2025 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE) | 979-8-3315-0279-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/METROXRAINE66377.2025.11340339

Development of a Human-inspired Long-Term Memory for Interactive Conversational Agents

Angelo De Marco
Dept Information Engineering
University of Pisa
Pisa, Italy
a.demarco11@studenti.unipi.it

Federico A. Galatolo
Dept Information Engineering
University of Pisa
Pisa, Italy
federico.galatolo@unipi.it

Lorenzo Cominelli
Dept Information Engineering
University of Pisa
Pisa, Italy
lorenzo.cominelli@unipi.it

Marco Pardini
Dept Information Engineering
University of Pisa
Pisa, Italy
marco.pardini@phd.unipi.it

Mario G.C.A. Cimino
Dept Information Engineering
University of Pisa
Pisa, Italy
mario.cimino@unipi.it

Alberto Greco
Dept Information Engineering
University of Pisa
Pisa, Italy
alberto.greco@unipi.it

Enzo Pasquale Scilingo
Dept Information Engineering
University of Pisa
Pisa, Italy
enzo.scilingo@unipi.it

***Abstract*—The ability to establish and maintain relationships with users is crucial for Interactive Conversational Agents (ICAs). This involves remembering information from previous interactions and utilizing it in future conversations—key behaviors that build rapport and demonstrate interest in the user, which are essential for sustained engagement. Current conversational agents typically rely on context from recent conversational turns. To enhance interaction quality, agents should be equipped with a Long-Term Memory (LTM) that encompasses entire past conversations. The state-of-the-art approach to incorporating LTM in ICAs involves saving all previous turns in an external storage system (e.g., a plain database). Despite achieving considerable results, this method is not biologically inspired and diverges from how human memory works—where only essential information is stored and reconstructed generatively using semantic memory. We propose a hierarchical memory system inspired by human memory stratification. It dynamically updates and retrieves contextually relevant episodic and declarative memories to support semantically enriched interactions. A significant innovation in our approach is the use of a Large Language Model (LLM) Agent to determine what structured information should be stored during conversations. Our preliminary results suggest that our approach maintains competitive performance compared to existing methods while introducing novel features such as graph-based persona representation and a reduced need for storing entire past dialogues.**

***Index Terms*—conversational agents, large language models , episodic memory, semantic memory, long-term memory**

I. Introduction

Long-Term Memory (LTM) in humans is a stratified and biologically efficient storage system that retains information for extended periods. It is typically divided into Episodic Memory—personal experiences tied to specific times and places—and Semantic Memory, which stores generalized world knowledge not linked to specific events. This dual structure enables humans to recall meaningful, context-rich content without remembering every detail.

We draw inspiration from this neurocognitive organization to enhance Interactive Conversational Agents (ICAs), including Embodied Conversational Agents (ECAs), by embedding them with artificial LTM modules. Our goal is to enable these agents to interact with users more naturally and meaningfully over time.

While Large Language Models (LLMs) are often treated as repositories of generalized world knowledge (i.e., semantic memory), we explore their potential for simulating episodic memory by leveraging external memory systems. These memory modules not only emulate *pattern completion*—the ability to reconstruct full memories from partial cues—but also *pattern separation*, which reduces overlap between stored memories, inspired by the Complementary Learning Systems (CLS) theory [1]

Our work introduces a biologically inspired architecture for ICA memory using a hybrid system composed of document and graph NoSQL databases. This architecture allows the agent to maintain a dual representation of user knowledge: an episodic memory storing structured summaries of conversations, and a semantic graph encoding high-level concepts, opinions, and social connections.

This paper presents the following key contributions:

- We propose PENCIL (Persistent Episodic and Neuro-symbolic Cognitive Interaction Layer), a dual memory system for ICAs that mirrors human memory stratification. Episodic memory is implemented using a document-based database, while semantic memory is represented using a graph database to capture abstracted user knowledge and relationships.
- All stored data—conversations, summaries, and user models—are designed to be interpretable and editable by humans. This supports transparency, manual customization, and improved alignment with user expectations or scripted contexts.
- We provide a basis for future works in the same field, suggesting directions to follow

979-8-3315-0279-9/25/$31.00 ©2025 IEEE

Authorized licensed use limited to: University of Pisa. Downloaded on January 31,2026 at 11:14:17 UTC from IEEE Xplore. Restrictions apply.

