What am I? – Complementing a robot's task-solving capabilities with a mental model using a cognitive architecture

Thomas Sievers[1,*], Nele Russwinkel[1] and Ralf Möller[2]

[1]Institute of Information Systems, University of Lübeck, Ratzeburger Allee 160, 23562 Lübeck, Germany
[2]CHAI-Institut, Universität Hamburg, 20354 Hamburg, Germany

Abstract

One way to improve Human-Robot Interaction (HRI) and increase trust, acceptance and mutual understanding is to make the behavior of a social robot more comprehensible and understandable for humans. This is particularly important if humans and machines are to work together as partners. To be able to do this, both must have the same basic understanding of the task and the current situation. We created a model within a cognitive architecture connected to the robot. The cognitive model processed relevant conversational data during a dialog with a human to create a mental model of the situation. The dialog parts of the robot were generated with a Large Language Model (LLM) from OpenAI using suitable prompts. An ACT-R model evaluated the data received by the robot according to predefined criteria – in our example application, hierarchical relationships were established and remembered – and provided feedback to the LLM via the application for prompt augmentation with the purpose of adapting or fine-tuning the request. Initial tests indicated that this approach may have advantages for dialogic tasks and can compensate for weaknesses in terms of a deeper understanding or "blind spots" on the part of the LLM.

Keywords

cognitive architecture, human-robot interaction, ACT-R, mental model, ChatGPT

1. Introduction

AI technologies are finding their way into our daily lives ever more quickly and to an ever greater extent. As a result, there is a greater need for AI systems that work more or less *"at eye level"* – on equal terms – with humans. Robots that interact with humans and solve tasks with a human partner must have some kind of model of the world, the situation, the task to be solved and the person they are interacting with. These robots require cognitive capabilities that may need to go beyond Cartesian mind-body dualism in order for the humans cooperating with these artificial cognitive agents to build a growing level of personal trust and mutual accountability [1].

Robots today already have impressive abilities in terms of navigation, interaction with objects and social interaction [2]. Kambhampati uses the term *human-aware AI* systems to describe methods that pay more attention to the aspects of intelligence that enable successful collaboration between people, including modeling the mental states of humans in the loop [3]. An inclusion of mental models also makes sense in terms of *human-centered AI* (HCAI), which aims to create AI systems that enhance and complement human capabilities rather than replace them [4]. The aim is to move from a mindset that focuses on algorithms to a human-centered perspective that also improves trust in and acceptance of social robots in Human-Robot Interaction (HRI).

In our opinion, cognitive architectures, with their ability to create mental models based on human cognitive abilities, can be used to provide robotic applications with a "human component" [5]. A

ALTRUIST, BAILAR, SCRITA, WARN 2024: Workshop on sociAL roboTs for peRsonalized, continUous and adaptIve aSsisTance, Workshop on Behavior Adaptation and Learning for Assistive Robotics, Workshop on Trust, Acceptance and Social Cues in Human-Robot Interaction, and Workshop on Weighing the benefits of Autonomous Robot persoNalisation. August 26, 2024, Pasadena, USA

[*] Corresponding author.

t.sievers@uni-luebeck.de (T. Sievers); nele.russwinkel@uni-luebeck.de (N. Russwinkel); ralf.moeller@uni-hamburg.de (R. Möller)
0000-0002-8675-0122 (T. Sievers); 0000-0003-2606-9690 (N. Russwinkel); 0000-0002-1174-3323 (R. Möller)

combination of robot sensing and data processing with such an architecture offers the possibility to use real-world data from the robot in mental models. Creating such models enables the robot to better understand the mindset of a human partner or to act in a way that is perceived as more natural by humans.

Cognitive architectures refer both to a theory about the structure of the human mind and to a computational realization of such a theory. Their formalized models can be used to flexibly react to actions of the human collaboration partner and to develop situation understanding for adequate reactions. Well-known and successfully used cognitive architectures are ACT-R (Adaptive Control of Thought - Rational) and SOAR [6]. ACT-R has an instance-based learning mechanism that enables intuitive decision-making and exploration of the cognitive processes and representations that underlie human behavior [7]. Pipitone et. al., for example, used ACT-R to implement a method for a robotic self-recognition by inner speech [8].

