Memolet: Reifying the Reuse of User-AI Conversational Memories

Ryan Yen
School of Computer Science
University of Waterloo
Canada
r4yen@uwaterloo.ca

Jian Zhao
School of Computer Science
University of Waterloo
Canada
jianzhao@uwaterloo.ca

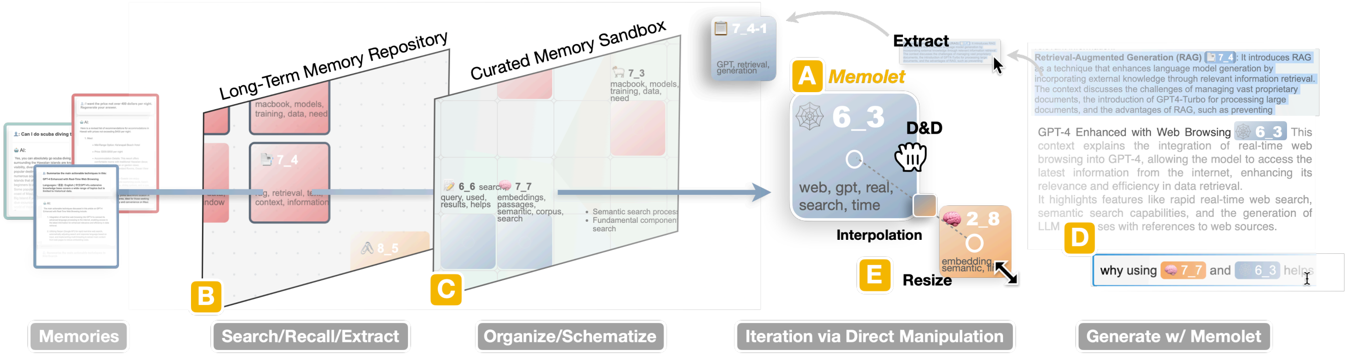

Long-Term Memory Repository
Curated Memory Sandbox
macbook, models, training, data, need
7_4
context, information
8_5
6_6 search
query, used, results, helps
7_7
embeddings, passages, semantic, corpus, search
7_3
macbook, models, training, data, need
Semantic search process
Fundamental component search
7_4-1
GPT, retrieval, generation
Extract
A
Memolet
6_3
web, gpt, real, search, time
D&D
Interpolation
2_8
embedding, semantic
E
Resize
B
C
D
Retrieval-Augmented Generation (RAG) 7_4: It introduces RAG as a technique that enhances language model generation by incorporating external knowledge through relevant information retrieval. The context discusses the challenges of managing vast proprietary documents, the introduction of GPT4-Turbo for processing large documents, and the advantages of RAG, such as preventing
GPT-4 Enhanced with Web Browsing 6_3 This context explains the integration of real-time web browsing into GPT-4, allowing the model to access the latest information from the internet, enhancing its relevance and efficiency in data retrieval. It highlights features like rapid real-time web search, semantic search capabilities, and the generation of LLM ses with references to web sources.
why using 7_7 and 6_3 helps
Memories
Search/Recall/Extract
Organize/Schematize
Iteration via Direct Manipulation
Generate w/ Memolet

Figure 1: The figure illustrates how users reuse memories for future generations by interacting with *Memolet*, the reification of memory reuse. (A) Our system initially embeds and projects all users' conversations to the long-term memory repository. (B) Users can search, recall, and extract *Memolet*s from this repository and transfer them to the curated memory sandbox. (C) This sandbox supports users in organizing and schematizing *Memolet*s based on their own sensemaking. (D) Finally, users can reuse these *Memolet*s by referring to them in the prompt and (E) refine the generation through direct manipulation.

ABSTRACT

As users engage more frequently with AI conversational agents, conversations may exceed their "memory" capacity, leading to failures in correctly leveraging certain memories for tailored responses. However, in finding past memories that can be reused or referenced, users need to retrieve relevant information in various conversations and articulate to the AI their intention to reuse these memories. To support this process, we introduce *Memolet*, an interactive object that reifies memory reuse. Users can directly manipulate *Memolet* to specify which memories to reuse and how to use them. We developed a system demonstrating *Memolet*'s interaction across various memory reuse stages, including memory extraction, organization, prompt articulation, and generation refinement. We examine the system's usefulness with an N=12 within-subject study and provide design implications for future systems that support user-AI conversational memory reusing.

