Revealing the correct sense(s) to the memory system eliminates the possibility of unintended divergent learning, which could occur without truthful feedback and would obfuscate the algorithm results.

Unlike the non-memory baselines, these algorithms have the potential to improve with added exposure to the corpus, and thus we performed 10 sequential runs of each. The results are summarized in Table 4 and report task performance on the 1st and 10th run on the SemCor test set.

The first algorithm we evaluated was *recency* of presentation. This algorithm maintains only the most recently presented sense for each lexical word/part of speech pair, which is returned at the time of the next query of the same pair (selecting randomly from amongst multiple simultaneous presentations). This algorithm performs well if the same word sense is used repeatedly in immediate succession.

The next algorithm was *frequency* of presentation. This algorithm maintains the number of presentations of each word sense and then selects the most frequent sense at the time of query. This algorithm performs well if particular senses of words are generally more common than others in a corpus, as opposed to being highly dependent upon sentence context. As an experimental condition, we initialized the frequency of each word sense to its absolute frequency within the full Brown corpus. We found that this initialization provided more than 4% improvement on the first run, but the improvement was only about 0.1% after 10 runs. This condition is labelled "Frequency*" in Table 4. This is comparable to the "Frequency Bias" result in Table 3, with the added improvement in Table 4 coming from updated frequency values as the algorithm gains exposure to the corpus.

Finally, to establish an upper bound on the degree to which recency and frequency can individually contribute to WSD performance, we implemented an *oracle* algorithm. For each word query, this algorithm scores both the *recency* and *frequency* algorithms described above and returns the result that provided the greater score. As with *frequency*, we label as "Oracle*" the variant that initializes frequency with overall corpus frequency. This algorithm performs well for a word query when either recency or frequency is informative to effective sense selection.

	Run 1	**Run 10**
Recency	72.34%	74.43%
Frequency	71.69%	76.53%
Frequency*	75.97%	76.62%
Oracle	79.51%	84.08%
Oracle*	83.87%	84.18%

Table 4. Memory Baseline Results.

We draw three conclusions from the data summarized in Table 4. First, we note that with the exception of pure *recency*, which does not achieve *frequency bias* performance, all memory-based algorithm results for run 10 are greater than all non-memory baselines (see Table 3). This result suggests that memory access history in SemCor, with very little corpus exposure, yields a performance benefit in the WSD task, an advantage that is not dependent upon MRD definition quality (unlike Lesk and its variants). Second, based upon the run 10 results, we can expect memory-endowed agents to disambiguate up to about 84% of SemCor words, simply via memory retrievals, with the potential to improve performance with additional reasoning. Finally, in comparing the run 10 results of "Frequency" vs. "Frequency*" and "Oracle" vs. "Oracle*" we have preliminary evidence that it is unnecessary to bootstrap learning in frequency-biased memories with corpus-specific initialization information, as the empirical history of presentation within the text corpus quickly captures these regularities.

6 MEMORY BIAS MODEL

We have presented evidence that recency and frequency of memory access yield performance benefits in the WSD task on the SemCor dataset. However, to apply these findings to a memory system, we require a model of how these properties combine to bias selection of word senses (recall that the *oracle* algorithm in the previous section is not possible to implement, as

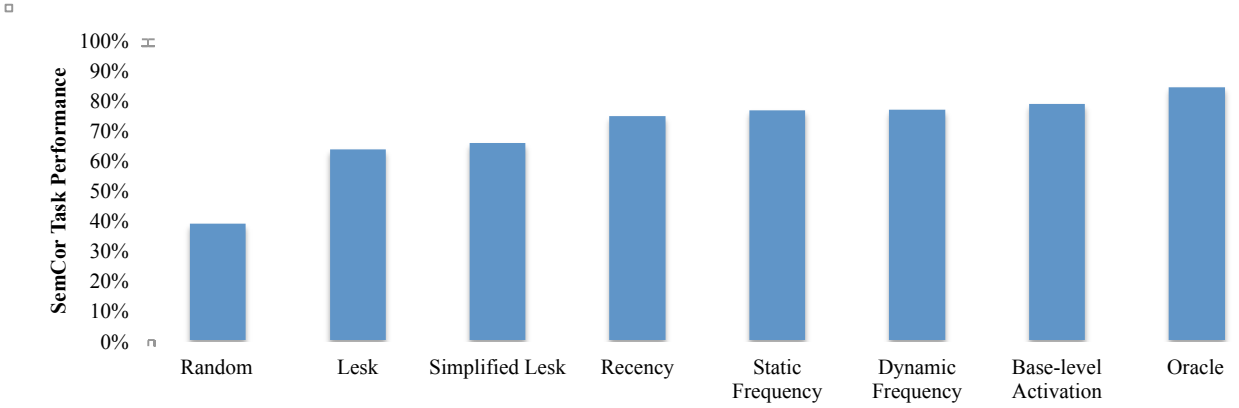

Figure 2. SemCor task performance comparison of non-memory baseline algorithms and run 10 memory-based algorithms.

it requires the memory system to evaluate correct sense assignments during word sense selection).

The base-level activation component of the declarative memory module in the ACT-R cognitive architecture [8], based upon the rational analysis of memory [2], offers one such model that has been widely used within the cognitive modelling community. This model computes activation bias of a memory according to the following equation:

$$\ln(\sum_{j=1}^{n} t_j^{-d})$$

where n is the number of presentations of the memory, t_j is the time since the j^{th} presentation, and d is a free decay parameter.

We evaluated this model in the same fashion as the memory baseline algorithms. The only experimental condition was the value of the decay parameter, over which we performed an exploratory sweep of 12 values between, but not including, 0 and 1, and found that d=0.7 resulted in the best run 10 performance. We found that base-level activation yielded 74.45% task performance on SemCor in the first run and 78.47% after 10 runs. This run 10 task performance bests all non-memory and memory baselines, and the run 1 task performance is an improvement to the *recency* and *frequency* run 1 results.

Figure 2 summarizes SemCor task performance. It includes all non-memory baselines, as well as those memory baselines that are not initialized using corpus-specific information. Note that for clarity, we have re-named the memory form of *frequency* as "Dynamic Frequency."

7 EFFICIENT MEMORY BIAS

Data in the previous section demonstrates that the base-level model is relatively effective in the WSD task on the SemCor dataset. Additionally, base-level activation is the dominant model used for cognitive models of human memory phenomena. However, it has been shown to not scale to large bodies of knowledge for long-lived agents [16]. Thus we consider here the challenges associated with efficiently integrating this model as a source of bias within a long-term memory system.

One scaling issue arises when calculating base-level bias as presentation history (n) grows large. However, there are known methods to mitigate this problem (such as a constant-sized history) and so we do not discuss this issue further.

The primary scaling challenge occurs because the base-level model includes a sum over memory presentations (t_j), and that these temporal distances change at *each* point in time for *every* memory. Thus, a naïve integration of the base-level model as a source of bias in memory retrieval must calculate bias values for each candidate memory, a potentially expensive computation given large memory stores and ambiguous cues. Our prior work [4] details methods for efficiently biasing memory retrievals, assuming that only a *constant* number of memories change bias value at any point in time. To better satisfy this assumption for the base-level model, the remainder of this section provides a preliminary evaluation, within the WSD task, of two novel heuristics that seek to identify *only* those memories for which bias *must* be calculated during a memory retrieval. We begin with a description of evaluation metrics and baselines, describe the heuristics, and analyse our results (Table 5).

The first metric, *updates*, refers to the number of memories that require bias calculation during retrieval (lower is better). The *average* is a measure of expected efficiency and the *maximum* refers to expected reactivity. The second metric, *validation*, is a measure of quality and refers to the proportion of queries in the WSD task that result in the same word sense as a *naïve* baseline, wherein bias calculations are performed for all candidate memories. Table 5 also includes a *stable* baseline, wherein memory bias calculations only occur at the time of presentation. This heuristic exploits a regularity of the base-level model: from the time that bias is calculated for a memory, this value is guaranteed to *over-estimate* the true bias value until the memory is presented again in the future. Note that while this heuristic requires no updates, validation suffers by 27.15%.

Our first novel heuristic, *NT*, refers to the (N)umber of memory accesses and most recent (T)ime of access. Both of these statistics can be efficiently maintained incrementally and we reasoned that if *neither* of these values of memory A is greater than that of memory B, it is unlikely that the bias value of A is greater than that of B, and therefore it is unnecessary to compute the bias value of A. This heuristic reduces the average number of updates by 55%, as compared to *naïve*, and maintains a very high level of validation, as compared to *stable*, but has no impact on maximum updates.

To reduce maximum updates, we developed a second novel heuristic, *NTM*, which augments *NT* with incremental (M)aintenance of memories: NTM clears the presentation history, and updates the bias value, of those memories for which the time since the most recent presentation is greater than a

threshold (τ). This type of incremental maintenance can be implemented efficiently [17] and we reasoned that the result over time would be many fewer memory candidates with substantive presentation histories. We found that increasing the maintenance threshold (i.e. permitting "older" histories) had the effect of decreasing average updates, while increasing maximum updates and validation; however, due to the exponential decay of the base-level model, all three metrics exhibited "knees". We performed a preliminary sweep of the threshold parameter on SemCor and report the data in Table 5 for the setting that yielded the fewest maximum updates and greatest validation (τ=10). The result is a more than 80% reduction in maximum updates, as compared to *NT*, while maintaining a high level of validation, as compared to *stable*, and a moderate average number of updates, as compared to *NT* and *naïve*. This data represents initial evidence that a high-fidelity base-level model can be efficiently implemented in a memory system, even as the number of memories grows large.

	Avg Updates	Max Updates	Validation
Naïve	2.94	31	100%
Stable	0	0	72.85%
NT	1.32	31	100%
NTM	1.74	6	99.87%

Table 5. Heuristic Results (τ=10).

8 DISCUSSION

We have analysed a formulation of the WSD task on the SemCor data set and have shown preliminary evidence that recency and frequency of sense assignment, biases common to human-inspired computational memory models, are beneficial to task performance. We have also presented evidence that in this task and data set, the base-level model [8] is effective at combining these properties as a source of retrieval bias, and we described and evaluated preliminary heuristics to efficiently incorporate base-level activation within a long-term memory system.

Our over-arching goal is to develop long-term memory mechanisms that are efficient and effective across a wide variety of tasks. We have made progress towards evaluating one class of memory bias on one data set for the word sense disambiguation task, but this leaves much future work to be done. First, to make sure we are not over fitting for the SemCor dataset, we plan to evaluate additional WSD datasets (ex. SENSEVAL [13]). Furthermore, to gather evidence that recency and frequency of memory access are generally useful in a long-term memory system, we plan to evaluate these biases in tasks other than WSD. While the base-level model has been shown to be effective, there is additional room for improvement, as illustrated by the task performance of the *oracle* algorithm, and thus we also plan to develop and evaluate additional memory bias models. We also plan to evaluate the computational run-time of bias algorithm implementations within real agents, such as to understand the trade-offs between computational efficiency, model validity, and bias functionality. Finally, our results demonstrate that memory retrievals alone can successfully disambiguate up to 84% of the words in SemCor, but what of the remaining words? We plan to integrate this work into running agents and explore the interactions of memory retrievals and other, complimentary processing mechanisms and sources of knowledge.

ACKNOWLEDGEMENTS

The authors acknowledge the funding support of the Air Force Office of Scientific Research.