We explain in the subsequent sections which information should be stored in memory and how to select and retrieve it during conversations.

II. Related Work

A. Biological Inspiration

The Complementary Learning Systems (CLS) theory [1] explains how the entorhinal cortex interfaces between the hippocampus and neocortex to manage encoding and retrieval. Two key mechanisms are central:

- **Pattern Completion:** The hippocampus reconstructs a complete memory from a partial cue by activating the associated memory trace.
- **Pattern Separation:** Only the top 4% of units with the highest excitatory input activate, ensuring similar memories are stored as distinct patterns and reducing interference.

A complementary view, explored in [2], posits that episodic memory can be treated as a generative process. Not only can episodes be reconstructed from cues, but it is also unnecessary to store complete episodes. This process, termed **semantic completion**, encodes key features and reconstructs missing details using generalized semantic knowledge.

We will mimic these approaches, following biological process to generate and retrieve memories, as highlighted in Figure 1.

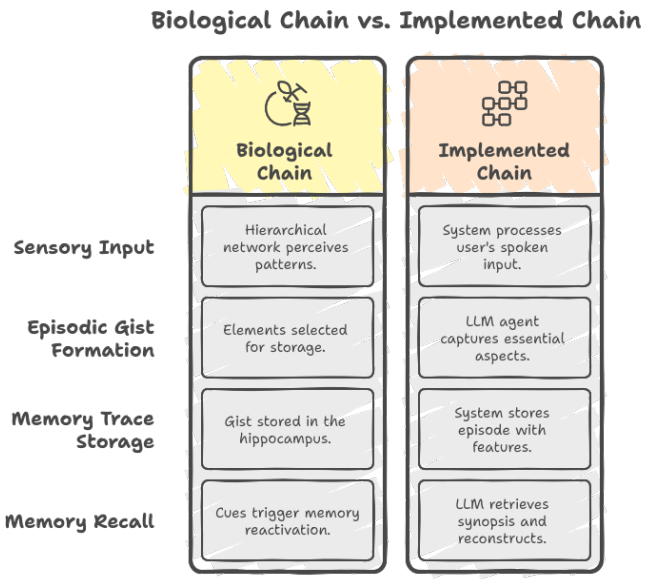

Fig. 1. Our implementation closely mimic biological processes

B. Long-Term Memory in Socially Appropriate Conversations

In Human-Robot Interaction (HRI), memory plays several roles that support socially appropriate behaviour [3]:

- Remembering factual information such as names, hobbies, nationality, etc. [4], [5];
- Following user intention, i.e. goals or plans of a user that need to be tracked over time [6];
- Encompass past events and experiences shared between robot and user [7], [8];
- Track user patterns, understanding behavior and preferences [9], [10];

Our system is capable of storing such information through LLM reasoning and summary capabilities.

C. Long-Context Transformers

Large Language Models (LLMs) struggle to maintain long-term context. Their fixed context window [11] causes performance degradation in prolonged conversations, leading to inconsistencies and forgetfulness (e.g., forgetting user preferences).

Efforts to extend transformer context [12], [13] often result in increased computational cost and reduced interpretability [14]. Consequently, Retrieval-Augmented Generation (RAG) techniques [15] are emerging as a more efficient alternative for long-term memory integration.

Differently from popular techniques, we employ an agent to decide *"at-conversation-time"* which are the most useful facts to store.

D. LLM Agents

The integration of LLMs into software engineering has introduced new interaction paradigms. Agents can be equipped with tools and reasoning strategies that determine when and how to use these tools [16], [17]. In our work, we adapt this idea by enabling an agent to decide when to store or retrieve memories during conversation.