KEYWORDS

Memory Reuse, Retrieval Augmented Generation, Human-AI

UIST '24, October 13–16, 2024, Pittsburgh, PA, USA

1 INTRODUCTION

Recent advances in generative AI-driven conversational agents have become a common method for users to perform tasks in various domains [55]. As users engage in more conversations and share additional details, they may discover valuable contextual information scattered at multiple previous conversations that can enrich their current conversation with the AI [30, 50, 67, 74]. In such scenarios, users encounter difficulties in resuming their conversations from where they left off, as the model may not consistently retain all pertinent memories [50, 92]. To address this issue, it is important to enable the reuse of *"memories"*—past conversations between users and generative AI [8, 31, 93]. Reusing these memories helps users reduce the need for time-consuming prompt engineering from scratch [21, 87, 88], ensure the generated results are trustworthy [57], and tailored responses to the particular context without hallucination disconnected from the memory [32, 77].

However, users face challenges in reusing memories due to the opaque nature of how current AI-driven conversational agents handle memory [31, 89, 98]. Users lack understanding of how much information is memorized by the AI and have limited control over the memory management strategies of these AI-driven conversational agents [8, 31, 93]. Additionally, users have difficulty discerning which memories are being used for generation, which hinders their ability to assess if the model accurately reuses desired memories for the current task [23, 51, 54, 95]. Therefore, to gain control over AI generation and to ensure that specific prior memories are reused without hallucinating, users often need to sift through numerous pairs or prompts/responses to find the relevant context and manually copy and paste it into the new conversation with AI. This process can be quite challenging and time-consuming, as it requires users to make sense of and recall memories [38, 49, 65, 81], extract relevant memories from various conversations [6, 11, 28], organize and integrate these memories based on their usage [14, 64], specify how AI should reuse the memories [94], and iterate on this process until the generation satisfies the users' needs [48, 79].

This research aims to explore designs that enable users to have direct control over how they want to reuse memories during conversations with generative AI. Derived from prior theories on knowledge reusing [2], information foraging [68, 69] and knowledge externalization [56], we identified several challenges and design guidelines to support users across the stages of the memory reusing process. We introduce a novel concept, *Memolet*, which reifies the notion of *reusing memories* from past conversations with generative AI (Figure 1). *Memolet* is an interactive first-class object that enables users to specify what and how the memory should be reused by direct manipulation. Users can begin by searching and extracting related *Memolet*s from a long-term memory repository consisting of all past conversations with AI to specify what memories should be reused for the current task (Figure 1.B). Then, users can organize and schematize these extracted *Memolet*s within a curated space, externalizing their thoughts on how these memories are related (Figure 1.C). Afterward, users can articulate prompts referencing these *Memolet*s to specify how they should be reused. Finally, users can refine the AI generation by manipulating the referenced *Memolet*s to align with their intentions (Figure 1.D&E).

We evaluated our system through a two-phase within-subject study involving 12 participants who regularly converse with generative AI. We demonstrate the versatility of *Memolet* in three distinct scenarios (i.e., expository writing, programming, and travel planning), wherein participants were tasked with interacting with both our system and *Baseline* in phase two, reusing conversations gathered from phase one. Our findings suggest that our system can help participants recall memories, reduce their cognitive load in organizing multiple memories, have greater perceptual control over the generative process, and be able to express how they wish to reuse memories. Overall, this paper explores the concept of memory reuse as an interactive object and an interactive system in which users can interact with these *Memolet*s to express their intentions.

2 RELATED WORK

We review prior theories and systems related to information and knowledge reuse, current methods for factual text generation and techniques for managing memory in AI-driven conversational agents.

2.1 Information Sensemaking and Reusing

Reusing information and knowledge is common across various fields like writing [16, 82], programming [19, 46], and web content management [71, 96], especially in collaborative settings where knowledge dissemination is crucial [63, 66]. Previous studies in information sensemaking have developed systems aimed at facilitating the preceding stages of knowledge reuse as proposed by Markus [2]. Mapping out these systems to the process of knowledge reuse includes capturing or documenting knowledge [6, 11, 28], packaging it for reuse [6, 11, 28], and distributing and reusing it [5, 20]. These systems have proven beneficial for individuals or groups in sensemaking information to accomplish assigned tasks.