We thought of on an idea for using an ACT-R model to control and manage the dialog parts of the robot in a guessing game called "What am I?". With the use of large language models (LLMs) from OpenAI's Generative Pretrained Transformer (GPT, commonly known as ChatGPT) [9] we generated what the robot was saying via ChatGPT, created a mental model of essential parts of the dialog in parallel to control the dialog by feedback of the results and outputs of the model, which flowed into the generation of the ChatGPT prompt for the next robot utterance.

In the following, we explain our ideas on this topic, provide an insight into the ongoing work and summarize our initial findings.

2. Methodology

When using large language models, the challenge remains to generate texts that fulfill complex constraints. As an example to overcome this challenge, Zhang et al. proposed the application of lexical constraints in such language models using tractable probabilistic models (TPMs) [10].

Our approach to address the constraint problem with ChatGPT was to use a cognitive architecture to control the LLM's utterances. Since ChatGPT does not have a dynamic cognitive model of the human conversation partner and therefore quickly reaches its limits in cognitive processes, we used an ACT-R model that tracks the course of the conversation and intervenes if necessary.

We have only just started working on this. However, since it is possible to integrate an ACT-R model with bidirectional communication into a robot application as described by Sievers et. al. [11] and thus also establish the connection to an LLM, we hypothesize the following:

Hypothesis It is possible to control the output of an LLM via prompts that are influenced by a cognitive model.

This approach should result in an advantage in terms of a more human-like chain of thought and thus a more human-like behavior – in our test scenario so far only in relation to the robot's utterances – followed by a better understanding and thus also greater acceptance by humans.

2.1. ACT-R

The basic mechanism of ACT-R consists of the main components *modules*, *buffers* and *pattern matcher* [12]. There are two types of modules: Perceptual-motor modules forming the interface with the real world (*motor module* and *visual module*), and the memory modules comprising *declarative memory* consisting of facts and *procedural memory* consisting of productions. Productions represent knowledge about how something should be done. Figure 1 gives an overview of the main components.

ACT-R accesses its modules (with the exception of the procedural memory) via special buffers. The buffers form the interface to this module. The buffer content represents the state of ACT-R over time. The pattern matcher attempts to find a production that corresponds to the current state of the buffers. Only one production can be executed at a time. Productions can modify the buffers during execution and thus change the state of the system. Cognition is therefore represented in ACT-R as a sequence of production firings.

In our approach, we did not use the visual and motor modules to provide input to the system. Our application wrote the relevant data from the LLM utterances directly into the declarative memory of the ACT-R model. In this way, new memories were created that could be accessed later. The results of the model's considerations were fed back to the robot application via the goal buffer.