REFERENCES

[1] Nuxoll, A., Laird, J. E. Extending Cognitive Architecture with Episodic Memory. In: *Procs. of AAAI* (2007).
[2] Anderson, J., Schooler, L. Reflections of the Environment in Memory. In: *Psychological Science*, 2 (6), pp. 396-408 (1991).
[3] Douglass, S., Ball, J., Rodgers, S. Large Declarative Memories in ACT-R. In: *Procs. of ICCM* (2009).
[4] Derbinsky, N., Laird, J. E., Smith, B. Towards Efficiently Supporting Large Symbolic Declarative Memories. In: *Procs. of ICCM* (2010).
[5] Navigli, R. Word Sense Disambiguation. *ACM Computing Surveys*, 41 (2), pp. 1-69 (2009).
[6] Miller, G. A. WordNet: A Lexical Database for English. *Communications of the ACM*, 38 (11), pp. 39-41 (1995).
[7] Miller, G. A., Leacock, C., Tengi, R., Bunker, R. T. A Semantic Concordance. In: *Procs. of DARPA Workshop on Human Language Technology* (1993).
[8] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., Qin, Y. An Integrated Theory of the Mind. In: *Psychological Review*, 111 (4), pp. 1036-1060 (2004).
[9] Mihalcea, R., http://www.cse.unt.edu/~rada/downloads.html
[10] Francis, W. N., Towell, G., Voorhees, E. M. Towards Building Contextual Representations of Word Senses Using Statistical Models. In: *Procs. of Workshop on the Acquisition of Lexical Knowledge from Text* (1993).
[11] Lesk, M. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In: *Procs. of SIGDOC* (1986).
[12] Vasilescu, F., Langlais, P., Lapalme, G. Evaluating Variants of the Lesk Approach for Disambiguating Words. In: *Procs. of LREC* (2004).
[13] Kilgarriff, A., Rosenzweig, J. English SENSEVAL: Report and Results. In: *Procs of LREC* (2000).
[14] Banerjee, S., Pedersen, T. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: *Lecture Notes in Computer Science*, 2276:136-145 (2002).
[15] Porter, M. F. An Algorithm for Suffix Stripping. In: *Program: Electronic Library and Information Systems*, 40 (3), pp. 211-218 (2006).
[16] Kennedy, W. G., Trafton, J. G. Long-Term Symbolic Learning. In: *Cognitive Systems Research*, 8 (3), pp. 237-247 (2007).
[17] Nuxoll, A. M., Laird, J. E., James, M. Comprehensive Working Memory Activation in Soar. In: *Procs. of ICCM* (2004).

On Memory Systems for Companion Robots: Implementation Methodologies and Legal Implications

Paul Baxter[1], Rachel Wood[1], Tony Belpaeme[1] and Marco Nalin[2]

Abstract. Companion robots are becoming increasingly prevalent in a wide variety of domains. The development of realistic long-term human-robot interaction is desirable and this entails the extension of interactions over multiple episodes. Memory systems are thus required in support of this goal. While current memory systems for artificial agents (and companion robots in particular) are currently restricted to symbolic database structures, this is not guaranteed to remain the case, with an increasing number of approaches using sub-symbolic representation schemes. This position paper explores the legal and ethical consequences of this shift of perspective by examining a range of solutions to the problem of data removal from artificial memory systems, specifically in the context of healthcare applications, and concludes that the current legislative provisions for data processing and protection may be inadequate for the next generations of companion robots.

1 INTRODUCTION

Artificial (robotic) companion agents are becoming increasingly important in the domain of human-robot interaction studies and applications. As this line of research advances, increasing emphasis is being placed on extended interactions, in which the robot and human participants engage in multiple, extended and progressive interactions, rather than the single stand-alone interaction episodes that are generally the focus of human-robot interaction. With this shift in emphasis, it becomes increasingly necessary for the artificial agent (be it purely virtual, or embodied in a physical robotic platform) to use information personal to the human participant [1][2], whether this information has been acquired only from prior interaction, or from another source (such as pre-existing details from a personal medical record). This information may be used to adapt the agent's behaviour to suit the person's needs, including the ability to sustain interactions, and to provide better social support – i.e. to provide higher levels of companionship.

In the context of health-related applications, robotic companion agents may play a number of different roles. Three broad cases may be envisaged: (1) the robot acts as an entertainment device, with no explicit healthcare functionality; (2) the robot has a medical role (such as in the support of clinical objectives, e.g. rehabilitation programmes), and is assigned personal medical patient information prior to any interactions; (3) the robot has a medical role, but does not have any specific personal medical information, although it may gain this through interaction. This distinction of roles similarly applies to any other application-specific domain, such as education, social care, or games. The type of information dealt with as the agent fulfils any of these general roles will be different, as will the interaction requirements. However, the overarching aim of companion agent research would be to encompass the functionality required by each of these roles in a single system. As such, the role of the agent, and hence the type of personal information dealt with, is in this paper taken in its most general sense.

The agent's acquisition of personal information through interactions with a human participant requires what may be termed a 'memory system'. This system is responsible for the storage of a wide range of information, from the name of the interacting person, details of their physical appearance (for the purposes of visual identification for example) to aspects of their preferences and habits. The extension of this memory system to interface with specific pre-existing information would also be desirable in certain circumstances, particularly where the purpose of the agent is to provide both companionship and specific support, such as for diabetic patients [3]. In the most general terms, these memory systems must fulfil the roles of encoding, storage and retrieval [4][5], although these are functional requirements, and do not necessarily have to correspond to structural aspects of a computational architecture. Current memory system implementations for artificial agents are generally based on database-like structures, where information is explicitly represented, and may be added to or removed within the defined structure, although an alternative perspective is gaining support (section 3).

Since such agents can be considered as Information Technology (IT) systems, the knowledge that an artificial agent acquires through interaction will be subject to personal data protection legislation. This requirement means that personal data is secure, accessible and updateable by the subject, and erasable if necessary. It is implicitly based on the assumption that the computational structures underlying the memory systems of artificial agents have particular properties and characteristics – namely that they are essentially databases in which information is identifiable and removable if required. However, for artificial agents, if this implicit assumption were to become invalidated, the consequences are undetermined.

It is the aim of this paper to show that should this assumption of explicit information storage be violated, the current means of providing data and privacy protection will become inadequate. Given that distributed forms of control and information encoding

[1] Centre for Robotics and Neural Systems, University of Plymouth, U.K. Email: {paul.baxter, rachel.wood, tony.belpaeme}@plymouth.ac.uk

[2] Fondazione Centro San Raffaele del Monte Tabor, Milan, Italy. Email: {nalin.marco}@hsr.it

are becoming increasingly prevalent, this question is one that requires consideration. This paper is therefore an assessment of data protection issues arising from changes in the computational implementation of memory systems from a legal and ethical perspective: an overview of the social and ethical issues may be found in [6].

The first part of this paper presents a very brief review of the general legal issues involved in the storage and use of patient information by an artificial companion, which illustrates the implicit assumption of a database-like structure for the memory system. An overview of the conceptual move away from strict database structures for the storage of information for artificial agents then demonstrates the need to explore what consequences this has. A set of solutions to this problem is proposed within the current legal framework, but the limitations of these solutions indicate that more appropriate solutions may only be possible with further legislative developments to reflect the disparate range of companion robot implementation methodologies.

2 BACKGROUND

Personal information, its acquisition and processing, is protected by privacy legislation, such as the European Parliament "Directive on the protection of individuals with regard to the processing of personal data and the movement of such data" (1995), [7], and related national laws (such as the U.K. Data Protection Act 1998 [8]). These statutes define the rights of the person to whom the gathered information relates, and includes, for example, the right to know whether and what information is held, and the right to block, update and erase this data (subject to certain constraints). Whilst this clearly covers any medical data, any information acquired by an artificial companion agent in other domains could also be subject to these restrictions, with this potentially also extending to personal preferences, habits and characteristics that are only acquired directly through interaction (rather than drawn from a pre-existing database, such as communication preferences). Thus, in the case of a companion agent designed to learn the personal preferences of a patient, if the patient were to request it, then all the information acquired would have to be removed from the system.

For current human-robot interaction architectures, this requirement is either circumvented or fulfilled by two main factors: (1) at present, human-robot interaction is generally constrained to short-term exchanges that are largely predicated on reactive behaviour on the part of the artificial agent; and (2) any storage of information that does occur in these architectures uses database structures, which means that individual records are explicitly stored, and may be recalled and deleted if required without having an impact on the robot control system used.

Current human-robot interactions have typically been limited to single isolated episodes [1]. As a consequence, memory systems of the type introduced above, which allow the agent to modify its behaviour on the basis of prior interactions, are not well developed. For these artificial companion agents, the problem of data protection does not arise. However, as human-robot interaction develops and extended interactions become more prevalent (including multiple discrete but linked interaction episodes), the necessity of memory to support the desired functionality has become clear, e.g. [9][10]. Where such implementations exist, the use of explicit representation database structures means that deletion of information is a trivial task without functional consequence for the database structure itself.

These developments in companion robotics have been paralleled by new approaches to the design and implementation of general control architectures for artificial agents, including those for memory, based on inspirations from neurobiological empirical evidence. These methods centre on the use of distributed, neural network-like systems in which information is processed in sub-symbolic form. Neural network controllers (and similar) are often not amenable to traditional functional decomposition and the distribution of information within such systems can render them relatively opaque to such analysis. However, network based control systems have been shown to be highly effective mechanisms for robust, dynamic organisation of behaviour in artificial cognitive systems of the type required for companion agents. The use of such architectures entails new perspectives on how information can be processed and stored, leading to a necessity for re-examining the legal and ethical issues incurred; specifically, whether current legislative provisions are adequate in the face of differing implementation methodologies.

3 IMPLEMENTATION METHODOLOGIES

Memory is increasingly being applied to human-robot interaction studies, with previous approaches involving agent behaviour adaptation only within the context of a single interaction episode [1]. The memory systems that are implemented are typically part of cognitive architectures in which memory is regarded as inherently passive, i.e. as a storage system to which information is sent and from which it can be retrieved. For example, the episodic memory system implemented in the SOAR cognitive architecture periodically stores a snapshot of all the contents of the central computation workspace [11]. This snapshot may be subsequently recalled in its entirety and used to bias future processing in the central workspace. Similarly, [12] introduce an explicit representation memory architecture for autobiographical memory with short-term and long-term components. These examples are symptomatic of the general approach that emphasises a division of cognition and related processes into functional modules; memory being in this case a separable function from cognition. Whilst significant advances have been made, it is not clear that such systems are sufficient to underpin the robust, extended and unstructured social interactions with human users required of artificial companion agents.

In contrast with the explicit separation of memory and cognitive processes, there are an increasing number of integrative, biologically derived frameworks that emphasise the reverse. For example, the principle of distributed memory and cognition proposes that cognition, memory and perception share a common substrate [13]. The application of this memory-based principle to artificial cognitive agents, including companion agents, necessitates a view of memory as an active process central to cognition [14][15]. In practical terms a memory-based approach to cognition places the acquisition and handling of distributed memory at the heart of a computational implementation, rather than an information processing mechanism, which, based on the computing metaphor of biological cognition, implies the presence of a passive storage device (memory). The consequences of this perspective on the

legal issues highlighted above (section 2) have not thus far been addressed.

Consider, for instance, a generic sub-symbolic face recognition system. In this case, the recognition system does not explicitly store individual instances of faces (for template matching for example), but rather statistical properties in a distributed manner, which give the benefits of robustness and compactness, e.g. [16]. For this example system, a sub-symbolic network (which may take on a variety of forms and properties) is trained on a series of faces, such that at the end of the training period, the output of the network is 1 (or 'yes') if a newly presented face has been previously seen, or 0 ('no') if the face was not encountered in the training phase. This recognition network thus has the capacity to positively identify a specific set of faces from all possible faces, but without the storage of any individual face images.

Let it be assumed that this face recognition system was trained as part of a study in which volunteer patients consensually provide the face images for the training set. If at some future time one of the patients withdraws consent from the programme, then it may become necessary to remove all personally identifiable information related to the individual from the study. For the face recognition system, the fact that the information contained within the network is in a distributed and statistical form means that there is no formal requirement for this information to be removed according to personal information processing legislation. However, the network may nevertheless be used to provide positive identification of the individual concerned: that this recognition is possible may amount to confirmation of that individual being a participant of the clinical trial at a later date. The problem is thus apparent: while explicit individually identifiable information may have been removed, the capacity to provide explicit personal identification remains. This issue of personal identification, and thus of potential violation of privacy rights, is one that at the present time has been ill-considered, particularly given the specific details of different types of memory system. The question is then how this type of information, in general terms, may be removed such that the requirement for the removal of personally identifiable information under certain circumstances is upheld, within the current legislative framework.

Another potential problem related to the use of information held in sub-symbolic form is that of data privacy in the context of social interactions. Considering that a single artificial companion agent may be interacting with multiple individuals, the issue is how to maintain a separation between the personal information of these individuals such that the confidential data of one person is not revealed by the agent to another individual through interactions. In the case of database-like structures for the storage of information, this may be addressed through the tagging of individual entries as belonging to particular individuals, and blocking the recall of these entries in the presence of others. However, in the case of distributed information, this becomes more problematic, since the overlapping nature of storage results in the same difficulties in identifying individual pieces of information as exposed in the discussion of the face recognition system example. While these problems can be technically addressed to a large extent using explicit meta-knowledge control systems, they merely shift the problem: if personal information or identifiers need to be removed from this meta-control system, the same issues persist.