Recent works have also introduced systems to support the sensemaking of LLM-generated content [81]. These systems support users in exploring and organizing generated results by breaking the linear structure of conversational interfaces and providing a curated space for users to make sense of the generation. This organization and structuring of information serve as a vital initial step for users to efficiently reuse the information [22, 58]. Building upon this prior research, our work extends the focus on understanding and supporting the process of user-AI conversational memory reuse. We grounded our proposed memory reuse stages and design guidelines (Figure 2) with the existing theory of knowledge reusing [2], information foraging and sensemaking [68, 69] and knowledge externalization [56]. By operationalizing these design guidelines into our system, we aim to scaffold users' sensemaking of memories before reusing them in new conversations with generative AI.

2.2 Memory Reuse in AI-Conversational Agents

LLMs have become an essential building block of current AI-driven conversational agents due to their human-like response generation [13]. However, current LLMs often are limited to handling long-term memory [37, 72, 77] and remain opaque about how they use longer contexts in downstream tasks [50, 92]. As a result, users may find it challenging to resume previous conversations where they left off and may need to manually copy and paste related memories for the AI to anchor to the correct context for the generation. Several conversational management strategies have been proposed even before the widespread adoption of transformer-based language models [84], including techniques for retaining the persona of chatbots [41, 97]. Other methods have also been suggested to guarantee that the responses generated are contextually appropriate, such as summarization [85] and refinement [99], aiming to minimize redundancy while maintaining essential information. Moreover, relevant memories can be retrieved utilizing information retrieval techniques to contextualize current inputs to AI [8, 26, 93]. However, the process of "remembering" remains complex for machines [44, 85], requiring human interventions to control the reuse of memory. Further, current AI-driven conversational agents encounter challenges in navigating diverse and complex conversations [25, 83, 90], require users to go back and forth between different conversations to collect the needed memories for reuse.

Despite improvements in memory-augmented generation, users often lack a clear understanding of how generative AI and conversational agents handle memories [31]. Current tools focus on accessing and editing chat histories to manage conversational memories [31, 61]. Our work shifts the focus from managing histories to supporting memory reuse. We transform conversational memories from static text entries into dynamic, interactive objects, *Memolet*s, enabling users to retrieve, review, and directly manipulate these memories to express their intentions for reuse.

3 SCENARIO AND DESIGN GUIDELINES

To elucidate the motivations underlying the design of *Memolet*, we walkthrough an example scenario of how a programmer interacts with conversational AI, which presents several key **[C]**hallenges

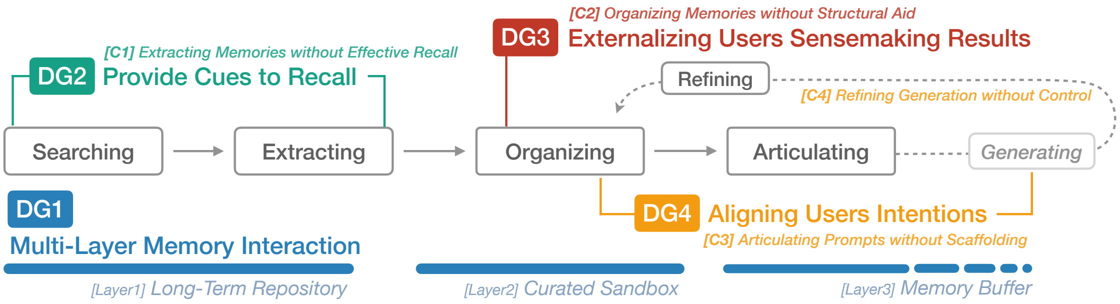

[C2] Organizing Memories without Structural Aid
DG3 Externalizing Users Sensemaking Results
[C1] Extracting Memories without Effective Recall
DG2 Provide Cues to Recall
Refining
[C4] Refining Generation without Control
Searching
Extracting
Organizing
Articulating
Generating
DG1
Multi-Layer Memory Interaction
DG4 Aligning Users Intentions
[C3] Articulating Prompts without Scaffolding
[Layer1] Long-Term Repository
[Layer2] Curated Sandbox
[Layer3] Memory Buffer

Figure 2: User-AI conversational memories reusing process with four Design Guidelines and Challenges. The first three stages of memory reuse are derived from the foraging loop within the information sensemaking process [69], outlining the processes of users seeking information, searching and filtering it, and reading and extracting information. Additionally, we draw inspiration from knowledge externalization strategies, where users extract, organize, and integrate pieces of information to scaffold the comprehension process [35]. These stages are further mapped to the knowledge reusing framework [2], which encompasses capturing and documenting knowledge, packaging and distributing knowledge, and reusing knowledge.

across different stages of memory reusing derived from the theory of knowledge reusing [2], information foraging [68, 69] and knowledge externalization [56]. Aligning with these challenges, we then introduce **[D**esign **G]**uidelines for crafting systems that support conversational memory reuse (Figure 2).