E. Related Architectures

In this field of study, two notable architectures are:

- MemoryBank [18], which encodes event summary and user portrait offline given past conversations, and embed a mechanism of forgetting strategy biologically inspired from Ebbinghaus Curve;
- ChatDB [19], which is the first example of architecture augmenting LLMs with relational databases as symbolic memory, enabling structured storage and manipulation through SQL instruction directly generated from the model

We take the best from both, creating a service which is capable of encode summaries and portraits summarizing conversations in a structured and online manner, not being limited by user portraits but being able of generate higher level concepts.

III. Methodology

Our methodology draws direct inspiration from biological paradigms of memory, particularly the stratified organization of human memory into episodic and semantic components. The system architecture, illustrated in Figure 2, comprises three key modules: the Embodied Conversational Agent

Authorized licensed use limited to: University of Pisa. Downloaded on January 31,2026 at 11:14:17 UTC from IEEE Xplore. Restrictions apply.

(**ECA**), a Large Language Model (LLM)-based **Agent**, and a dedicated Memory **Service**. Although the embodiment of the agent is optional, and the architecture remains agnostic to the physical instantiation of the interface, we adopt the term ECA for generality and to adapt to our real-world experiment, as written in later sections.

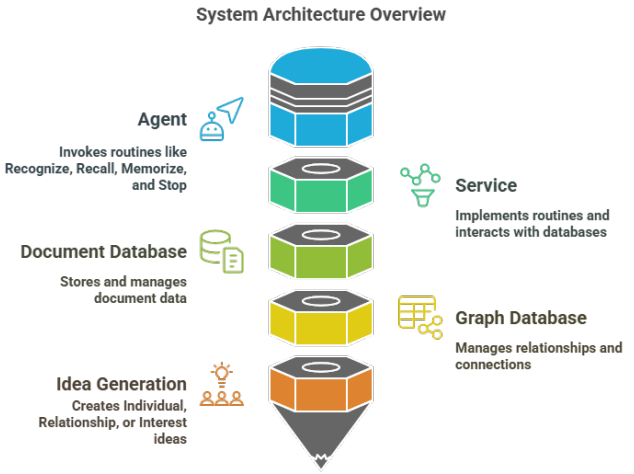

Fig. 2. Overview on how PENCIL writes new opinions during conversations

1) The interaction process begins when the user engages in conversation with the ECA. The agent must be equipped with the ability to not only process user input linguistically but also semantically *understand* and retain relevant information from the exchange. Following each interaction, the LLM Agent (referred subsequently simply as Agent) autonomously evaluates the input and determines whether and what information should be committed to long-term memory. This decision-making is grounded in the relevance, novelty, and potential future utility of the content.
2) Once a decision to store information is made, the agent summarizes and structurally annotates the event, highlighting key components such as involved entities, contextual background, emotional tone, and any inferred user preferences or beliefs. These structured summaries are passed to the Memory Service (referred subsequently simply as Service), which is responsible for persisting the information in either episodic or more declarative form, depending on its nature.
3) During memory retrieval (i.e., recall), the Agent issues a query to the Service based on the current conversational context. The Service responds by retrieving relevant episodic instances or traversing the declarative graph to return interconnected high-level knowledge. This retrieval process is inherently generative: both the agent and the memory module are capable of reconstructing memories or reasoning over abstractions, thereby emulating the neurocognitive mechanism of *semantic completion*—the ability to reconstruct full knowledge structures from partial cues using generalized semantic knowledge.

Through this design, the agent exhibits memory behaviors that are more human-like: it does not indiscriminately store all experiences, but rather filters, abstracts, and organizes information in a structured and interpretable fashion. This memory-aware capability supports socially appropriate, personalized, and context-rich interactions over long-term engagements.

In the following, we outline the primary workflows involved in our architecture:

A. Recognizing Individuals

When the ECA encounters a new individual, it must identify the person in order to access the corresponding memory. To achieve this, the system employs a deep learning-based facial recognition module, denoted as `Face()`, which extracts visual embeddings from the individual's face. These features are then compared against stored embeddings using a similarity metric and a predefined threshold.