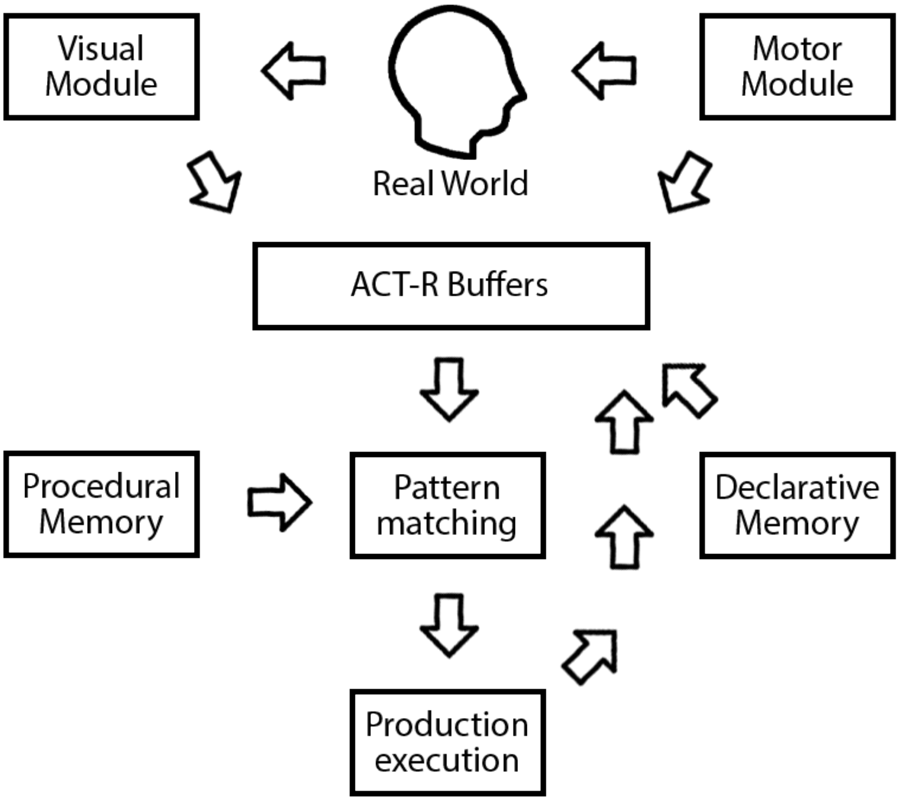


Figure 1: ACT-R modules, buffers and pattern matcher

2.2. Test scenario

The scenario for our test case was a guessing game called "What am I?". A participant – in this case the human – thought of a profession that the robot has to guess. The robot asked questions about the field of activity, etc., and the human could only answer *yes* or *no* to these questions. We used GPT-4o to create the conversational parts of a social robot. The LLM was instructed via system prompts to understand the rules of the game and everything we needed it to output.

For our studies we used the humanoid social robot Pepper [13], which is optimized for human interaction. In a dialog with humans, the robot application we created forwarded the utterances of the human dialog partner as input to the OpenAI API, which returned ChatGPT's answer as response. With each API call, the entire dialog was transferred to the GPT model. This allowed the model to constantly 'remember' what was previously said and refer to it as the dialog progressed. The text returned by the API was forwarded to the robot's voice and tablet output. The OpenAI API provides various hyperparameters that can be used to control the model behavior during an API call. To obtain consistent responses and exclude any randomness as far as possible we set the value for *temperature* to 0.

Figure 2 shows the setup for the bidirectional connection between ACT-R and the client application on the robot. The dispatcher as a part of the ACT-R framework acts as a kind of server and is necessary to establish a remote connection between the robot application as a client and ACT-R.

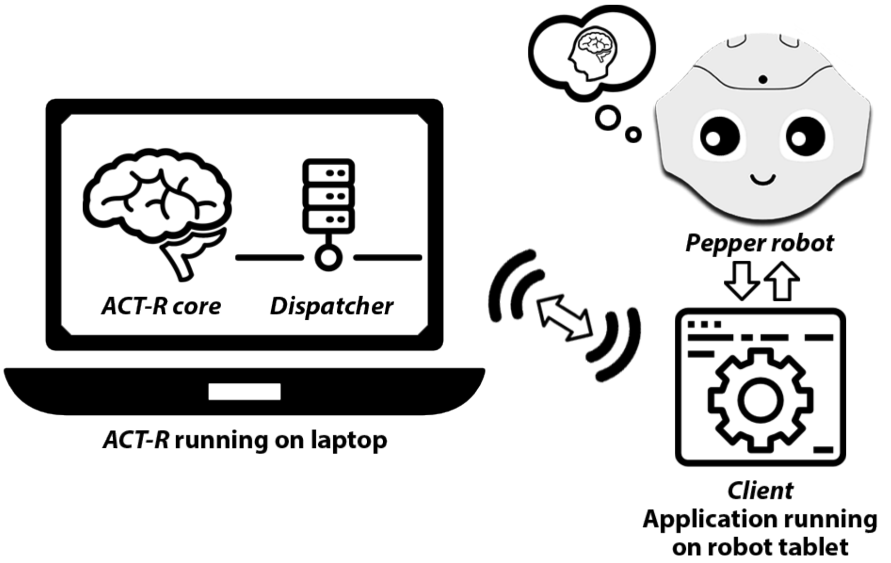


Figure 2: Connection between ACT-R / Dispatcher and the robot respectively the client application

In order to determine possible differences in guessing behavior, we compared game runs in which questions and responses were generated exclusively by ChatGPT itself with runs in which an ACT-R model was integrated that picks up and processes job-specific terms from the ChatGPT utterances. These job-specific terms must also be generated dynamically by ChatGPT.

2.3. Prompting the LLM

We used prompts for the system role to instruct GPT-4o to execute the tasks as a completion task. Zero-shot prompting [14] was used for this tasks. The system prompt for instructing the LLM consisted of the explanations on how to play the guessing game. In addition, the LLM was instructed to output keywords that characterize the profession, separated by commas in square brackets, for example [creative, computer, designer]. These job-specific keywords were transferred from the robot application to the ACT-R model and stored as a chunk in the declarative memory as sketched in Figure 3.