3.1 Motivating Scenario

Consider a programmer, Alicia, who is asked to preprocess and visualize a time series dataset. Having worked on the same datasets before, Alicia aims to leverage past conversations with generative AI about data preprocessing and visualization. However, this relevant information is scattered across numerous previous conversations.

[C1] Extracting Memories without Effective Recall. Alicia wants to reuse past conversations about time-series data preprocessing. However, she must manually search through multiple conversations to find relevant snippets, such as dynamic time warping (DTW) and seasonal decomposition. Eventually, she finds snippets from various methods but faces the tedious task of repeatedly copying and pasting them into the current conversation.

[C2] Organizing Memories without Structural Aid. After extracting memories, Alicia must reinterpret them for reuse [53]. She might categorize them based on their suitability for specific tasks, like handling seasonal patterns. However, Alicia is constrained to structuring the usage of these memories within a small input box. Without space for organizing memories hampers her to categorize them according to relevance and fully understand these memories.

[C3] Articulating Memories Usage without Scaffolding. Alicia aims to synthesize results aggregated from various memories from past conversations. However, expressing her intentions solely through words poses a challenge. She can only use keywords to reference the memory and lacks certainty if these keywords will guide the AI accurately. A single prompt might contain indications of which memories should be reused, how they should be reused, and Alicia's overall expectations for the generation.

[C4] Refining Generation without Control. Alicia may instruct the AI to modify, remove, or combine memories from various sources to refine responses. She might request the AI to *"include the low-pass filter code into the pattern search technique from DTW and add visualization steps from [Source X]..."* However, this approach is challenging as it requires precise articulation and a clear understanding of how AI handles these provided contexts. Without this understanding, Alicia can not discern the reason for an incorrect generation or how to rectify it, potentially distracting current models and impacting future interactions [77].

3.2 [DG1] Interacting with Memories at Multi-Layers

Overall, Alicia's process of reusing memory involves multiple layers (Layer1-3 in Figure 2). First, she **searches** and **extracts** memories that might be related to her current task (1^{st} layer). After extracting relevant memories, she **organizes** and schematizes them in a curated space, grouping them according to themes or subtopics (2^{nd} layer). Then, she **articulates** her synthesized thoughts and insights into natural language prompts for the AI, guiding it in generating code that matches her intention (3^{rd} layer). Notice the 3^{rd} layer involves an iterative process where Alicia must **refine** the generation until she is satisfied.

This multi-layered interaction mirrors how humans cognitively encode and retrieve memories from the past. Inspired by the Atkinson-Shiffrin Model [7] and Baddeley's Model of Working Memory [76], we aim to design the interaction with *Memolet* involving multiple layers as well, progressing from long-term memory (1^{st} layer) to a central executive space that controls working memories (2^{nd} layer). This process is complemented by the episodic buffer (3^{rd} layer), serving as a temporary storage that retains integrated memories from various sources. Next, these extracted *Memolet*s transition to a *curated space*, where users can actively organize them based on their own reinterpretation of how these memories should be reused. Lastly, we employ the metaphor of an *episodic buffer* to retain the results from the curated space, allowing users to apply them in the input box of the chat and serve as the context for the generative AI.

Furthermore, we explore several interaction designs to assist users in navigating through different layers, thereby easing the cognitive burden of context switching.

3.3 [DG2] Provide Visual Cues to Recall and Extract Memories

Memory recall is the prime requisite for effectively reusing memories in future conversations with AI [43]. With the exponential growth of conversations serving various purposes, it has become challenging for users to recall where specific conversations are located and retain the low-level detail of the memory. In the above scenario, Alicia recalled several memories about time series data preprocessing that might be suitable for reuse. However, the exact memory may not contain keywords like *"dynamic time warping,"* but encapsulated in a function `euclidean_distance_matrix(x, y)`. Therefore, the design of *Memolet* should incorporate crucial memory anchors that facilitate easy recall of memories. Considering that memories may scattered across various conversations, our design aims to support users in recalling and extracting memories based on their semantic meaning, eliminating the need to navigate through numerous conversations to find relevant memories.