If a match is found, all subsequent memory operations during the current interaction are associated with that individual. Otherwise, a new memory region is initialized in persistent storage to represent the new identity.

```
< conversation start >
face_features = Face(person)
ID = find(face_features, metric, threshold)
if ID is New:
   < ECA presents itself >
else:
   < ECA has recognized the person >
```

Listing 1. Recognition Pseudocode

B. Memorizing Information

During the conversation, the Agent may determine that a particular fact or event should be stored. It produces a memory representation m as a function of the current turn t and the embedded conversational context, i.e., $m = f(t \mid \text{context})$.

The memory is then structured by the Memory Service and stored in the document-oriented storage system, which constitutes the *Episodic Memory*. This ensures temporal and contextual coherence for each memorized interaction.

C. Generating Higher-Level Representations

At the end of an interaction, the ECA performs reflective reasoning on the conversation to extract abstract, high-level information about the individual. These derived concepts are stored in the *Declarative Memory* and fall into three main categories:

1) ***Individual-Level Knowledge****:* This includes personal information about the subject, such as name, age, nationality, occupation, and field of study.

2) ***Relational Knowledge****:* This refers to the subject's relationship with other individuals, such as familial or professional ties. The Service is responsible for identifying the second party and inferring the nature of the relationship.

Authorized licensed use limited to: University of Pisa. Downloaded on January 31,2026 at 11:14:17 UTC from IEEE Xplore. Restrictions apply.

*3) **Interest-Based Knowledge**:* This category includes topics and activities of interest to the subject, including hobbies or areas of curiosity.

Each high-level idea is stored in the Declarative Memory and is linked to a set of supporting episodic memories ξ. These episodes serve as evidence for the inferred concept. Additionally, each idea is assigned a *Conversation-To-Live (CTL)* value, which allows the idea to decay and eventually expire unless it is reinforced in future conversations. Thanks to this system, when an idea is no longer explored for a misunderstanding of the Agent or for a change-of-mind of the user, It will disappear from the ICA memory.

D. Recalling Stored Information

When the Agent determines that previously stored knowledge is necessary for continuing a conversation, it generates a natural language query q and sends it to the Memory Service.

Upon receiving the query, the Service first identifies the memory type most likely to contain the relevant information. It then extracts a set of memory hints and uses them to synthesize a response for the Agent.

```
< Agent queries the Service with a query q >
selected_memory = dispatch(q)
hints = getHints(selected_memory)
answer = solve(q, hints)
< Service sends answer to the Agent >
```

Listing 2. Recalling Pseudocode

IV. Implementation

A. Long-Term Memory Architecture

To manage long-term memory, we utilize two distinct NoSQL database systems based on the nature of the stored memory:

1) Episodic Memory: This component is implemented as a document-oriented database, where each individual is associated with a unique document containing facial embeddings and episodic data. Each *episode* records details such as the event content (*what* happened), location (*where*), time (*when*), and other individuals involved. If the ECA is equipped with emotion inference capabilities, the detected emotional state is also stored within the episode.

2) Declarative Memory: This component is implemented as a graph database. Each person corresponds to a connected component in the graph, with a central node containing personal attributes inspired by popolar QA datasets [20]. Neighboring nodes represent either consolidated *interests*, *relationships*, or temporary *floating ideas*. As discussed previously, an idea is supported by a collection of episodes. When this supporting set reaches a sufficient threshold, the idea is promoted to a persistent node. Otherwise, if the associated CTL counter expires, the idea and its subtree are removed.

B. Agent Implementation

The Agent used in this work is the one introduced in [21], and is equipped with four main routines: `Recognize()`, `Memorize()`, `Recall()`, and `Stop()`. These routines are implemented by the Service, while the Agent is responsible for determining the appropriate moment to invoke each routine during a conversation.

C. Service Implementation

The Service is responsible for the actual implementation of the routines invoked by the Agent. All prompts and configurations used during deployment will be made publicly available alongside the source code.