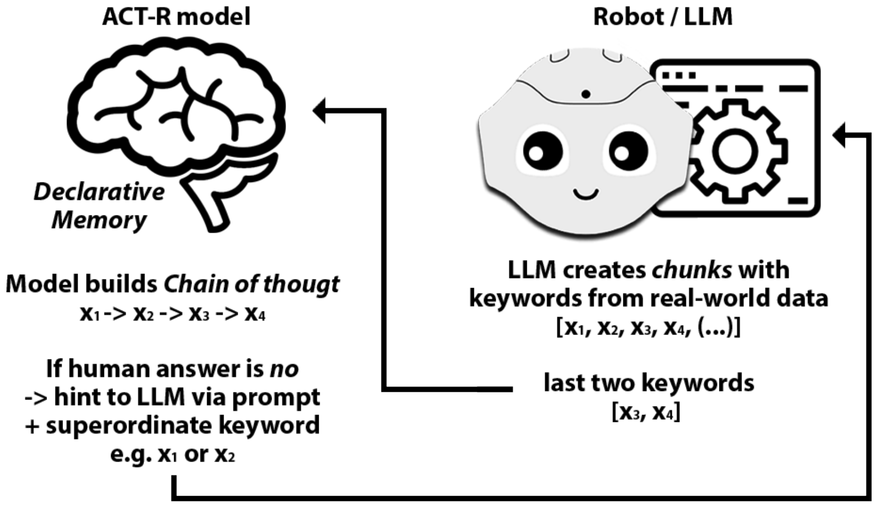
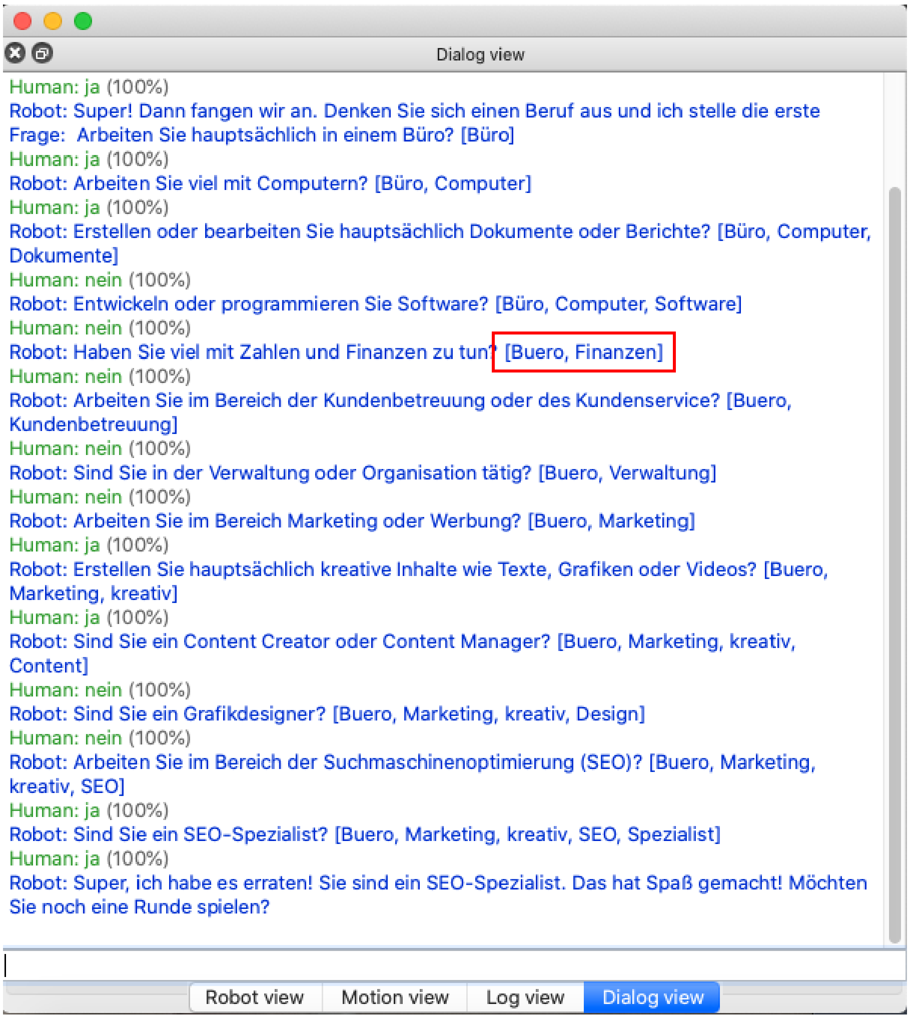