3.4 [DG3] Flexibly Externalizing Users' Sensemaking Results about Memories

As users extract multiple memories potentially applicable to new conversations, they encounter the challenge of managing them cognitively [45, 47]. To mitigate this, our design aims to externalize users' thought process of reusing *Memolet*s, encompassing the organization and integration of memories from various sources to align with their reuse intentions. This memory organization stage aligns with the knowledge externalization strategy steps, involving selection, organization, and integration [15, 56]. By leveraging knowledge externalization strategies, users can record their thought processes using *persistent* and *manipulable* representations [15, 18, 35]. While various representations can operationalize this externalization process, graphical representations are more effective than simple note-taking [70, 80]. Additionally, maintaining flexibility in representing users' sensemaking results on memories is important due to the diversity across tasks and users.

3.5 [DG4] Aligning Users' Intentions to Reuse Memories by Direct Manipulation

In considering the intention behind reusing memories with AI through memory manipulation, we draw from Elizabeth Loftus' reconstructive memory theory [53]. This theory suggests that memories are not precise replicas but are reconstructed during recall, implying that users may reuse memory for various purposes [75]. Consequently, we conceptualize the interaction with a *Memolet* as a form of semantic interaction [24, 78]. Here, the manipulation of *Memolet*s serves to convey how users intend the memories to be reused. Given the diverse semantic meanings assigned by individual users, interactions with *Memolet*s should allow users to create, modify, delete, and integrate memories based on their intentions to iterate on the generation. For example, when Alicia articulates prompts to generate code with a specific pipeline that reuses memories from noise reduction, pattern search, and visualization, she should be enabled to effectively add and remove memories based on her current input to AI. Additionally, she should be able to adjust the usage of memories after validating the generation. For instance, if the generation does not include the low-pass filter step within the dynamic time-warping function, she can directly convey the idea of combining these two memories with ease.

4 SYSTEM DESIGN

Our design process follows a user-centered iterative approach involving four frequent users of AI-driven conversational agents, all of whom use such systems daily (3 males, 1 female; aged 21-34, M=26.8, SD=3.12). Based on feedback, we validate the challenges and operationalize the design guidelines above. Major iterations (Appendix A.1), including the integration of latent space into the linear chat interface, were implemented. However, participants reported the need to use an external notebook application to externalize their thoughts. Another low-fidelity prototype revealed that long-term memory space and sensemaking space should be separated, as the memories are interacted with for different purposes.