4 TOWARDS PRINCIPLED SOLUTIONS

In order to propose solutions to the issues raised in the previous section, it is necessary to provide a characterisation of the range of potential memory structures. A continuum is defined that may be used to characterise the type of memory system used (or which may eventually be used), from a structured flat database on one extreme, to an unstructured sub-symbolic network on the other (figure 1). An analysis of some pertinent points along this continuum may be used to place the issues raised in the previous section into context, and allows solutions to the problem of sensitive personal information removal to be explored. It should be noted that one solution that applies in all cases is the deletion of the entire architecture, both control system and memory components, to ensure that no personal information remains in any form. However, this is undesirable as all incidental information would also be lost. This section therefore explores the alternatives, although as will be described, data removal based on the unstructured network perspective on the implementation of memory leads to complete system ablation as the only viable option.

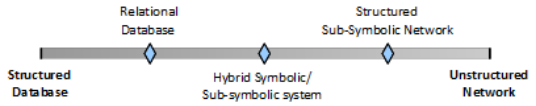

Figure 1. A conceptual continuum of memory system types, with structured databases on one extreme, and unstructured sub-symbolic networks at the other extreme. The relationship between the intermediate points is relational and illustrative.

Five representative points may be identified along this continuum. These points capture some important properties of five different types of implementation methodologies described above (section 3). Indeed, the use of this continuum enables a principled account of potential solutions to be formed, which would not be possible given disparate implementation types.

A flat *structured database* is at one end of the continuum as it allows an explicit representation of data in human readable form in a single structure. This is typically tailored for a specific type of information in a particular format, and so is inflexible to changes in application context, and is a standard type of implementation used for memory systems. However, it is fully compliant with the necessity for data removal, since information is discrete and independent. A similar argument applies to *relational databases*, where information is stored in multiple flat structures (a partially-distributed information representation), which are related to on another in a specific pre-determined manner.

Network structures may also be used to store data, where information is encoded in a distributed fashion in the weights of links between nodes (e.g. artificial neural networks), and is thus not explicitly represented. Combined with database structures, this forms *hybrid symbolic/sub-symbolic systems*. This type of system is particularly relevant for robotics work, where the sub-symbolic aspect is responsible for handling low-level sensory and motor data, and the symbolic aspect responsible for higher level, abstracted information, e.g. [17]. For the database-like components, information may be readily identified and removed without affecting the functionality of the system. However, for the sub-symbolic network component, the selective removal of

certain items of information is problematic, as discussed with the face recognition example above. If the symbolic data related to a specific patient is removed from the database, then an unchanged sub-symbolic network component would no longer result in the patient-specific behaviour that it previously would have done, whilst preserving the general functionality of the architecture. However, in this case, the information (so the positive identification of a face in the above example) would remain in distributed form in the network.

With the progressive replacement of the explicit database structures in a system with sub-symbolic network analogues, *structured sub-symbolic network systems* store information in a distributed manner, but the structure and arrangement of multiple networks allow a limited interpretation of the informational contents (such as specific types of information being restricted to a certain sub-network), as seen in animal-inspired cognitive architectures, e.g. [18]. In this case, there is a significant problem with data removal. Two main approaches may be used to circumvent this. Firstly, the links between the sub-networks may be removed, meaning that only single-modality information can be processed. Whilst this decomposes the information linked to an individual into what is essentially anonymous information, it has two main drawbacks: (1) the system as a whole is impaired to the extent that it is no longer fit for purpose, and (2) modal-specific information still persists in the individual sub-networks, although the context in which it is identifiable to an individual may be removed. The second approach that may be used in certain cases is the replacement of individual sub-networks that may contain the most pertinent information. For example, if the interaction between artificial agent and patient was non-verbal, then it may suffice to replace the face recognition network with a blank analogue. In this case, all of the information contained in this network would be lost (including the ability to recognise other individuals), but other modality information may be preserved (such as recognition of objects and voices, if they are supported by other networks). However, this solution is dependant on the precise forms of interaction that the agent has engaged in with the patient, and will not be a comprehensive removal of all relevant information.

Finally, at the other end of the continuum, in *unstructured networks*, there is no uniquely identifiable structure to the sub-symbolic network, and the information encoded by it is held in a distributed manner – there is no means of determining the contents of the memory system by examining the system itself, only through its behaviour (i.e. its output given an input). In this case, the problem of information removal is at its most acute. There are two possible resolutions to this problem based on the interpretation of data removal informed by symbolic database structures. The first of these is the deletion of the entire system in order to guarantee that the sensitive information has been removed. Whilst this does provide the necessary personal data protection, it also results in the loss of all other information, and any capabilities developed by the system through perhaps extended interactions with other patients. The second option is to periodically produce network state images, in a process akin to software versioning control. If information requires removal, then the system could be reverted to a previous state prior to when this information had been acquired. Whilst this provides an imperfect possible solution to the problem, and even then, it only does so in a relatively limited number of situations. Given that the target domain of these artificial companion agents is extended human-robot interaction, and that each such agent may be reasonably expected to interact with multiple patients, a roll-back to a previous version would necessarily also lose information acquired from interactions with other patients than the subject whose records are to be deleted. Furthermore, given the potentially extended time-scales over which information may need to be deleted (e.g. a request for data removal months or years later), the versioning approach is clearly not ideal. In summary, with an unstructured sub-symbolic system acting as both controller and memory system, there appears to be no satisfactory way of dealing with selective data removal, within the current interpretation of companion robots as IT systems.

Each of the successive points on the continuum illustrate the increasing difficulty that the current approach to personal information management will have with progressively more distributed information representation. Even though each of these systems are not necessarily currently applied to the domain of robotic companion agents, this does not circumvent the need to examine the consequences if they do find eventual application.

5 IMPLICATIONS AND CONSEQUENCES

Some potential resolutions to the problem of managing data stored in sub-symbolic and symbolic/sub-symbolic hybrid systems depend on the perspective adopted as to the locus of responsibility for information management. In the context of medical practice these issues are particularly pertinent, with the maintenance of patient confidentiality largely reliant on the compliance of medical workers. Patient records are potentially available to a large number of employees associated with any given health care facility. Confidentiality is maintained through the adherence of staff to codes of professional conduct. Workers with access to sensitive information are personally responsible for ensuring that they do not breach patient confidentiality. This aspect of the 'duty of care' is backed up by the threat of sanctions against medical staff who fail to observe good practice in data protection: a similar situation to that which exists in other domains.

What happens if we seek to apply the same reasoning to companion robots operating in hospital settings? On this view the companion agent should be treated as having the same responsibility for maintaining patient confidentiality as any other health worker. The obvious problem here is that while medical workers can normally be relied upon to understand their moral obligation to preserve confidentiality, the same assumption clearly cannot be made for an artificial agent [6]. Thus responsibility devolves to the designers of the artificial system, where they would be required to build-in safeguards to ensure that data protection requirements could be met. As described above (section 4) it is not clear how this could be done in the case of sub-symbolic and hybrid systems.

This problem is particularly acute in the context of networks whose structure is, at least in part, derived through interaction (i.e. unstructured networks). Networks in which on-line learning underpins significant aspects of functionality cannot easily be 'wound-back' to 'erase' particular user without some concomitant loss of that functionality.

In the context of these issues it is also interesting to ask how we define the role of 'user' with regard to such systems. On one level, the user of a robot companion can be understood to be the person with whom the robot interacts, such as a patient.

However, the role of user arguably carries with it responsibilities for the behaviour of the system that are not commensurate with the situation of the patient. If the robot is viewed as a medical tool, used to perform various functions in a healthcare setting then it becomes less appropriate to view the patient as the user, with the medical staff (and ultimately the institution for which they work) taking on the role of user instead, as a means of facilitating patient care. This distinction between notions of the user might appear trivial but the issue of who would be responsible for misuse of such equipment clearly is not.

Placing responsibility for the data acquired and stored by a robot companion device in the hands of medical staff still does not answer the question of how such information should be treated and patient confidentiality protected. The problem is complicated by the fact that it is not obvious how we should characterise data stored in a network form. Can such information be regarded as anonymous and thus, like aggregated statistical information, be exempted from data protection legislation? While it seems quite reasonable to consider data in this form to be anonymous this step still does not deal with the problem of identification. The face recognition example, discussed above (section 3), concerns a system that, while not identifying individual faces, can recognise that it has seen a particular face before. It is not clear if such recognition could be said to render an individual identifiable. By extension, if such a system were to respond in a probabilistic rather than binary fashion then it is even less obvious that its outputs could be said to constitute identification of an individual.

It seems that in order to find appropriate practical solutions to data protection problems posed by novel computational architectures it is first necessary to develop more robust notions of what can be treated as data, where the responsibility for data protection lies and what constitutes a proper degree of anonymisation for information held in such systems: i.e. updating the current legislative provisions.

6 CONCLUSION

With the increasing necessity for long-term human-robot interactions, the development of memory systems for artificial companion agents is becoming an important issue, with issues of data protection and privacy consequently coming to the fore. Whilst database-based approaches are currently prevalent, the increasing application of network-based implementations makes these issues more acute, with a need to fully explore the consequences of these different implementation methodologies. This paper has provided the foundations of such a discussion, by proposing a means of characterising memory system structures, and proposes that whilst the current (European) legislative framework may be applicable in a limited manner, it is necessary to re-evaluate the legal status of future artificial companion agents, the data that they will deal with, and the manner in which these data are processed.

ACKNOWLEDGEMENTS

This work is funded by the EU FP7 ALIZ-E project (grant 248116).

REFERENCES

[1] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, 42: 143-166, (2003).

[2] C.D. Kidd, "Designing for Long-Term Human-Robot Interaction and Application to Weight Loss," *PhD Thesis*, MIT, U.S.A. (2008).

[3] O.A. Blanson Henkemans, W.A. Rogers, A.D. Fisk, M.A. Neerincx, J. Lindenberg, and C.A.P.G. van Der Mast, "Usability of an adaptive computer assistant that improves self-care and health literacy of older adults," *Methods of Information in Medicine*, 47: 82-87, (2008).

[4] R.C. Atkinson and R.M. Shiffrin, "Human Memory: a proposed system and its control processes," *The Psychology of Learning and Motivation*, K.W. Spence and J.T. Spence, eds., New York: Academic Press, 89-195, (1968).

[5] N. Derbinsky and N.A. Gorski, "Exploring the Space of Computational Memory Models," *Proceedings of the Remembering Who We Are – Human Memory for Artificial Agents Symposium, AISB 2010*, M. Lim and W.C. Ho, eds., Leicester, U.K., 38-41, (2010).

[6] P.A. Vargas, W.C. Ho, M. Lim, S. Enz, and R. Aylett, "To Forget or Not to Forget : Towards a Roboethical Memory Control," *Proceedings of the New Frontiers in HRI Symposium, AISB'09*, Edinburgh, U.K. (2009).

[7] *European Parliament Directive 95/46/EC: On the protection of individuals with regard to the processing of personal data and on the free movement of such data*, (1995).

[8] *U.K. Data Protection Act 1998*, (1998).

[9] W.C. Ho, K. Dautenhahn, M.Y. Lim, P. a Vargas, R. Aylett, and S. Enz, "An initial memory model for virtual and robot companions supporting migration and long-term interaction," *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, 277-284, (2009).

[10] R. Kadlec and C. Brom, "I can (almost) remember what you are doing : from actions to tasks," *Proceedings of the Remembering Who We Are – Human Memory for Artificial Agents Symposium, AISB 2010*, M. Lim and W.C. Ho, eds., Leicester, U.K., 27-30, (2010).

[11] A.M. Nuxoll and J.E. Laird, "Extending Cognitive Architecture with Episodic Memory," *AAAI Conference on Artificial Intelligence*, R.C. Holte and A. Howe, eds., Vancouver, Canada: AAAI Press, 1560-1565 (2007).

[12] W.C. Ho, K. Dautenhahn, and C. Nehaniv, "Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model," *Connection Science*, 20: 21-65, (2008).