ACT-R model
Robot / LLM
Declarative Memory
Model builds *Chain of thougt*
x_1 -> x_2 -> x_3 -> x_4
If human answer is *no*
-> hint to LLM via prompt
+ superordinate keyword
e.g. x_1 or x_2
LLM creates *chunks* with
keywords from real-world data
[x_1, x_2, x_3, x_4, (...)]
last two keywords
[x_3, x_4]

Figure 3: Transfer of the last two keywords to the ACT-R model and possible feedback to the LLM

When the robot made a chain of incorrect guesses, the system prompt was modified and provided with information from the mental model of the guessing process stored in ACT-R's declarative memory. This type of prompt augmentation ensures that a system prompt appears with information such as "Your guess is currently going in the wrong direction" and additionally a stored superordinate job-specific keyword to help ChatGPT out of a possible dead end.

Figure 4 shows a typical progression of a profession guessing with ChatGPT and the ACT-R component in German, here in the emulator of the software development kit (SDK) for the Pepper robot. The LLM had focused on the first two keywords "office" and "computer" and the guessing went in the wrong direction. Then there was a rebound in which the second keyword was replaced by a new one, in this case "finance". This break or regression in the hierarchy of job-specific keywords is characterized by a frame. The keyword "computer" was omitted and a new start was made from the generic term "office". This finally led to the correct guess of the profession searched for with the keywords [office, marketing, creative, SEO, specialist]. Such a change would not have taken place without an intervention by the mental model in ACT-R.



Human: ja (100%)
Robot: Super! Dann fangen wir an. Denken Sie sich einen Beruf aus und ich stelle die erste Frage: Arbeiten Sie hauptsächlich in einem Büro? [Büro]
Human: ja (100%)
Robot: Arbeiten Sie viel mit Computern? [Büro, Computer]
Human: ja (100%)
Robot: Erstellen oder bearbeiten Sie hauptsächlich Dokumente oder Berichte? [Büro, Computer, Dokumente]
Human: nein (100%)
Robot: Entwickeln oder programmieren Sie Software? [Büro, Computer, Software]
Human: nein (100%)
Robot: Haben Sie viel mit Zahlen und Finanzen zu tun? [Buero, Finanzen]
Human: nein (100%)
Robot: Arbeiten Sie im Bereich der Kundenbetreuung oder des Kundenservice? [Buero, Kundenbetreuung]
Human: nein (100%)
Robot: Sind Sie in der Verwaltung oder Organisation tätig? [Buero, Verwaltung]
Human: nein (100%)
Robot: Arbeiten Sie im Bereich Marketing oder Werbung? [Buero, Marketing]
Human: ja (100%)
Robot: Erstellen Sie hauptsächlich kreative Inhalte wie Texte, Grafiken oder Videos? [Buero, Marketing, kreativ]
Human: ja (100%)
Robot: Sind Sie ein Content Creator oder Content Manager? [Buero, Marketing, kreativ, Content]
Human: nein (100%)
Robot: Sind Sie ein Grafikdesigner? [Buero, Marketing, kreativ, Design]
Human: nein (100%)
Robot: Arbeiten Sie im Bereich der Suchmaschinenoptimierung (SEO)? [Buero, Marketing, kreativ, SEO]
Human: ja (100%)
Robot: Sind Sie ein SEO-Spezialist? [Buero, Marketing, kreativ, SEO, Spezialist]
Human: ja (100%)
Robot: Super, ich habe es erraten! Sie sind ein SEO-Spezialist. Das hat Spaß gemacht! Möchten Sie noch eine Runde spielen?