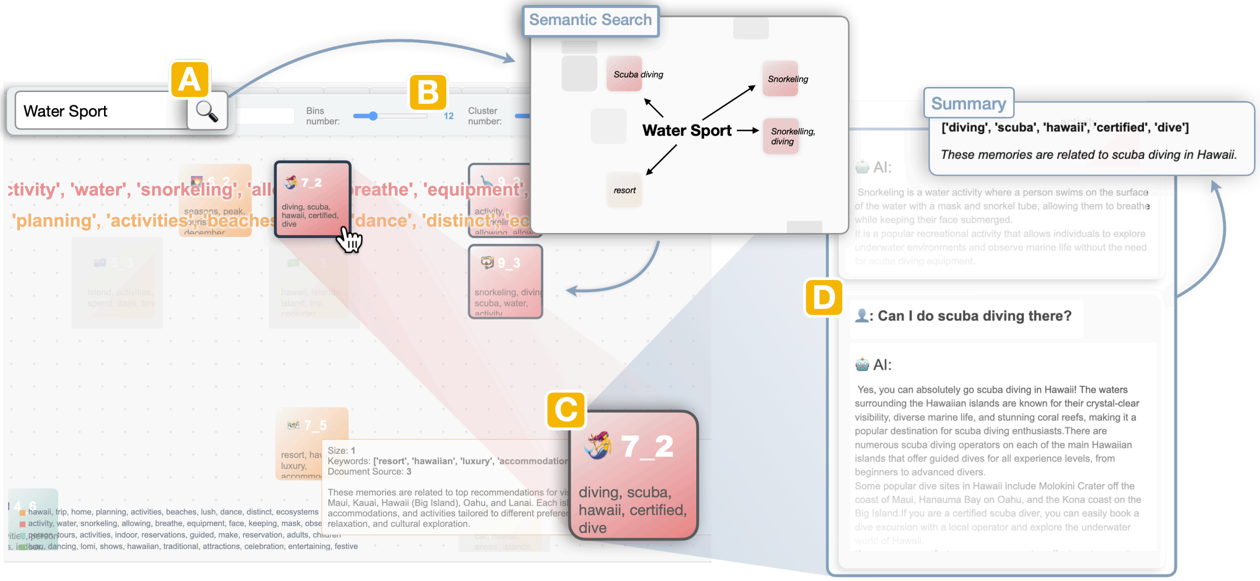
4.1 Reifying the Reuse of Memory

To support users expressing intentions of memory reusing, we reify the reuse of user-AI conversational memory as a *persistent*, *interactive*, *first-class object* called *Memolet* [10, 27]. A *Memolet* (Figure 3.C) represents a piece of past conversations users have had with AI, which may include one or multiple prompt/response pairs based on their semantic similarity. Specifically, we encode all conversations (pairs of prompts/responses) through sentence embedding to capture semantic similarities between conversations. When aggregating consecutive prompt/response pairs into a *Memolet*, considering both temporal relationships and semantic similarities between them (Appendix A.2.1). Users can interact with these *Memolet*s to convey their intention of reusing memories within their new conversations with agents [9]. To enable users to flexibly repurpose the usage of *Memolet* according to their needs in different scenarios [75], we unified the design of *Memolet*s, with variations only in colour, keywords, and icons.

4.2 System Overview and Multi-Layer Interaction with Memolet [DG1]

In the system, we design the user interaction with *Memolet* at multiple layers based on different stages of memory reusing described in Figure 2. Here, we provide an overview of the system and interactions with *Memolet*. Detailed techniques applied from natural language processing (NLP) and information retrieval (IR) in our implementation will be described in Section 4.6.

According to DG1, we propose a multi-layered approach for interacting with *Memolet*s. First, users access a **long-term memory repository** containing *Memolet*s, where they can *search* and recall memories using visual cues like keywords and summarizations (Figure 1.B). Users can select and *extract* relevant *Memolet*s and then pass to a **curated memory sandbox**, serving as a space for *organizing* and schematizing them based on users' interpretations of how they should be reused (Figure 1.C). Users can move around


A
Water Sport
B
Bins number:
12
Cluster number:
Semantic Search
Scuba diving
Snorkeling
Water Sport
Snorkelling, diving
resort
7_2
diving, scuba, hawaii, certified, dive
9_3
snorkeling, diving, scuba, water, activity
7_5
resort, hawaiian, luxury, accommodation
4_6
'activity', 'water', 'snorkeling', 'allowing', 'breathe', 'equipment',
'planning', 'activities', 'beaches', 'dance', 'distinct', 'ecosystems'
hawaii, trip, home, planning, activities, beaches, lush, dance, distinct, ecosystems
activity, water, snorkeling, allowing, breathe, equipment, face, keeping, mask, observe
person, tours, activities, indoor, reservations, guided, make, reservation, adults, children
luau, dancing, lomi, shows, hawaiian, traditional, attractions, celebration, entertaining, festive
C
Summary
D

Size: 1
Keywords: ['resort', 'hawaiian', 'luxury', 'accommodation
Dcoument Source: 3

These memories are related to top recommendations for vis
Maui, Kauai, Hawaii (Big Island), Oahu, and Lanai. Each isl
accommodations, and activities tailored to different preferen
relaxation, and cultural exploration.

['diving', 'scuba', 'hawaii', 'certified', 'dive']

These memories are related to scuba diving in Hawaii.

AI:

Snorkeling is a water activity where a person swims on the surface of the water with a mask and snorkel tube, allowing them to breathe while keeping their face submerged.
It is a popular recreational activity that allows individuals to explore underwater environments and observe marine life without the need for scuba diving equipment.

👤: Can I do scuba diving there?

AI:

Yes, you can absolutely go scuba diving in Hawaii! The waters surrounding the Hawaiian islands are known for their crystal-clear visibility, diverse marine life, and stunning coral reefs, making it a popular destination for scuba diving enthusiasts.There are numerous scuba diving operators on each of the main Hawaiian islands that offer guided dives for all experience levels, from beginners to advanced divers.
Some popular dive sites in Hawaii include Molokini Crater off the coast of Maui, Hanauma Bay on Oahu, and the Kona coast on the Big Island.If you are a certified scuba diver, you can easily book a dive excursion with a local operator and explore the underwater world of Hawaii.