[13] J.M. Fuster, "Cortical Dynamics of Memory," *International Journal of Psychophysiology*, 35: 155-164, (2000).

[14] R. Wood, P. Baxter, and T. Belpaeme, "A developmental perspective on memory-centred cognition for social interaction," *Proceedings of the Tenth International Conference on Epigenetic Robotics*, B. Johansson, E. Sahin, and C. Balkenius, eds., Örenäs Slott, Sweden, pp. 183-184, (2010).

[15] P. Baxter, "Foundations of a Constructivist Memory- Based approach to Cognitive Robotics," *PhD Thesis*, University of Reading, Reading, U.K.(2010).

[16] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, 3: 71-86, (1991).

[17] T.D. Kelley, "Symbolic and Sub-symbolic representations in computational models of human cognition: what can be learned from biology?," *Theory and Psychology*, 13: 847-860, (2003).

[18] J.G. Taylor, M. Hartley, N. Taylor, C. Panchev, and S. Kasderidis, "A hierarchical attention-based neural network architecture, based on human brain guidance, for perception, conceptualisation, action and reasoning," *Image and Vision Computing*, 27: 1641-1657, (2009).

Memory Systems for Cognitive Agents

Uma Ramamurthy[1] and Stan Franklin[2]

Abstract. The several different memory systems in human beings play crucial roles in facilitating human cognition. To build artificial agents that have cognitive capabilities similar to those of human beings, we have to develop these agents based on architectures modelling what we know of human cognition from neuroscience, psychology and cognitive science. In this paper we describe the various memory systems in the LIDA Architecture, which implements Global Workspace Theory. We discuss the interaction between these memory systems, feelings and emotions, and consciousness in the context of cognitive cycles. Finally, we look at our current work on spatial memory in the LIDA model.

1 INTRODUCTION

Human memory seems to come in myriad forms: sensory, procedural, working, declarative, episodic, semantic, long-term memory, long-term working memory and perhaps others. To achieve human-like cognitive capabilities in artificial agents, we have to build them with principles of human cognition and learning. When an autonomous artificial agent [19] is equipped with computational versions of human cognitive features, such as multiple senses, perception, various forms of memory including transient episodic memory and declarative memory, learning, emotions, multiple drives, it is called a cognitive agent [18]. Such cognitive agents promise to be more flexible, more adaptive, more human-like than classical software systems because of their ability to learn, and to deal with novel input and unexpected situations. One way to design and implement cognitive agents is to build them within the constraints of the Global Workspace Theory (GWT) [2], [3], a psychological theory that gives a high-level, abstract account of human consciousness and cognition.

Per Global Workspace Theory, one of the most fundamental functions of consciousness is to provide access among separate sources of information. Effectively, consciousness creates access to various memory systems of a cognitive agent. In the following sections, we will discuss the various human memory systems that play a role in the Learning Intelligent Distribution Agent (LIDA), a model of cognition that implements Global Workspace Theory [7], [37]. The main aims of the LIDA model include understanding how the mind works as well as building smarter and better artificial cognitive systems. The LIDA model, which is both computational and conceptual, includes modules for perception, various types of memories, "consciousness", action selection, deliberation, volition, and several types of learning technologies [23].

[1] St Jude Children's Research Hospital, Memphis, TN 38105, USA. Email: uma.ramamurthy@stjude.org

[2] Dept. of Computer Science and Institute for Intelligent Systems, The University of Memphis, Memphis, TN 38152, USA. Email: franklin@memphis.edu

2 MEMORY SYSTEMS

The memory modules in LIDA are not unique to this model. Other cognitive architectures like SOAR, ACT-R and Clarion for example, have multiple memory systems in them. In LIDA, the approach to memory is more systemic and granular. Let us consider the different memory systems of the LIDA model, short-term to long-term.

Sensory memory holds incoming sensory data in sensory buffers and performs the initial processing. It provides a workspace for integrating the features from which representations of objects and their relations are constructed. There are different sensory memory registers for different senses and probably a separate sensory memory for integrating multimodal information. Sensory memory decays at the fastest rate, measured in tens of milliseconds.

Working memory is the scratchpad of the mind. It holds sensory data, including visual images and inner speech, together with their interpretations. There are separate working memory components associated with the different senses, the visuo-spatial sketchpad and the phonological loop [8], [5]. Its decay rate is in tens of seconds.

Episodic or autobiographical memory is memory for events having features of a particular time and place [10]. This memory system is associative and content-addressable.

An unusual aspect of the LIDA model is its transient episodic memory (TEM), an episodic memory with a decay rate measured in hours. Our hypothesis is that a conscious event is stored in transient episodic memory by a broadcast from a global workspace. A corollary to this

hypothesis says that conscious contents can only be encoded in long-term declarative memory via consolidation from transient episodic memory.

Humans have a variety of long-term memory types that may decay exceedingly slowly. Memory research distinguishes between procedural memory, the memory for motor skills including verbal skills, and declarative memory. Declarative memory (DM) is composed of autobiographical memory and semantic memory, memories of fact or belief typically lacking a particular source with a time and place of acquisition. Declarative memory systems are accessed by means of cues from working memory.

We see a clear distinction between perceptual memory (recognition memory [34]) and sensory memory (similar to Taylor [42]). Our model distinguishes between semantic memory and perceptual associative memory (PAM) and hypothesizes distinct mechanisms for each [20]. PAM memory is a memory for individuals, categories, actions, feelings, events, and their relations. PAM plays the major role in recognition, categorization, and more generally the assignment of interpretations. Upon presentation of features of an incoming stimulus, PAM returns precepts, the beginnings of meaning. We venture that PAM is evolutionarily older than TEM or declarative memory. This further points to the likelihood, though not at all certain, that they have different neural mechanisms. Since the contents of TEM consolidate into DM, which contains semantic memory, these facts suggest the possibility of separate mechanisms for PAM and semantic memory.

Procedural memory in LIDA is a modified and simplified form of Drescher's schema mechanism [14], the scheme net. The scheme net is a directed graph whose nodes are (action) schemes and whose links represent the 'derived from' relation. A scheme consists of an action, together with its context and its result. At the periphery of the scheme net lie empty schemes (schemes with a primitive action, but no context or results), while more complex schemes consisting of actions and action sequences are discovered as one moves inwards.

3 TEM AND DM IN LIDA

Transient episodic and declarative memories have distributed representations in the LIDA model. There is evidence that this is also the case in the nervous system [20]. In this model, these two memories are implemented computationally using a modified version of Kanerva's Sparse Distributed Memory (SDM) architecture [26], [36]. The SDM architecture has several similarities to human memory [26] and provides for "reconstructed memory" in its retrieval process:

- Fast divergence in SDM is equivalent to knowing that one does not know.
- Neither converging nor diverging indicates the tip-of-the-tongue state.
- Rehearsal happens by writing a datum many times to memory. A datum rehearsed well is retrieved with fewer iterations than an item that is stored only once.
- Full and overloaded memories exhibit momentary feelings of familiarity that fade away rapidly.
- Forgetting increases with time because of intervening write operations (interference), as well as decay.

A preconscious percept consisting of a selection of the contents of sensory memory, together with recognitions, categorizations and other interpretations produced in PAM, are stored in working memory. Only the conscious portion of the contents of working memory (actually long-term working memory [16]) is stored in TEM. Information from the same conscious content is used to update PAM, TEM, and procedural memory. The undecayed contents of TEM are consolidated into DM at a later time offline. Retrieval from the content-addressable, associative TEM and DM memories uses recently stored unconscious contents of working memory as cues.

In the next section, we will describe LIDA's cognitive cycle and the role played by the various memory systems in effecting human-like cognitive processing in this artificial agent.

4 LIDA'S COGNITIVE CYCLE

LIDA's processing can be viewed as consisting of a continual iteration of Cognitive Cycles. Each cycle consists of units of understanding, attending and acting. During each cognitive cycle the LIDA agent first makes sense of its current situation as best as it can by updating its representation of its world, both external and internal. By a competitive process, as specified by Global Workspace Theory, it then decides what portion of the represented situation is most in need of attention. Broadcasting this portion, the current contents of consciousness, enables the agent to finally choose an appropriate action which it then executes. Thus, the LIDA cognitive cycle can be subdivided into three phases, the understanding phase, the consciousness phase, and the action selection phase.

Beginning the understanding phase, incoming stimuli activate low-level feature detectors in Sensory Memory. The output is sent to PAM where higher-level feature

detectors feed into more abstract entities such as objects, categories, actions, events, etc. The resulting percept is sent to the Workspace where it cues both Transient Episodic Memory and Declarative Memory producing local associations. These local associations are combined with the percept to generate a current situational model; the agent understands what's going on right now.

Attention Codelets begin the consciousness phase by forming coalitions of selected portions of the current situational model and moving them to the Global Workspace. A competition in the Global Workspace then selects the most salient coalition whose contents become the content of consciousness that is broadcast globally.

In the action selection phase of LIDA's cognitive cycle, possibly relevant action schemes are recruited from Procedural Memory. A copy of each such is instantiated with its variables bound and sent to Action Selection, where it competes to provide the action selected for this cognitive cycle. The selected instantiated scheme triggers Sensory-Motor Memory to produce a suitable algorithm for the execution of the action. Its execution completes the cognitive cycle.

The LIDA model hypothesizes that all human cognitive processing is via a continuing iteration of such cognitive cycles. The unconscious elements of these cycles are proposed to occur asynchronously, with each cognitive cycle taking roughly 200-300 milliseconds. These cycles cascade, that is, several cycles may have different processes running simultaneously in parallel. This cascading must, however respect the serial nature of conscious processes that are necessary to maintain the stable, coherent image of the world [21], [32]. The cascading cycles, which partially overlap, allows a rate of cycling in humans of five to ten cycles per second. There is considerable evidence from cognitive psychology and neuroscience that is consistent with such cognitive cycling in humans [28], [41], [46], [48].

5 FORGETTING IN MEMORY SYSTEMS

Forgetting is a fundamental aspect of memory. Historically, decay [15], [12], [35] and interference [30], [27], [47] have been proposed as two theories on forgetting. Retrieval failures have also been proposed as the possible basis for forgetting – memories never disappear; they just cannot be retrieved [43]. We do not take this view, and build decay into every memory system.

Altmann and Gray [1] have proposed a functional theory of decay, which says that decay and interference are functionally related. If a memory trace decays, it interferes less with future memory traces. This theory states that when an attribute is to be updated frequently in memory, its current value decays to prevent interference with later values; and the decay rate adapts to the rate of memory writes. Wixted [49] has proposed that recently formed memories which have not yet consolidated are vulnerable to interference from mental activity and memory formation.

Memory researchers hypothesize about decay in working memory [25]. While there is debate and controversy over decay in declarative/autobiographical memory, decay in transient episodic memory is a hypothesis that the LIDA model offers.

Decay plays two roles in these cognitive agents: modelling the cognitive processes in memory (assuming the hypothesis that there is decay in human memory systems) and providing the solution to the memory capacity problem of the SDM architecture. Decay is essential in the modified SDM architecture utilized in the LIDA model for Transient Episodic Memory (TEM). Decay ensures that the detailed memory traces of episodes that have occurred in the past few hours are retrievable. Without decay, the SDM architecture will retrieve a high-level, aggregate of all the traces written to that region of the binary space, and not the specific trace that is expected from a TEM. To be able to retrieve details of episodes with cues such as 'where did we park our car this morning?' or 'what did we have for dinner yesterday night?' we hypothesize that decay is required in the modified SDM that will be used as transient episodic memory.

We have tested different types of decay mechanisms in our modified SDM module, including linear decay, exponential decay and inverse sigmoid decay [38]. The inverse sigmoid decay function models the memory hypotheses of decay mechanism by rapid decay of the less rehearsed episodes while episodes which were rehearsed most experienced a very slow decay. Those episodes rehearsed most were retrievable after several decay cycles while all other episodes written fewer times decayed away in the first couple of decay cycles. This high grade filtering ensures that only relevant, important, unique, urgent and highly emotion-based episodes are retained in transient episodic memory, as they come to consciousness many times and are thus written many times to TEM.