1) Recognition: The Agent extracts facial embeddings using the FaceNet architecture [22]. Matching is performed using cosine similarity, with a threshold set to $\epsilon = 0.93$. This value was empirically determined to reduce unnecessary self-introductions, which occurred more frequently at higher thresholds.

2) Memorization: To enforce structured memory formation, we adopted and extended the `Instructor`[1] library, adapting it to support both proprietary and open-source LLMs. During the memorization phase, the Agent requests the generation of an `Episode Object`, which is subsequently stored in the Episodic Memory along with a unique per-person Event Identifier (EID).

3) Recall: The recall process is performed in two stages using the same LLM infrastructure. First, the model classifies the query q as better suited for either Declarative or Episodic memory. If Declarative memory is selected, all first-degree nodes connected to the person node are retrieved and used to construct a response. If Episodic memory is selected instead, a relevance-based retrieval mechanism is used to identify and process the most pertinent episodes. Additionally, if the Declarative memory yields an unsatisfactory response—as judged by either the Agent or the user—the system automatically falls back to querying the Episodic memory.

4) Idea Generation: At the conclusion of each conversation, the episodes recorded during that interaction are analyzed to generate high-level ideas. Leveraging the `Instructor` framework, the LLM classifies each candidate into one of the predefined idea categories (Individual, Relationship, or Interest), facilitating the abstraction of meaningful semantic concepts from episodic data.

V. Experiments and Results

To evaluate our system, we adopted the *Memory Bank* dataset, publicly available via the GitHub repository[2]. This dataset comprises a collection of 15 fictional characters, each associated with a comprehensive long-term dialogue history and a corresponding set of probing questions aimed at assessing memory retrieval performance. As the official answers to these probing questions are not publicly released, we devised a custom evaluation strategy based on the Mean Average Precision (MAP) metric as defined in [20]. For answer quality assessment, we employed BERT-based semantic similarity metrics to evaluate the relevance of predicted answers with respect to our synthesized ground truth.

[1]https://github.com/567-labs/instructor

[2]https://github.com/zhongwanjun/MemoryBank-SiliconFriend

Authorized licensed use limited to: University of Pisa. Downloaded on January 31,2026 at 11:14:17 UTC from IEEE Xplore. Restrictions apply.

This evaluation framework ensures that both the factual accuracy and semantic coherence of the system's memory recall capabilities are rigorously assessed.

A. Real-World Setting

For our experiment we employed Abel [3], an hyperealistic humanoid developed at University of Pisa in collaboration with Center E.Piaggio. Its capabilities includes text-to-speech and speech-to-text, making It capable of having active conversations with humans.

B. Memorize Calls Generation

To simulate realistic interaction with our Embodied Conversational Agent (ECA), we replicated the original conversations in the dataset by prompting the `LLaMA3-70B`[4] model to decide when to invoke the `memorize()` routine. Our approach differs from prior work in that we do not store every conversational turn. Instead, we rely on the agent's judgment to store only relevant facts, closely mimicking human-like memory selection processes.

C. Ground Truth Synthesis

Since no official ground truth answers are provided with the dataset, we employed GPT-4 to synthesize both the answers and the keyword sets required for MAP score calculation.

During this process, we observed that many of the keyword sets contained noise—specifically, keywords already present in the question itself (e.g., "shared books" and "May 6th" in the question "On May 6th, I shared some books with you [...]"). If these keywords were not retrieved in the agent's response, the MAP score would incorrectly penalize the system. To mitigate this, we manually reviewed all keyword lists and removed such noisy or misleading terms, ensuring that the evaluation only reflects truly informative retrievals.

D. Performance Evaluation

A critical challenge in evaluating a memory-based system lies in distinguishing between failures of memory storage and failures of memory retrieval. Specifically, if the agent fails to invoke `memorize()` at the appropriate moment during the conversation, it becomes impossible for the system to recall that information later. In such cases, the failure is attributable to the agent's decision-making process, not the memory service itself.