Figure 3: Long-Term Memory Repository. (A) Users can utilize semantic search to find related *Memolet*s; (B) Users can adjust parameters to re-cluster or modify the bin size of *Memolet*; (C) Each *Memolet* is accompanied by visual cues such as icons, keywords, clustering colors, and summaries; (D) Each *Memolet* may contain one to many pairs of prompts/responses based on semantic meaning.

these *Memolet* and group related memories together based on their understanding. Lastly, users can reference these curated *Memolet*s in the input box when *articulating* the prompt to converse with agents (Figure 1.D). The AI-generated content also provides references on how it used these memories. We refer to these *Memolet* passed to the AI as "contexts" in the **memory buffer**, allowing the user to adjust their utilization by direct manipulation throughout the iterative generation process. Users can further *refine* the generation by manipulating the *Memolet*s, such as merging, emphasizing, or adding/removing memories (Figure 1.E).

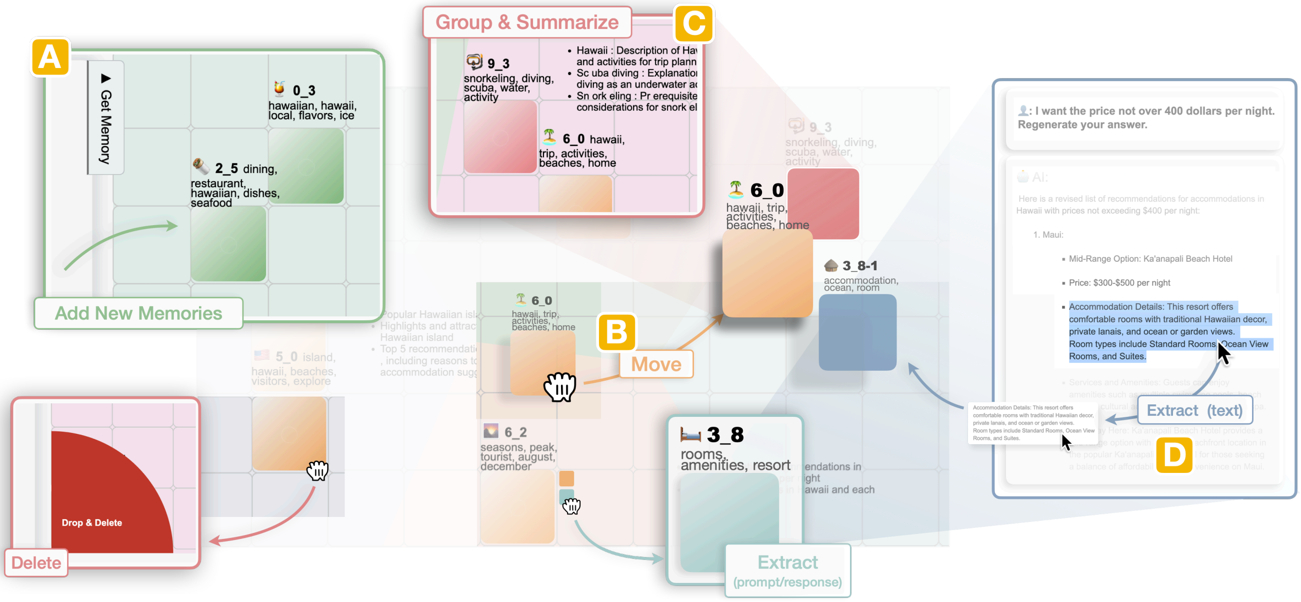
4.3 Memories Recall and Extraction [Layer1/DG2]

The process of reusing memory begins with the recall of relevant past conversations with the AI. To facilitate memory recall, we provide various cues to users. Textual summaries are offered for each conversation utilizing the OpenAI gpt-3.5-turbo (Appendix A.7.4) [62], providing insights into their content and the usage of specific *Memolet* (Figure 3.D). We also extract keywords for individual *Memolet*s and clusters using the TF-IDF vectorization, highlighting significant terms within the conversations (Figure 3.C). Additionally, unique IDs are allocated based on their location (e.g., ID: x_y refers to the *Memolet* at column x and row y). Each *Memolet* is assigned a unique icon as well, selected based on the most semantically similar icon to its contained conversations. We achieve this by utilizing the same Sentence Transformer model for encoding *Memolet*s to encode the name of icons into dense embeddings, calculating similarity using cosine similarity[1]. Users can also hover over conversations to view additional details about their content via tooltips.