6 MEMORY CONSOLIDATION

The Memory Consolidation hypothesis has been discussed and debated from the time it was proposed over a hundred years ago by Müller and Pilzecker [33]. In this hypothesis, it is believed that the hippocampal complex acts as a temporary indexer linking traces in other cortical regions. With repeated reference and retrieval of the memory traces, direct cortico-cortical connections get established and these connections are independent of the hippocampal function [45]. The exact processes and purpose of this mechanism are still unclear. Many believe that consolidation occurs over hours and days, and during

our REM sleep. There is also debate about this process being conscious vs. subconscious. The LIDA model conjectures the need for two episodic memories, transient episodic memory and long term declarative memory. As pointed out in the previous section, the first is needed to recall details of events that would, over time, be wiped out by interference from similar events. In the LIDA model, events reach DM only by consolidation from TEM.

We use the LIDA model to propose a design for memory consolidation. We hypothesize that in cognitive agents based on the LIDA model, the memory traces which have not decayed away from transient episodic memory (TEM) are consolidated into the agent's declarative memory (DM). The contents of every conscious broadcast get stored in TEM. Over time and without rehearsing that information, those memory traces in TEM will decay. On the other hand, when those traces are rehearsed and hence strengthened, they will remain in the TEM. We hypothesize that at regular intervals (perhaps equivalent to human sleep cycles), the cognitive agent transfers the contents of its TEM to its DM.

The two memories – TEM and DM – based on the modified SDM architecture have identical address spaces. The TEM employs a faster inverse sigmoid decay function tuned to the domain in which the cognitive agent lives. The DM has a variable decay rate based on the inverse sigmoid decay function but with parameters different from those of TEM. The decay mechanism in TEM is crucial in ensuring that only memory traces that are significant, relevant and important to the cognitive agent are consolidated to DM. A ball seen under a bush on a morning walk will be encoded in TEM, but is unlikely to be consolidated into DM unless some particular meaning gives it an affective boost, or brought it to consciousness multiple times leading to multiple encodings.

At specific intervals, defined by the parameter 'consolidation time', the consolidation mechanism goes into action. Since the two memories have identical address space, there will be a one-to-one correspondence between their hard locations. The consolidation mechanism transfers the contents of the bit-counters of each hard location in the modified SDM used in the TEM to the corresponding hard location in DM. The parameter 'consolidation time' may be tuned dependent on the domain in which the cognitive agent lives. We hypothesize that this will be in the order of a few hours. The consolidation mechanism may also be triggered by other internal or external states.

7 DISCUSSION

The main goal of our research work in the LIDA model is to understand how minds work, be they human, animal or artificial. In that spirit, the LIDA model has a very granular architecture accounting for various cognitive processes. The cognitive cycle of the LIDA model provides an important tool for fine-grained analyses of cognitive processes. We have several memory systems in the model as described in this paper, based on both psychological, neuroscience and evolutionary evidence as well as on the interactions these memories have with consciousness per Global Workspace Theory [20].

As must be true with any computational/conceptual model of human cognition, the LIDA model is replete with gaps, areas in which it cannot yet offer explanations. One such gap with reference to human memory systems and artificial agents that we are currently working on is *spatial memory*.

In the human brain, two neural systems facilitate encoding of self-location [13]: they are (1) the *place cells* in the hippocampus for encoding unique environments and (2) *grid cells, border cells* and *head-direction cells* in the parahippocampal and entorhinal cortices for mapping positions and directions in all environments. Humans and many animals construct multiple spatial maps, also called cognitive maps [31] generated by these two neural systems. These spatial maps can be extended by adding multiple maps together.

Episodic memory is for the recording the 'what', 'where' and 'when' of events. The 'where' component of episodic memory results in cognitive maps. We hypothesize that a separate memory module/mechanism is needed in the LIDA model to account for such spatial memory/cognitive maps. While considering this memory module, we have to address several issues related to this memory:

- What is the interaction between the spatial memory and the other memory systems in the LIDA model?
- How does consciousness interact with spatial memory?
- What will be the basic representation of a spatial map, and how will it be accessed?
- If complex spatial maps are created from smaller fragments, how are the different fragments linked together and where are they stored?
- How do we represent very large environments in these spatial maps?
- Is there a decay mechanism in spatial memory and if so, what type of decay is to be employed in this memory?

As yet we have only tentative answers to a few of these questions. Taking advantage of this so far relatively rare occurrence of neuroscience providing a mechanism, a primitive spatial map will be represented in a picture like fashion inhabited by land-marks (objects). The representation will denote the size, shape and orientation of the object as well as its position and distance relative to

other landmarks. Each object will also be connected back to its corresponding node in LIDA's PAM, so as to make connections with features and relations of the object that are known to LIDA.

LIDA's spatial memory must interact with its PAM as well as with its two episodic memories so as to provide locations for events [29]. We envision much of this interaction taking place through LIDA's preconscious working memory, but just how is still an open question.

Spatial memory will be a long term memory system. Like most of those in the LIDA model, it will have a network structure with nodes corresponding to spatial maps and links to inclusion (being a subset of). Again as in other forms of long term memory in LIDA, spatial learning will have to be both selectionist (reinforcing existing spatial maps) and instructionalist (creating new spatial maps, or updating the content of existing maps).

Consciousness will play the same role with spatial memory as it does with all other memory systems. We learn that to which we attend, that is, the contents of consciousness.

As we continue work on understanding, designing and implementing spatial memory in the LIDA model, we hope that it will take us one step closer to realizing a more comprehensive and complete model of cognition. Using this model to build artificial agents will enhance our understanding of the interaction amongst these various memory systems, and between these memory systems and consciousness.

REFERENCES

[1] Altmann, E. M. & Gray, W. D. 2002. Forgetting to remember: The functional relationship of decay and interference. Psychological Science, 13(1), 27-33.

[2] Baars, B. J. 1988. A Cognitive Theory of Consciousness. Cambridge.: Cambridge University Press.

[3] Baars, B.J. 1997. In the Theater of Consciousness: The Workspace of the Mind. NY: Oxford University Press.

[4] Baars, B J. 2003. How brain reveals mind: Neural studies support the fundamental role of conscious experience. Journal of Consciousness Studies 10: 100–114.

[5] Baars, Bernard J and Stan Franklin. 2003. How conscious experience and working memory interact. Trends in Cognitive Science 7: 166–172.

[6] Baars, B J, T Ramsoy, and S Laureys. 2003. Brain, conscious experience and the observing self. Trends Neurosci. 26: 671–675.

[7] Baars, Bernard J and Stan Franklin. 2009. Consciousness is Computational: The LIDA model of global workspace theory. International Journal of Machine Consciousness, Vol 1, Issue 1, pp. 23-32.

[8] Baddeley, A. D. 1993. Working memory and conscious awareness. In Theories of memory, ed. A. Collins, S. Gathercole, M. Conway, and P. Morris. Howe: Erlbaum.

[9] Baddeley, A. D. 2000. The episodic buffer: a new component of working memory? Trends in Cognitive Science 4:417-423.

[10] Baddeley, Alan, Martin Conway, and John Aggleton. 2001. Episodic memory. Oxford: Oxford University Press.

[11] Blackmore, Susan. 1999. The meme machine. Oxford: Oxford University Press.

[12] Brown, J. 1958. Some tests of the decay theory of immediate memory. Quarterly Journal of Experimental Psychology,10:12-21.

[13] D Derdikman and E. I. Moser. 2010. A manifold of spatial maps in the brain. Trends in Cognitive Sciences, December 2010, Vol. 14, No. 12, p. 561-569.

[14] Drescher, G. 1991. Made Up Minds: A Constructivist Approach to Artificial Intelligence. Cambridge, MA: MIT Press.

[15] Ebbinghaus, H. 1885/1964. Memory: A contribution to experimental psychology. New York: Dover.

[16] Ericsson, K. A., and W. Kintsch. 1995. Long-term working memory. Psychological Review 102:211-245.

[17] Franklin, S. 1995. Artificial Minds. Cambridge MA: MIT Press.

[18] Franklin, S. 1997. Global Workspace Agents. Journal of Consciousness Studies 4:322-334.

[19] Franklin, S., and A. C. Graesser. 1997. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In Intelligent Agents III. Berlin: Springer Verlag. 21-35.

[20] Franklin, S, B J Baars, U Ramamurthy, and Matthew Ventura. 2005. The role of consciousness in memory. Brains, Minds and Media 1: 1–38.

[21] Franklin, S. 2005. Evolutionary Pressures and a Stable World for Animals and Robots: A Commentary on Merker. Consciousness and Cognition 14:115-118.

[22] Franklin, S. and Ramamurthy U. 2006. Motivations, Values and Emotions: 3 sides of the same coin. Proceedings of the Sixth International Workshop on Epigenetic Robotics, Paris, France, September 2006, Lund University Cognitive Studies, 128; p. 41-48.

[23] Franklin, Stan, Uma Ramamurthy, Sidney K. D'Mello, Lee McCauley, Aregahegn Negatu, Rodrigo Silva L., and Vivek Datla. 2007. LIDA: A Computational Model of Global Workspace Theory and Developmental Learning. AAAI 2007 Fall Symposium - AI and Consciousness: Theoretical Foundations and Current Approaches.

[24] Freeman, W J. 2002. The limbic action-perception cycle controlling goal-directed animal behavior. Neural Networks 3: 2249-2254.

[25] James, Michael. 2002. Modelling Working Memory Decay in Soar. Online Proceedings of the 22nd North American Soar Workshop, Ann Arbor, MI (http://www.eecs.umich.edu/~soar/sitemaker/workshop/22/James-S22.PDF)

[26] Kanerva, P. 1988. Sparse Distributed Memory. Cambridge, MA: The MIT Press.

[27] Keppel, G. and Underwood, B. J. 1962. Proactive inhibition in short-term retention of single items. Journal of Verbal Learning and Verbal Behavior, 1:153-161.

[28] Massimini M, Ferrarelli F, Huber R, Esser Steve K, Singh H, et al., 2005. Breakdown of Cortical Effective Connectivity During Sleep. Science. 309: 2228–2232.

[29] McCall, R., Franklin, S., & Friedlander, D. 2010. Grounded Event-Based and Modal Representations for Objects, Relations, Beliefs, Etc. Paper presented at the FLAIRS-23, Daytona Beach, FL.

[30] McGeoch, J.A. 1932. Forgetting and the law of disuse. Psychological Review, 39:352-370.

[31] McNaughton, B. L., Battaglia, Francesco P., Jensen, O., Moser, Edvard I., & Moser, M.-B. 2006. Path inte-gration and the neural basis of the 'cognitive map'. Nature Reviews Neuroscience, 7, 663–678.

[32] Merker, Bjorn. 2005. The liabilities of mobility: A selection pressure for the transition to consciousness in animal evolution. Consciousness and Cognition 14: 89–114.

[33] Müller G.E. and Pilzecker A. 1900. Z. Psychol. 1, 1.

[34] Nadel, L. 1992. Multiple memory systems: What and why. J. Cogn. Neurosci., 4, 179–188.

[35] Peterson, L. R. and Peterson, M. J. 1959. Short-term retention of individual verbal items. Journal of Exper-imental Psychology, 58:193-198.

[36] Ramamurthy, U., S. D'Mello, and S. Franklin. 2004. Modified Sparse Distributed Memory as Transient Episodic Memory for Cognitive Software Agents. IEEE International Conference on Systems, Man and Cy-bernetics (SMC2004).
[37] Ramamurthy,U., B J Baars, S K D'Mello and S Franklin. 2006. LIDA: A Working Model of Cognition. Proceedings of the 7th International Conference on Cognitive Modelling, p 244-249.
[38] Ramamurthy, Uma, Sidney K. D'Mello, and Stan Franklin. 2006. Realizing Forgetting in a Modified Sparse Distributed Memory System. Proceedings of the 28th Annual Conference of the Cognitive Science Society, p. 1992-1997.
[39] Seth, A K, B J Baars, and D B Edelman. 2005. Criteria for consciousness in humans and other mammals. Consciousness and Cognition 14: 119–139.
[40] Shanahan, M P. 2006. A cognitive architecture that combines internal simulation with a global workspace. Consciousness and Cognition 15: 433-449.
[41] Sigman M, Dehaene S, 2006. Dynamics of the Central Bottleneck: Dual-Task and Task Uncertainty. PLoS Biol. 4.
[42] Taylor, J. G. 1999. The Race for Consciousness. Cambridge, MA: MIT Press.
[43] Tulving, E. 1968. Theoretical issues in free recall. In T.R. Dixon & D.L. Horton (eds.) Verbal Behaviour and General Behaviour Theory, Prentice Hall, Englewood Cliffs, N.J.
[44] Tulving, E. 1985. Memory and consciousness. Canadian Psychology 26:1-12.
[45] Tulving, E. and Craik, I.M. Fergus. 2000. The Oxford Handbook of Memory. Editors. Oxford University Press.
[46] Uchida N, Kepecs A, Mainen Zachary F, 2006. Seeing at a glance, smelling in a whiff: rapid forms of perceptual decision making. Nature Reviews Neuroscience. 7: 485–491.
[47] Waugh, N. C. and Norman, D. A. 1965. Primary Memory. Psychological Review, 72:89-104.
[48] Willis J, Todorov A, 2006. First Impressions: Making Up Your Mind After a 100-Ms Exposure to a Face. Psychological Science. 17: 592–599.
[49] Wixted, John T. 2004. The Psychology and Neuroscience of Forgetting. Annual Rev. Psychol. 2004. 55:235-69.