To address this, we implemented a two-fold evaluation strategy:

1) **Inclusive Evaluation:** All probing questions are considered, including those for which the agent failed to store the necessary information. This evaluation reflects the end-to-end system performance, including the agent's ability to identify and memorize relevant information. We refer to this as the *"noisy"* metric.
2) **Exclusive Evaluation:** Only questions for which the required information was successfully stored via the agent's `memorize()` calls are considered. This isolates the performance of the memory service itself, allowing us to assess its precision and semantic retrieval effectiveness independent of the agent's storage decisions.

This dual evaluation strategy ensures a fair and comprehensive understanding of our system's capabilities and limitations, separating the contributions of the agent and the memory service to the final performance.

We evaluated different configurations in recall execution:

1) Memorize calls executed with Mixstral8x7B and recall done with the same model.
2) Memorize calls executed with Mixstral8x7B and recall done with GPT-4o.
3) The same as the above two configurations, but considering only questions for which the answer has been captured by an LLM agent calling Memorize.

Results are summarized in Table I. For detailed results per subject, these will be released in supplmentary material within the source code.

TABLE I
COMPARISON OF MODELS

Model	NoisyMAP	MAP	NoisySBERT	SBERT
Mixstral8x7B	0.4316	0.5081	0.4680	0.5628
GPT-4o	**0.5764**	**0.6532**	**0.5882**	**0.6608**

These results are compared with the correctness scores of the MemoryBank framework, as shown in Table II. In their work, correctness was defined as a score assigned by human evaluators, taking values of 0, 0.5, or 1. These scores were first averaged per response and then averaged across all responses for each subject.

We consider our MAP scores to be comparable to their correctness scores, since all keywords in our evaluation were manually verified by humans. This human validation step ensures that our automated metric remains aligned with human judgment.

TABLE II
MEMORYBANK CORRECTNESS COMPARISON

Model	Correctness
$\text{SiliconFriend}_{\text{ChatGLM}}$	0.438
$\text{SiliconFriend}_{\text{ChatGPT}}$	**0.716**

Our service has several unique features compared to the MemoryBank framework. Notably, our approach uses only general-purpose LLMs without fine-tuning or few-shot learning. Additionally, the graph summary represents an innovation that has not yet been evaluated, but by design, it is expected to significantly improve user engagement when interacting with the ICA. This graph-based persona representation allows the ICA to access and utilize summaries of the user's relationships, interests, and opinions, facilitating more personalized and relevant interactions.

[3]https://old.unipi.it/index.php/news/item/22750-mi-presento-sono-abel
[4]https://ollama.com/library/llama3:70b

Authorized licensed use limited to: University of Pisa. Downloaded on January 31,2026 at 11:14:17 UTC from IEEE Xplore. Restrictions apply.

VI. Conclusions and Future Directions

In conclusion, this research demonstrates that integrating biological principles with advanced AI techniques can significantly improve the memory capabilities of ICAs. The proposed LTM system not only addresses the limitations of current conversational agents but also sets a foundation for more sophisticated and human-like interactions in the future. This work underscores the importance of interdisciplinary approaches in advancing AI technologies, combining insights from neuroscience, computer science, and artificial intelligence to create more robust and capable systems.

Future work will involve connecting the graph memories of different users, enabling the ICA to discuss common interests and relationships between people. Mapping this knowledge on a graph lays the foundation for real-world conversations requiring solving a "graph task." For example, if two users share a mutual friend passionate about hiking, the ICA can dynamically generate conversations involving planning a group hike or sharing hiking experiences. This capability will significantly enhance the relevance and engagement of ICAs in social interactions, making them more adept at navigating complex human relationships. As additional future direction, we aim to enable the Agent to reason about consolidate opinions (and not only floating ones) in order to maintain a better aligned knowledge with the user's current interests.