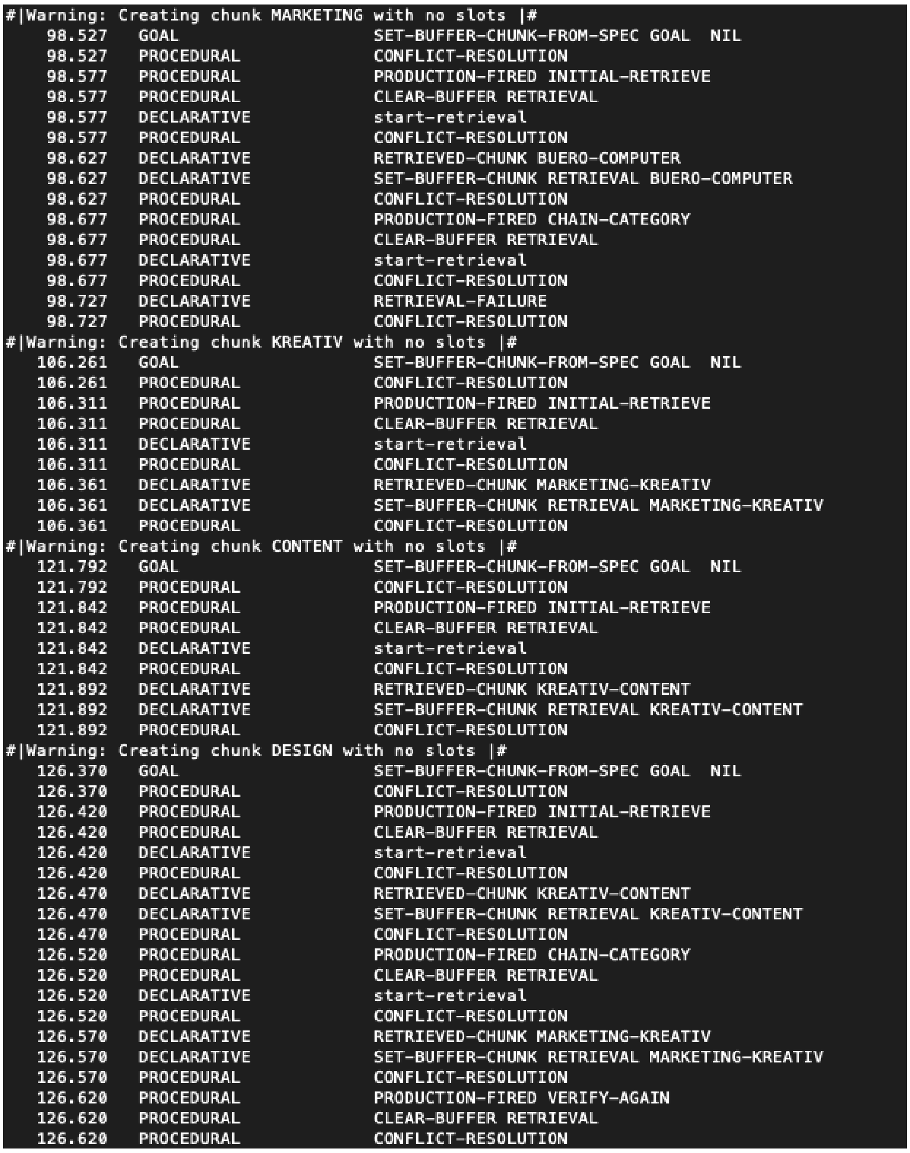
Figure 4: Dialog view of the robot emulator for testing the application with an intervention of the ACT-R model marked by a frame

2.4. Mental Modeling with ACT-R

The ACT-R model we developed for initial tests was rather simple. It used the goal buffer to exchange data with the robot application. There were no pre-created chunks in the declarative memory, all memory contents were created as soon as they were mentioned as keywords by the ChatGPT utterances. However, in principle, additional a-priori knowledge could be provided in the declarative memory.

For each guess attempt, the last two keywords in the square brackets were passed to the model as an *object-category* pair and stored. In addition, this chunk was specified as the current goal in order to focus the model's attention on it. During the course of the guessing game, the model was able to establish a reference to higher-level keywords by linking these keyword pairs and thus maintained an overview of the whole structure of the guessing process.