Implementing a data mining approach to episodic memory modelling for artificial companions

Matthias U. Keysermann[1] and **Alex A. Freitas**[2] and **Patricia A. Vargas**[3]

Abstract. The main goal of this work is to implement and test two different data mining approaches for retrieving and classifying the information within a computational episodic memory model developed for artificial companions. As the information stored in our episodic memory model reflects mainly certain past events we have elaborated an appropriate data structure and created the corresponding event data. Data mining techniques were then implemented for processing the knowledge within the memory model. The data mining task addressed here is classification and further prediction. Two Bayesian classifiers were evaluated by analysing prediction performance in general, comparing the results obtained in specific and more realistic scenarios. This work is a first step towards the full incorporation of data mining techniques to episodic memory modelling for artificial companions. Future work includes the processing of hierarchical data and other machine learning techniques in order to facilitate the creation of more believable artificial companions/robots.

1 Introduction

The idea of living together with robots still seems to be a future vision but becomes more and more realistic as advances are made in building and investigating artificial companions [1]. These artificial companions rely on an intelligent system which enables them to capture data from its surroundings, execute necessary actions and most importantly to interact and communicate with humans.

Such a system could reside on different hardware platforms and can be also transferable to different devices. When used at home the system could run on a stationary computer while at work a laptop might serve as the hardware basis. Moreover, portable electronic devices are widely accepted and therefore it is not unlikely to bring an artificial companion to almost everywhere you go.

Being accompanied by such a system, interaction is a central aspect and it is essential to organise it as appropriate as possible. Having the companion reacting naturally is very important to generate a familiar feeling of interaction for the user. As the human memory is involved in several cognitive tasks, the incorporation of a computational memory model is essential. Modelling characteristics like abstraction, generalisation and forgetting leads not only to more believable companions but could also improve human robot interaction in general [10, 11, 12].

As the topic of memory modelling is huge and has many different subtopics, the limitation to a specific area is necessary. Psychology literature provides an overview of the human memory and tries to structure and subdivide it into different parts [2, 8] (figure 1). In this paper we focus on the *Episodic Memory* which stores information about single events, i.e. episodes.

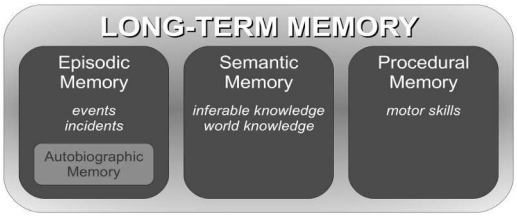

Figure 1. Different parts in the model of the Long-Term Memory.

Although the basic idea about using Bayesian data mining techniques for memory modelling has already been introduced by others [11], this particular approach has neither been implemented nor evaluated before in the given domain of robots as artificial companions. Both the application of the classification methods as being used here and the scenarios tested provide a new perspective on memory modelling for these artificial companions. The data structure explained in [11] has been extended and further specified and sample data for this specific structure has been newly generated.

Section 2 introduces the data structure, which has been elaborated for the usage with memory modelling. This includes a detailed description of the attributes. Section 3 familiarises the reader with the data mining methods. The implemented classification approaches are briefly described by explaining first common characteristics of the implemented classifiers and then mentioning specificities of each of them. A description of the data set as well as an explanation of the evaluated scenarios and the results obtained by applying the classification approaches to the data are given in section 4. Section 5 discusses the work and its outcomes. Sections 6 and 7 conclude the paper and give an outlook on future work.

2 Memory modelling for artificial companions

The episodic memory modelling done here is only a small part of a more complete model of the human memory and thus is based on the work developed by the LIREC project [1]. The LIREC project tries to develop a memory model that includes not only *Long-Term Memory* and *Short-Term Memory* but also many sub-parts. The project addresses as well lower level aspects [4], for instance, capturing and processing sensor data.

[1] Heriot-Watt University, Edinburgh, UK, email: muk7@hw.ac.uk
[2] University of Kent, Canterbury, UK, email: A.A.Freitas@kent.ac.uk
[3] Heriot-Watt University, Edinburgh, UK, email: P.A.Vargas@hw.ac.uk

2.1 Episodic memory structure

Information stored in our *Episodic Memory* reflects mainly certain events happened in the past [2, 8]. This holds as well for the *Autobiographic Memory* except that remembered events in here address personal experiences. In general all these events comprise persons, animals or items involved in a certain activity which happened at a specific location at a particular time. Therefore each event is represented by predefined attributes. All events together are held in a list of single events. The attributes can be nominal or numeric. Every attribute can contain *null* values for some events, indicating that the value of the attribute is unknown or not applicable to those events.

Furthermore the given attributes have a hierarchical structure which makes it possible to specify a more detailed data/information about an event depending on how much is known. This allows the consideration of rough knowledge about a situation. At the same time a lot of detailed information can be given in order to incorporate as much knowledge as possible in the decision making process.

The complete attribute structure is shown in figure 2. The task of identifying and capturing particular attributes and their values in a real world environment is not discussed in this paper as it is not the focus of our work.

The attribute hierarchies are described as follows:

Id: Serves as a unique identifier for each event and makes it possible to distinguish events which are identical in terms of the other values.

Subject: Specifies the subject that actively executes the given task.

Subjects are subdivided by their *type* which can be a *person*, an *animal* or an *item*. The latter one is rather uncommon to be a subject but stays in the hierarchy because of consistency as the same hierarchy is used as well for the object.

Another distinction is made by the *category*. *Persons* are categorised into *male* or *female*, *animals* can fall in the category *dog*, *cat*, *fish*, etc. and for items there are numerous possibilities, e.g. *book*, *newspaper*, *fruit*, etc.

Also a *name* can be given. The name can be the full name for *persons* (e.g. *Peter Simon Smith*), a pet's name (e.g. *Kitty*) or for *items* the particular name, e.g. *The Times* for a newspaper.

Furthermore the *date of birth* can be specified which is applicable for *persons* and *animals*. This might be useful when considering age-related habits. The age doesn't have to be stored but can be computed by additionally considering the date of the event. Even *items* can have a date of birth when thinking of it as a date of production. For *books* or *newspapers* the date of publication would be suitable.

Task: Specifies the task or action which is executed by the subject. The hierarchy allows a distinction between a *main* and a *sub task* where the *sub task* allows a more detailed description of the executed action. This is also useful when there are different ways of how to carry out a task. Though in many cases the *sub task* is not applicable and therefore can't be specified. Both *main* and *sub task* are usually stated as a verb.

Object: Specifies the object which is passively involved in a specified task. Attributes are the same as for subject. The object is very often an *item* but of course can be also an *animal* (e.g. when a person washes a dog) or a *person* (e.g. during a talk).

Place: Specifies the global place at which the event happens. The *country* is useful when modelling cultural, ethical or even religious aspects. By including this information the companion is able to adopt to different cultures and treat them correspondingly. The *city* is important when using the same companion in different cities, i.e. the companion can move place without the need of erasing previously stored knowledge.

Location: Specifies the particular location at which the event happens and is more specific than place.

The *environment* allows to distinguish between different application areas, e.g. *home* and *work*. As behaviours and habits can differ significantly from one environment to another it is reasonable to store this information.

Furthermore *room* and *location* are stored which is important for capturing reoccurring habits. *Room* can also mean areas like the garden or a car park. The *location* describes the particular location inside the room or area, e.g. *table*, *bed*, *desk*, etc.

Date and Time: Specifies the exact *date* and *time* when an event happens. Year, month and day as well as hours, minutes and seconds can be specified. These attributes are of major importance when recognising reoccurring habits.

Also useful is the current day of the week (i.e. *Monday*, *Tuesday*, etc.) which makes it much easier to find weekly patterns. To avoid storing redundant information the day of the week is not saved as another attribute as it can be computed when knowing the exact date.

Privacy: Specifies to whom information about a certain event should be concealed. The structure is the same as for subject although only subjects of type *person* seem to be reasonable.

By having *null* values, in other words, leaving all the attributes unset, means that the information can be disclosed to everybody. When at least the type is given, the interpretation changes, e.g. having the type *person* but category, name and date of birth *null* would conceal the information to every person. By specifying these other attribute values more fine-grained restrictions are possible, e.g. information can be disclosed to everybody except men. With the date of birth also age-related restrictions are possible if interpreting the computed age as a lower limit for revealing information about the event.

Emotions: Specifies the emotions involved both for subject and object. An *emotion classification* like *positive* or *negative* can be given as well as a specific *emotion* like *happy*, *sad*, etc.

These attributes apply for persons, partly for animals but not for items. It might be used by the companion in order to change someone's mood (e.g. to make someone happy) by causing certain events or giving corresponding suggestions.

Furthermore emotions can be stored for the companion itself. Here the focus lies on taking into account the influence of emotions on how information is stored and retrieved. Currently these values are not specified, i.e. are *null*, but are intended for future use.

3 Data mining approach

The data mining task addressed here is classification, where each instance (event) consists of a set of predictor attributes and a class. The goal of a classification algorithm is to predict the class of an instance, based on values of its predictor attributes. A classification

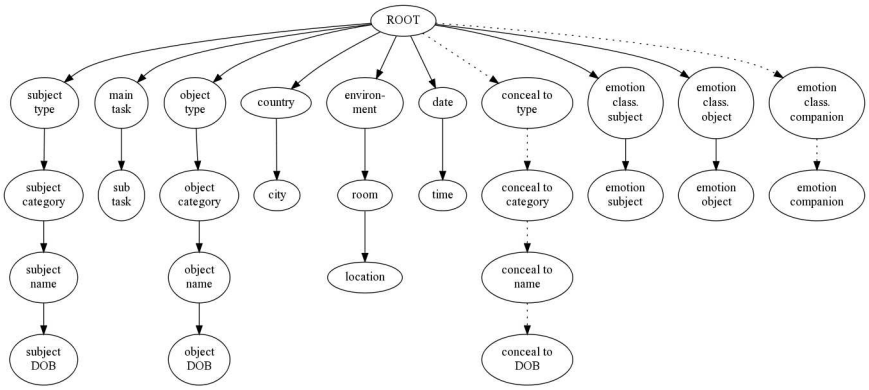

Figure 2. The attributes in their hierarchical structure (*id* not shown here). The attributes connected with dotted lines were not used yet and therefore the corresponding attribute values were always *null*. These attributes are included in this diagram as they will be incorporated into a future model.

problem consists of two phases. First, the algorithm builds a classification model from the training set (where the classes of all instances are known), and then the model is applied to predict the class of instances in the test set, consisting of instances unseen during training.

Two probabilistic approaches have been implemented, namely a simple *Naive Bayes Classifier* and a *Bayesian Network Classifier* [7, 14].

All classifiers build a model by computing and storing lists of probabilities for nominal values or mean and standard deviation in order to represent a normal distribution for numerical values. For nominal values also a list of values encountered during the training phase is kept. None of the classifiers stores all the events used for training.

Nominal values are represented and treated as strings. The classifiers enumerate them internally to map them to the corresponding probabilities. To avoid probabilities of 0 we use a simplified *Laplace estimator*, hence nominal value counts are initialised to 1 instead of 0.