ACKNOWLEDGMENTS

Work partially supported by the European Commission under the NextGenerationEU program, Extended Partnership PNRR PE1 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI". Work partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab project (Departments of Excellence),

References

[1] R. C. O'Reilly, R. Bhattacharyya, M. D. Howard, and N. Ketz, "Complementary learning systems," *Cogn Sci*, vol. 38, pp. 1229–1248, Aug 2014.

[2] Z. Fayyaz, A. Altamimi, C. Zoellner, N. Klein, O. T. Wolf, S. Cheng, and L. Wiskott, "A model of semantic completion in generative episodic memory," *Neural Computation*, vol. 34, no. 9, pp. 1841–1870, 2022.

[3] X. Zheng, H. Ishiguro, and D. F. Glas, "Four memory categories to support socially-appropriate conversations in long-term hri," 2019.

[4] A. M. Sabelli, T. Kanda, and N. Hagita, "A conversational robot in an elderly care center: An ethnographic study," in *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 37–44, 2011.

[5] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, A. Schultz, and J. Wang, "Designing robots for long-term social interaction," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1338–1343, 2005.

[6] A. Tapus and M. J. Mataric, "Socially assistive robots: The link between personality, empathy, physiological signals, and task performance.," in *AAAI spring symposium: emotion, personality, and social behavior*, pp. 133–140, 2008.

[7] T. Kanda, R. Sato, N. Saiwaki, and H. Ishiguro, "Friendly social robot that understands human's friendly relationships," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2215–2222 vol.3, 2004.

[8] M. K. Lee, J. Forlizzi, S. Kiesler, P. Rybski, J. Antanitis, and S. Savetsila, "Personalization in hri: A longitudinal field experiment," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 319–326, 2012.

[9] D. Glas, K. Wada, M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita, "Personal greetings: Personalizing robot utterances based on novelty of observed behavior," *International Journal of Social Robotics*, vol. 9, 04 2017.

[10] J. Fasola and M. J. Matarić, "A socially assistive robot exercise coach for the elderly," *J. Hum.-Robot Interact.*, vol. 2, p. 3–32, June 2013.

[11] S. Pawar, S. M. T. I. Tonmoy, S. M. M. Zaman, V. Jain, A. Chadha, and A. Das, "The what, why, and how of context length extension techniques in large language models – a detailed survey," 2024.

[12] A. Bertsch, U. Alon, G. Neubig, and M. R. Gormley, "Unlimiformer: Long-range transformers with unlimited length input," 2023.

[13] Z. He, Z. Qin, N. Prakriya, Y. Sun, and J. Cong, "Hmt: Hierarchical memory transformer for long context language processing," 2024.

[14] X. Wang, M. Salmani, P. Omidi, X. Ren, M. Rezagholizadeh, and A. Eshaghi, "Beyond the limits: A survey of techniques to extend the context length in large language models," *arXiv preprint arXiv:2402.02244*, 2024.

[15] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[16] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[17] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, and J. M. Zhang, "Large language models for software engineering: Survey and open problems," in *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*, pp. 31–53, IEEE, 2023.

[18] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang, "Memorybank: Enhancing large language models with long-term memory," 2023.

[19] C. Hu, J. Fu, C. Du, S. Luo, J. Zhao, and H. Zhao, "Chatdb: Augmenting llms with databases as their symbolic memory," 2023.

[20] Y. Du, H. Wang, Z. Zhao, B. Liang, B. Wang, W. Zhong, Z. Wang, and K.-F. Wong, "Perltqa: A personal long-term memory dataset for memory classification, retrieval, and synthesis in question answering," *arXiv preprint arXiv:2402.16288*, 2024.

[21] M. Pardini, F. A. Galatolo, L. Cominelli, A. De Marco, M. G. Cimino, A. Greco, and E. P. Scilingo, "A comparative evaluation of function-calling llms in a cognitive architecture." Unpublished manuscript, Dept. Information Engineering, University of Pisa, 2025.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 815–823, IEEE, June 2015.

Authorized licensed use limited to: University of Pisa. Downloaded on January 31,2026 at 11:14:17 UTC from IEEE Xplore. Restrictions apply.