The tracer output of the running ACT-R model in Figure 5 shows an example of how memory chunks are created in declarative memory using job-specific keywords. In addition, the productions that were triggered are displayed as well as older chunks retrieved from memory with previously stored *object-category* pairs. The production called *chain-category* searches the stored memory chunks for matching keywords in order to identify term concatenations and, if successful, to trace the chain back to the source term and transfer it to the robot application. The application is then able to use this "chain of thought" with the next ChatGPT prompt for helpful hints and influence the guessing process.



```
#|Warning: Creating chunk MARKETING with no slots |#
     98.527   GOAL                   SET-BUFFER-CHUNK-FROM-SPEC GOAL  NIL
     98.527   PROCEDURAL             CONFLICT-RESOLUTION
     98.577   PROCEDURAL             PRODUCTION-FIRED INITIAL-RETRIEVE
     98.577   PROCEDURAL             CLEAR-BUFFER RETRIEVAL
     98.577   DECLARATIVE            start-retrieval
     98.577   PROCEDURAL             CONFLICT-RESOLUTION
     98.627   DECLARATIVE            RETRIEVED-CHUNK BUERO-COMPUTER
     98.627   DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL BUERO-COMPUTER
     98.627   PROCEDURAL             CONFLICT-RESOLUTION
     98.677   PROCEDURAL             PRODUCTION-FIRED CHAIN-CATEGORY
     98.677   PROCEDURAL             CLEAR-BUFFER RETRIEVAL
     98.677   DECLARATIVE            start-retrieval
     98.677   PROCEDURAL             CONFLICT-RESOLUTION
     98.727   DECLARATIVE            RETRIEVAL-FAILURE
     98.727   PROCEDURAL             CONFLICT-RESOLUTION
#|Warning: Creating chunk KREATIV with no slots |#
    106.261   GOAL                   SET-BUFFER-CHUNK-FROM-SPEC GOAL  NIL
    106.261   PROCEDURAL             CONFLICT-RESOLUTION
    106.311   PROCEDURAL             PRODUCTION-FIRED INITIAL-RETRIEVE
    106.311   PROCEDURAL             CLEAR-BUFFER RETRIEVAL
    106.311   DECLARATIVE            start-retrieval
    106.311   PROCEDURAL             CONFLICT-RESOLUTION
    106.361   DECLARATIVE            RETRIEVED-CHUNK MARKETING-KREATIV
    106.361   DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL MARKETING-KREATIV
    106.361   PROCEDURAL             CONFLICT-RESOLUTION
#|Warning: Creating chunk CONTENT with no slots |#
    121.792   GOAL                   SET-BUFFER-CHUNK-FROM-SPEC GOAL  NIL
    121.792   PROCEDURAL             CONFLICT-RESOLUTION
    121.842   PROCEDURAL             PRODUCTION-FIRED INITIAL-RETRIEVE
    121.842   PROCEDURAL             CLEAR-BUFFER RETRIEVAL
    121.842   DECLARATIVE            start-retrieval
    121.842   PROCEDURAL             CONFLICT-RESOLUTION
    121.892   DECLARATIVE            RETRIEVED-CHUNK KREATIV-CONTENT
    121.892   DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL KREATIV-CONTENT
    121.892   PROCEDURAL             CONFLICT-RESOLUTION
#|Warning: Creating chunk DESIGN with no slots |#
    126.370   GOAL                   SET-BUFFER-CHUNK-FROM-SPEC GOAL  NIL
    126.370   PROCEDURAL             CONFLICT-RESOLUTION
    126.420   PROCEDURAL             PRODUCTION-FIRED INITIAL-RETRIEVE
    126.420   PROCEDURAL             CLEAR-BUFFER RETRIEVAL
    126.420   DECLARATIVE            start-retrieval
    126.420   PROCEDURAL             CONFLICT-RESOLUTION
    126.470   DECLARATIVE            RETRIEVED-CHUNK KREATIV-CONTENT
    126.470   DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL KREATIV-CONTENT
    126.470   PROCEDURAL             CONFLICT-RESOLUTION
    126.520   PROCEDURAL             PRODUCTION-FIRED CHAIN-CATEGORY
    126.520   PROCEDURAL             CLEAR-BUFFER RETRIEVAL
    126.520   DECLARATIVE            start-retrieval
    126.520   PROCEDURAL             CONFLICT-RESOLUTION
    126.570   DECLARATIVE            RETRIEVED-CHUNK MARKETING-KREATIV
    126.570   DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL MARKETING-KREATIV
    126.570   PROCEDURAL             CONFLICT-RESOLUTION
    126.620   PROCEDURAL             PRODUCTION-FIRED VERIFY-AGAIN
    126.620   PROCEDURAL             CLEAR-BUFFER RETRIEVAL
    126.620   PROCEDURAL             CONFLICT-RESOLUTION
```

Figure 5: Tracer output of the running ACT-R model showing memory chunks and firing productions

This knowledge of original and hierarchically superordinate job-specific keywords can be viewed as a mental model of the course of the game and used to give ChatGPT hints via appropriately supplemented system prompts if, for example, it has maneuvered itself into a dead end while guessing. We also used the possibility of influencing the activation of individual memory chunks in a positive or negative direction depending on the answers (yes or no) of the human player.