For each classifier we have implemented two approaches to cope with *null* values in predictor attributes during the training phase. Within the first approach, *null* values are ignored, i.e. the instances containing a *null* value for a given predictor attribute are ignored when computing the probabilities required by the *Naive Bayes* or *Bayesian Network Classifier*. In the second approach *null* values are treated as a separate value and probabilities are computed for the *null* value in the same way as for other values. During the test phase the treatment corresponds to the one chosen during the training phase.

Date and time values are converted to their equivalent in seconds and are handled as numeric values which in turn are approximated by the classifiers with a normal distribution. This allows the consideration of the distance between different dates and times which wouldn't be possible if date, month, year, etc. were discretized and treated as being nominal. In order to model weekly routines the classifiers utilise the computed, nominal *day of the week* attribute.

Next we describe each Bayesian classifier implemented.

3.1 Naive Bayes Classifier

The *Naive Bayes Classifier* offers an approach to predict class values completely based on the probability distribution of the attribute values of the data set. Given a set of attribute values the classifier returns the most probable target value. The more training instances there are to learn the classifier, the more reliable and precise are the decisions made. Further information can be found in [7, 14].

The *Naive Bayes Classifier* implemented here does not regard the hierarchical attribute structure as it treats the attributes as being flat and independent of each other. For each nominal attribute that is treated as a class attribute, the posterior probability of each class value can be calculated using two types of probabilities computed during training. One is the class' prior probability and the other is the conditional probabilities of that class given each of the predictor attribute values.

For instance, let A_i denote the attribute i (where A_1 refers to *subject type*, A_2 to *subject category*, A_3 to *subject name*, A_4 to *subject DOB*, A_5 to *main task*, etc.). Then a posterior probability for each value of the attribute A_i (i.e., each class) can be calculated as follows:

$$p(A_{ij}|\theta) = \frac{p(\theta|A_{ij}) \cdot p(A_{ij})}{\sum_{j \in J} p(\theta|A_{ij}) \cdot p(A_{ij})} \quad (1)$$

- J refers to the set of indices obtained by enumerating the values of the attribute A_i.
- A_{ij} refers to attribute A_i having the j-th value of the corresponding set.
- θ refers to the given evidence, i.e., the values of the predictor attributes in the instance being classified.
- $p(\theta|A_{ij}) = \prod_k p(B_k|A_{ij})$ where B_k denotes the value of the k-th predictor attribute in the test instance (or event) being classified, and the product is computed for all values of the index k (i.e. for all predictor attributes individually, ignoring attribute interactions).

Naive Bayes predicts, for each instance in the test set, the class (A_{ij} value) with the highest value of formula (1), among all values of the class attribute A_i.

3.2 Bayesian Network Classifier

As the *Naive Bayes Classifier* also *Bayesian Networks* offer a method for predicting a class value given certain attribute values. In comparison to the former, *Bayesian Networks* can handle conditional dependency amongst the attributes as they do not treat all attributes as being conditionally independent of each other. Therefore they are less constraining than the *Naive Bayes Classifier* in terms of their representational power. A detailed description of *Bayesian Networks* and how they can be used to infer missing attribute values is given in [7, 14].

In our approach every single attribute, i.e. each level of each hierarchy, is considered as being a node in the *Bayesian Network*. For

each class attribute whose value is to be predicted, a network is created with the following structure: each attribute which is at first level (e.g. *subject type*, *main task*, *object type*, etc.) and is a predictor attribute (i.e. different from the class attribute) has just one parent node, namely the class attribute. Every other attribute (except the class attribute) has exactly two parents, namely the class attribute and the corresponding parent (hierarchical order shown in figure 2). The class attribute itself has no parents.

Consider for example that *country* has to be predicted. First level attributes like *subject type*, *main task*, etc. (except *country*) have only *country* as a parent node. *subject category*, which is situated at the second level, has two parent nodes, namely *country* and its hierarchical parent attribute *subject type*. Similarly other non-first level attributes are incorporated into the network structure. Also *city* is treated as having two parent nodes in the *Bayesian Network*, although both parent attribute and class attribute are the same here (which is *country*).

Table 1. Percentage of *null* values for each attribute (shown values are rounded).

attribute name	*null* values	attribute name	*null* values
subject type	0.0%	date	0.0%
subject category	0.0%	time	0.0%
subject name	0.0%	conceal to type	100.0%
subject DOB	0.0%	conceal to category	100.0%
main task	4.3%	conceal to name	100.0%
sub task	68.7%	conceal to DOB	100.0%
object type	7.5%	emotion class.	
object category	7.5%	subject	0.0%
object name	7.5%	emotion subject	0.0%
object DOB	64.5%	emotion class.	
country	0.0%	object	64.5%
city	0.0%	emotion object	64.5%
environment	0.0%	emotion class.	
room	0.0%	companion	100%
location	0.0%	emotion companion	100%

4 Results

This section reports results using the Bayesian classifiers described in section 3 to predict the classes of all nominal attributes in the data set described in section 4.1. Two broad types of experiments were performed, one where the data was randomly partitioned into training and test sets (section 4.2) and another where the data was partitioned into training and test sets according to the temporal order of the events (section 4.3). In each experiment, each Bayesian classification algorithm is used to solve as many classification problems as the number of nominal attributes. In each of such problems, a different nominal attribute is treated as the class attribute and all the other available attributes are used as predictor attributes.

4.1 Data set

No suitable real-world event data was available to directly work on. Neither pre-made events existed nor similar information has been captured which could be used to build events from. Therefore the manual creating of events was needed. Of course these events should match as best as possible with real data and should reflect everyday routines, weekly habits, etc. These happen usually at the same day of the week, roughly at the same time and often at the same location or at least close together, i.e. reoccurring events have many things in common and differ only in small details.

In order to achieve that, the creation should be based on regular routines but happen to some extend in a partially random way. While most of the attribute values stay fixed for one kind of event, mainly for the hierarchies subject, task and place, the exact *time* differs slightly as well as the *location* or the particular *object name*. Emotions can be related e.g. to the current task or to the *environment* – always depending on the event at hand.

These partly-fixed event patterns were necessary. A completely random creation would not suit our purposes as it would hardly generate patterns. Thus it would not make it possible for a classifier to detect any patterns and make predictions based upon them.

In order to obtain a data set, first, domain values have been created for each attribute. The full list of nominal values (including numerical date of birth values when appropriate) is shown in table 2.

By using these values, events have been generated for two fictional characters, namely *George* from Edinburgh and *Amy* from Glasgow. These events have been generated day by day according to day templates. These templates describe a series of events happening throughout a single day. Usually only some attribute values are fixed whereas other values are not clearly defined. In the latter case the corresponding attributes can take up any appropriate value in terms of the corresponding hierarchy value (table 2). For instance if the object type is specified as *item* and the object category is *fruit* then the object name can be either *apple*, *banana* or *orange*.

For each event description a time range indicates the potential period of time when the event can happen. It happens only once at a certain moment within this period. An associated emotion refers to the emotion of the subject, the object's emotion (if applicable) was randomly chosen. This was done as well for all other values which are not explicitly stated in the corresponding templates. The only exceptions are the four privacy attributes and the two companion's emotion attributes of which all the values are always *null*.

For each character two day templates exist, one for working days and one for weekend days. For *George* a working day template describes 19 events, a weekend day template 13 events. For *Amy* the respective counts are 15 and 9.

The time period for which the events have been generated spans over two months, namely May and June 2010, which corresponds to 43 working days and 18 weekend days. Hence, 61 days of events for each character have been generated. Each event was considered as an instance by the Bayesian classification algorithms, so the data set consists of 1858 instances.

4.2 Experiments with a random data partitioning

The first set of experiments involved three different ratios of random allocation of the instances into the training and test sets as follows:

- 90% – 10% (1672 events in training set, 186 events in test set)
- 50% – 50% (929 events in training set, 929 events in test set)
- 10% – 90% (185 events in training set, 1673 events in test set)

For a given allocation the classification performance, i.e. the ratio of correct classifications amongst all predictions made, arises from the predictions made over all events in the test set. Predictions were made for every nominal attribute, the obtained results were then averaged. Excluded here were the attributes contained in the privacy hierarchy and the emotion attributes for the companion (as these attributes are always *null*). These attributes were not used as a predictor attribute either and thus they did not influence the computation of the likelihood.

Table 2. Attribute values used for the data set creation.

attribute name	attribute values	parent value
subject type	person, item	-
subject/object category	male, female	person
subject/object category	fruit, snack, beverage, liquor, newspaper, book, magazine, letter, article	item
subject/object name	Frederic, George, James, John	male
subject/object name	Amy, Betty, Helen, Kate	female
subject/object name	apple, banana, orange	fruit
subject/object name	cookie, brownie, muffin	snack
subject/object name	water, juice, coffee, tea	beverage
subject/object name	beer, wine	liquor
subject/object name	The Guardian, The Times, The Sun	newspaper
subject/object name	The Shining, Harry Potter and the Philosopher's Stone, Dr. Jekyll and Mr. Hyde	book
subject/object name	The Rolling Stone, The National Geographic, Time Magazine	magazine
subject/object name	personal letter, business letter	letter
subject/object name	summary, report, short story	article
subject/object DOB	21/04/1988	Frederic
subject/object DOB	15/01/1979	George
subject/object DOB	03/11/1957	James
subject/object DOB	25/10/1972	John
subject/object DOB	02/08/1991	Amy
subject/object DOB	11/02/1981	Betty
subject/object DOB	28/12/1977	Helen
subject/object DOB	14/05/1963	Kate
main task	eat, drink, read, write, speak, wash	-
sub task	handwrite, type	write
sub task	chat, talk, discuss	speak
country	United Kingdom	-
city	Edinburgh, Glasgow	United Kingdom
environment	home, work	-
room	living room, kitchen, bathroom, bedroom	home
room	office, cafeteria, meeting room	work
location	table, sofa, shelf	living room
location	cooker, fridge	kitchen
location	toilet, sink, shower	bathroom
location	bed, wardrobe, chest of drawers	bedroom
location	desk, printer, photocopier, bookshelf	office
location	queue, counter	cafeteria
location	chair, screen	meeting room
emotion class. subject/object	positive, negative	-
emotion subject/object	happy, excited, relaxed	positive
emotion subject/object	sad, bored, stressed	negative

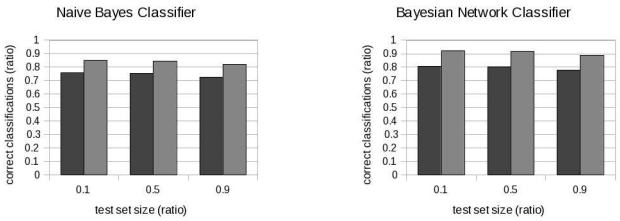

Figure 3. Averaged classification performance for different test set sizes averaged over all nominal class attributes. Shown in dark grey are the results when *null* values were ignored, results when *null* was treated as a separate value are shown in light grey.

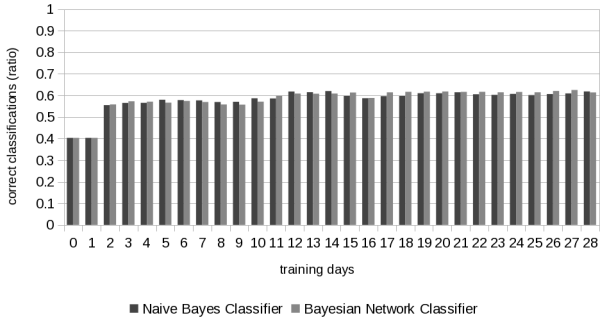

Figure 4. Averaged classification performance for different numbers of training days for a test set size of 7 days whereby only the *date* and *time* were known.

For a given partition ratio the allocation happens randomly, meaning that a new allocation can produce different results. Therefore for each classification process 10 runs were executed, having a new allocation in each run. Finally the results were averaged over the 10 runs.

The treatment of null values can have a major impact on classification performance. For all implemented classifiers this option has been tested by evaluating the performance when completely ignoring *null* values as well as considering them as being another attribute value. Averaged results for different test set sizes are shown in figure 3. As can be observed in this figure, the ratios of events correctly classified by the *Naive Bayes Classifier* and the *Bayesian Network Classifier* are greater when *null* values are treated as separate values.