3. Findings

After initial tests, it seems promising to pursue this approach further. Our hypothesis about the possibility of using a cognitive model to influence prompts to control the output of an LLM appears to be valid. Initial trials with our guessing game variants have shown advantages of the variant with the ACT-R model as support on some occasions. We tried out 16 different professions in the guessing game, both with and without ACT-R support. We only tried each profession once. With the support of the mental model, all professions were guessed. Without ACT-R support, 5 professions were not guessed, the LLM always repeated the same questions at some point without reaching the goal. However, we still need more test runs, more detailed investigations and experiments with variations of different parameters in the LLM (e.g. temperature) and ACT-R model (e.g. activation) in order to make clear comparative statements.

4. Conclusion

Since ChatGPT showed deviations in the questions and formulations during different rounds of the guessing game, even with the most deterministic basic setting possible, the profession searched for might be guessed more or less quickly, regardless of whether the ACT-R model is used. It is therefore difficult to assess in individual cases whether our approach actually brings benefits – both in relation to our test scenario and beyond. A larger number of test runs – later also in different application scenarios – should provide more clarity here.

The basic idea of creating and using a mental model with the help of a cognitive architecture is easily transferable to other applications and can therefore be used quite universally. The mental model that the robot builds of the current situation could be enriched with further real-world data, e.g. with emotion recognition and timing, in which the robot incorporates the obvious mood of the human counterpart into its considerations, or also with image data. In addition to language, the results of a mental model could also control other aspects of a social robot's behavior, such as movements and gestures. Technically, it would be advantageous for real-world applications if the ACT-R model could run on the robot's hardware. This should be possible with a robot whose programming is based on Python with a Python implementation of ACT-R.

References

[1] G. Sandini, A. Sciutti, P. Morasso, Artificial cognition vs. artificial intelligence for next-generation autonomous robotic agents, Frontiers in Computational Neuroscience 18 (2024). URL: https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2024.1349408. doi:10.3389/fncom.2024.1349408.

[2] A. Rossi, F. Garcia, A. Cruz Maya, K. Dautenhahn, K. Koay, M. Walters, A. K. Pandey, Investigating the Effects of Social Interactive Behaviours of a Robot on People's Trust During a Navigation Task, 2019, pp. 349–361. doi:10.1007/978-3-030-23807-0_29.

[3] S. Kambhampati, Challenges of human-aware ai systems, 2019. URL: https://arxiv.org/abs/1910.07089. arXiv:1910.07089.

[4] B. Shneiderman, Human-Centered AI, Oxford University Press, 2022.

[5] A. Werk, S. Scholz, T. Sievers, N. Russwinkel, How to provide a dynamic cognitive person model of a human collaboration partner to a pepper robot, Society for Mathematical Psychology, 2024 forthcoming.

[6] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, Y. Qin, An integrated theory of the mind 111 (2004) 1036–1060. doi:10.1037/0033-295X.111.4.1036.

[7] R. Thomson, C. Lebiere, J. R. Anderson, J. Staszewski, A general instance-based learning framework for studying intuitive decision-making in a cognitive architecture, Journal of Applied Research in Memory and Cognition 4 (2015) 180–190. doi:https://doi.org/10.1016/j.jarmac.2014.06.002, modeling and Aiding Intuition in Organizational Decision Making.

[8] A. Pipitone, A. Chella, Robot passes the mirror test by inner speech, Robotics and Autonomous Systems 144 (2021) 103838. doi:10.1016/j.robot.2021.103838.

[9] OpenAI, Transforming work and creativity with AI, Technical Report, 2024. URL: https://openai.com/product.

[10] H. Zhang, M. Dang, N. Peng, G. Broeck, Tractable control for autoregressive language generation, 2023. doi:10.48550/arXiv.2304.07438.

[11] T. Sievers, N. Russwinkel, How to use a cognitive architecture for a dynamic person model with a social robot in human collaboration (2024 forthcoming).

[12] R. Budiu, ACT-R / About, Technical Report, 2024. URL: http://act-r.psy.cmu.edu/about.

[13] Aldebaran, United Robotics Group and Softbank Robotics, Pepper, Technical Report, 2024. URL: https://www.aldebaran.com/en/pepper.

[14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, D. Amodei, Language models are few-shot learners, 2020. doi:10.48550/arXiv.2005.14165.