4.3 Experiments with a temporal data partitioning

In this scenario the classifiers use the knowledge learned in different days of training to predict what happens next. Here a temporal partitioning of the data set is used, different from the ones described in section 4.2. Events are grouped according to the value of the *date* attribute resulting in one event set for every single day. A certain number of successive days, i.e. events of the corresponding sets, is used to build the probabilistic model while for the following days events have to be predicted, thus events happened during the following days form the test set.

For the prediction each event of the test set is taken to obtain its *date* and *time*. Then the classifiers are given that information and have to predict all unknown values by making use of a consecutive classification method. This means that the classifiers predict all nominal attribute values that are set to *null*. Therefore the classifiers calculate posterior probabilities for each nominal attribute being set to *null*. Amongst all the computed posterior probabilities the maximum is chosen and the corresponding attribute value is set for the corresponding attribute. This process is repeated until there are no *null* values left in the nominal attributes of the given event. As this classification method doesn't work when *null* values are treated as another value, they had to be ignored by all classifiers.

Of course the *date* can confuse the classifiers as its mean is always in the period of the training events and probabilities become smaller the further an event is in the future. But there is still the *day of the week* attribute which can be reliably used to recognise weekly patterns.

To evaluate the predictions the ratio of correctly classified attribute values has been determined by comparing the predicted value with the actual value from the sample data. Finally results have been averaged to obtain the classification performance over the complete test set.

In order to figure out how many days of training are required and how the number of training days can improve classification performance, 29 different evaluations have been done starting with no training at all (0 days for training) up to four weeks, or 28 days (856 events), constantly increasing the number of training days.

The particular number of events contained in these training sets depends on the how many working days and how many weekend days are included in this period. The range of the days used in the training set always starts with the very first day present in the data set. The days following the last day of the training set are taken into account for the test set. The number of days for testing was first set to 7 which corresponds to 214 events, then to 1 which corresponds to 34 events if it was a working day or 22 events if it was a weekend day.

Only one single run is required for each number of training days, because training and test sets are generated deterministically.

For no or only one day of training the classification performance of all classifiers is rather low with about 40% (figure 4). Building the probabilistic model over one day more leads to a significant increase in the amount of correct predictions – for two training days about 55% of the attribute values are classified correctly. The first two days in the sample data are Saturday and Sunday. Although no working days have been trained so far, after two days both weekend days have been trained once, allowing to match both Saturday and Sunday by the *day of the week* attribute.

Surprisingly no other significant improvement can be spotted when the classification models include knowledge about working days. After 9 days of training the classification performance increases slightly until after 3 more days it reaches a little bit more than 60%. For more training days the amount of correct predictions remains roughly the same without any further significant improvements. All the classifiers handle this scenario almost equally well but cannot achieve really high prediction rates.

It is important to mention that only very little information was given and in practise it can be expected that more attribute values are either known or just given. The *country* and the *city* can be easily detected via *GPS* (*Global Positioning System*). If this information is not available, at least it is possible to manually restrict the predictions to a specific place and set these values correspondingly.

Another evaluation has been carried out whereby apart from *date* and *time* also the place attributes were known. The results (figure 5) are very similar to the ones previously discussed. Generally the prediction rate for all training days is a little bit higher than it was before,

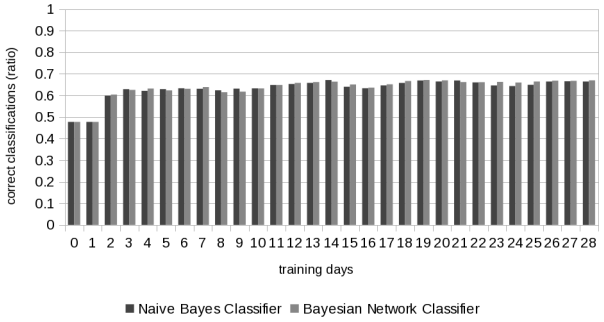

Figure 5. Averaged classification performance for different numbers of training days for a test set size of 7 days whereby *date*, *time*, *country* and *city* were known.

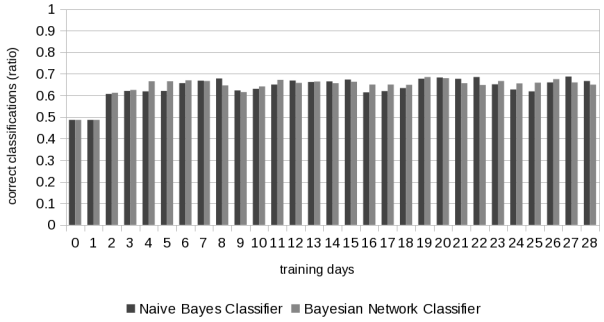

Figure 6. Averaged classification performance for different numbers of training days for a test set size of 1 day whereby *date*, *time*, *country* and *city* were known.

ranging from almost 50% to more than 65%. But again the only significant change in terms of prediction performance happens after two days of training.

A possible reason for lowered classification rates could have been the days being further in the future. This way the prediction of later days worsened the results. Therefore the test set size was set to only one day and results were determined again.

Also for testing with only one day, the results are not significantly better (figure 6). For all classifiers the classification performance varies slightly more than it does in the other two evaluations but never exceeds 70%. The prediction rates after 9 days and 16 days can be considered as local minima. In both cases this is after another weekend has just been trained and the proportion of trained weekend days is higher.

It can be concluded that a short period of training is enough. The performance can't be increased much more by longer training over more days. Here it has to be considered that the sample data mainly contained daily routines, separated by working days and weekends, and that the results can be quite different when patterns in the data change over time.

It should be noticed that the classiers were not able to predict *null* as a value due to the chosen treatment of *null* values. Hence, *null* values contained in the events could not be predicted correctly which in turn limited the achievable classication performance.

Overall no really high prediction rate can be expected in this scenario. Nonetheless, the classications made here are still a difcult task for the high amount of missing knowledge at the beginning of each consecutive classication task.

5 Discussion

Most often the performances of both classifiers were rather similar and none of them can be clearly declared to be the best approach.

The *Bayesian Network* achieved the highest average prediction rates but was closely followed by the *Naive Bayes Classifier*. The two approaches performed better on average when *null* values were taken into account.

The difference made by using this option was significant for the attribute *sub task* and the emotion attributes of the object. These attributes frequently take *null* as their value while the amount of *null* values in the sample data for the other attributes was very low (except for the privacy and companion's emotions attributes). In reality it is likely that some attribute values are unknown or cannot be obtained, which means that the event data contains more unset attributes.

No judgement about difference in performance of the two approaches can be made by comparing the results of the future predictions. But at least they show that only a small amount of events used for training is required to build an appropriate probabilistic model. It has to be mentioned that the patterns represented in these models arise from sample data which only contain rather simple patterns. Events containing more complex weekly or monthly routines may require more training.

6 Conclusions

Two different classifiers have been implemented. They are both probabilistic approaches but treat and represent the data in different ways. In different evaluation scenarios both approaches approved as being useful with good results and overall none of the two approaches performed really poorly.

Though more complex implementations can be thought of and better results are certainly possible when classifiers are more specialised and more advanced techniques are incorporated.

Nonetheless, the results discussed here show that data mining techniques are well-applicable to memory modelling and can produce useful predictions when dealing with discrete attributes. The storage of information about several different situations is possible and knowledge extraction can be performed with the techniques used here.

The idea of artificial companions seems probable but still the step from capturing raw sensor data to processed discrete attribute values has to be further developed.

The interaction with robots would benefit from incorporating artificial memory models that mimic our own [5, 6]. By making robots aware of the activities that happens around them over a longer period of time and letting them make assumptions about the situation they are currently involved in would make robots more believable and better suited for living together with human beings.

7 Future work

This work is a first step into a topic which deserves – due to its extend – much more investigation in the future. It has many different aspects and subtopics and therefore offers lots of potential for extensions in numerous ways.

Apart from creating sample data which contains more complex patterns, the privacy attributes can be filled with values and can be included when building and using prediction models. Furthermore

the classifiers can treat these attributes in a special way, e.g. apply filtering or other sorts of preprocessing.

The companion's emotion attributes have been ignored so far. As described in psychology literature [2, 8] recall and the retrieval process in general are dependent on the mood the corresponding individual is currently in. Therefore emotions of the companion could be taken into account when retrieving knowledge by the classifiers. Furthermore environment-dependency could be included as the human memory is influenced by surrounding conditions.

More complex models can be built by taking into account further aspects of the human memory. The process of forgetting has not been included in the implementations done here. The explicit modelling of forgetting theories like *trace-decay*, *interference* and *repression* would make the forgetting process more transparent and thus easier to evaluate.

In consideration of the increasing number of artificially controlled systems which are around us in our everyday lives, the consideration of ethical issues becomes necessary. Also in computational memory modelling the use of ethical rules is reasonable. Theories like the *deonthological*, *consequentialist* and *virtue-based* theory [13] could be implemented by using rule-based and state-based techniques as well as prediction schemes as described by [10, 12].

Apart from probabilistic models several other machine learning approaches could be tested on the attribute structure developed for this project. Other methods like *Artificial Neural Networks*, *Support Vector Machines* and *Decision Trees* have been applied to hierarchical classification in other domains [3, 9] and could be transferred to the memory modelling for artificial companions.

ACKNOWLEDGEMENTS

This work was partially supported by the European Community (EC) and is currently funded by the EU FP7 ICT-215554 project LIREC (LIving with Robots and Interactive Companions). The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein. We would like to thank the referees for their comments which helped improve this paper.

REFERENCES

[1] LIREC (LIving with Robots and Interactive Companions). http://lirec.eu/.
[2] Alan Baddeley, *Your Memory: A User's Guide*, Carlton Books, new illustrated edn., 2004.
[3] Alex A. Freitas and André de Carvalho, 'A tutorial on hierarchical classification with applications in bioinformatics', in *Research and Trends in Data Mining Technologies and Applications*, ed., D. Taniar, 175–208, Idea Group, (2007).
[4] Wan Ching Ho, Mei Yii Lim, Patricia A. Vargas, Sibylle Enz, Kerstin Dautenhahn, and Ruth Aylett, 'An initial memory model for virtual and robot companions supporting migration and long-term interaction', in *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2009), Toyama, Japan*, pp. 277–284, (2009).
[5] Mei Yii Lim, Ruth Aylett, Wan Ching Ho, Joao Dias, and Patricia A. Vargas, 'Human-like memory retrieval mechanisms for social companions', in *AAMAS 2011*, Taipei, Taiwan, (2011).
[6] Mei Yii Lim, Ruth Aylett, Wan Ching Ho, Sibylle Enz, and Patricia A. Vargas, 'Forgetting through generalisation - a companion with selective memory', in *AAMAS 2011*, Taipei, Taiwan, (2011).
[7] Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
[8] Alan J. Parkin, *Essential Cognitive Psychology*, Psychology Press, 2006.
[9] Carlos N. Silla and Alex A. Freitas, 'A survey of hierarchical classification across different application domains', *Data Mining and Knowledge Discovery*, **22**(1–2), 31–72, (April 2010).
[10] Patricia A. Vargas, Ylva Fernaeus, Mei Lim, Sibylle Enz, Wan Ho, Mattias Jacobsson, and Ruth Aylett, 'Advocating an ethical memory model for artificial companions from a human-centred perspective', *AI & Society*, **Online First**, 1–9, (2011).
[11] Patricia A. Vargas, Alex A. Freitas, Mei Yii Lim, Wan Ching Ho, Sibylle Enz, and Ruth Aylett, 'Forgetting and generalisation in a memory model for robot companions: a data mining approach', in *Proceedings of the AISB 2010, Leicester, UK*, (2010).
[12] Patricia A. Vargas, Wan Ching Ho, Mei Yii Lim, Sibylle Enz, and Ruth Aylett, 'To forget or not to forget: Towards a roboethical memory control', in *Killer Robots or Friendly Fridges: the Social Understanding of Artificial Intelligence, AISB 2009, Edinburgh, UK*, pp. 18–23, (2009).
[13] Wendell Wallach and Colin Allen, *Moral Machines: Teaching Robots Right from Wrong*, Oxford University Press, 2009.
[14] Ian H. Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, second edn., 2005.

Proceedings of AISB `11: Human Memory for Artificial Agents
Dimitar Kazakov and George Tsoulas (eds.)
ISBN 978-1-908187-04-8

Published by the Society for the Study of Artificial Intelligence and the Simulation of Behaviour
Printed by the University of York, York, UK