To help users easily understand the holistic view of all memories across various conversations, we visualize all embedded conversations within a long-term memory repository by reducing dimensionality via UMAP. By employing sentence embeddings, users can extract related *Memolet*s effectively since conversations with similar themes appear adjacent to one another. We further leverage the K-means algorithm to cluster *Memolet*s based on their content similarities to colour code these *Memolet*s. For instance, consider a student named Celine who is planning a trip to Hawaii using our system. She can extract memories related to *water sport* by selecting all coral-coloured *Memolet*s besides a *Memolet* representing scuba diving and snorkeling (Figure 3).

Additionally, we include a semantic search feature that allows users to search for *Memolet*s based on their queries (Figure 3.A). As users type their query, we dynamically encode it and compute the cosine similarity against the stored *Memolet*s. Related *Memolet*s are then highlighted with exact sentences extracted from the original prompts/responses in the conversations that closely match the search query. Users can also adjust the binning size of the *Memolet*, thereby modifying the threshold to include more or fewer pairs of prompts/responses within a *Memolet* (Figure 3.B). This binning mechanism is a common visualization method for dealing with large amounts of data points to reduce users' cognitive load [52]. The long-term memory repository is presented as a toggleable drawer, and users can navigate between this repository and the curated memory sandbox via a toggle button (Figure 4.A). When users add or remove *Memolet*s, an animation displays the newly

[1] The icon data is provided by the Full icon Image Dataset from Kaggle


A
Get Memory
0_3 hawaiian, hawaii, local, flavors, ice
2_5 dining, restaurant, hawaiian, dishes, seafood
Add New Memories
Group & Summarize
C
9_3 snorkeling, diving, scuba, water, activity
6_0 hawaii, trip, activities, beaches, home
B
Move
3_8-1 accommodation, ocean, room
5_0 island, hawaii, beaches, visitors, explore
6_2 seasons, peak, tourist, august, december
Drop & Delete
Delete
3_8 rooms, amenities, resort
Extract (prompt/response)
Extract (text)
D

- Hawaii : Description of Ha
and activities for trip plann
- Sc uba diving : Explanatio
diving as an underwater a
- Sn ork eling : Pr erequisite
considerations for snork el

: I want the price not over 400 dollars per night. Regenerate your answer.

AI:

Here is a revised list of recommendations for accommodations in Hawaii with prices not exceeding $400 per night:

1. Maui:
   - Mid-Range Option: Ka'anapali Beach Hotel
   - Price: $300-$500 per night
   - Accommodation Details: This resort offers comfortable rooms with traditional Hawaiian decor, private lanais, and ocean or garden views. Room types include Standard Rooms, Ocean View Rooms, and Suites.

Accommodation Details: This resort offers comfortable rooms with traditional Hawaiian decor, private lanais, and ocean or garden views. Room types include Standard Rooms, Ocean View Rooms, and Suites.

Figure 4: The Curated Memory Sandbox. (A) Users can organize and schematize the extracted *Memolet*s from the repository based on their own sensemaking results; (B) All movement will be snapped to the active grid; (C) When multiple *Memolet*s get closer, they will be grouped together; (D) Users can extract *Memolet*s from prompts/responses or selected text.

added or removed *Memolet*s in the curated memory sandbox, while maintaining their original positions in the repository.

4.4 Memories Organization and Schematization [Layer2/DG3]

We provide a curated memory sandbox for users to externalize their sensemaking results of these *Memolet*s extracted from the long-term memory repository, offloading users' cognitive load.

Drag & Drop. The sandbox is populated with *"active grids"* where *Memolet*s can be positioned, rearranged, and resized. All interactions with these *Memolet*s will snap to the active grid. For example, when the user drags a *Memolet*, the system provides feed-forward via a shadow to tell the user that the nearest active grid will snap to it; users can release the mouse and drop it onto that grid (Figure 4.B). Additionally, users can drop *Memolet*s into a cornered delete area to remove certain memories.

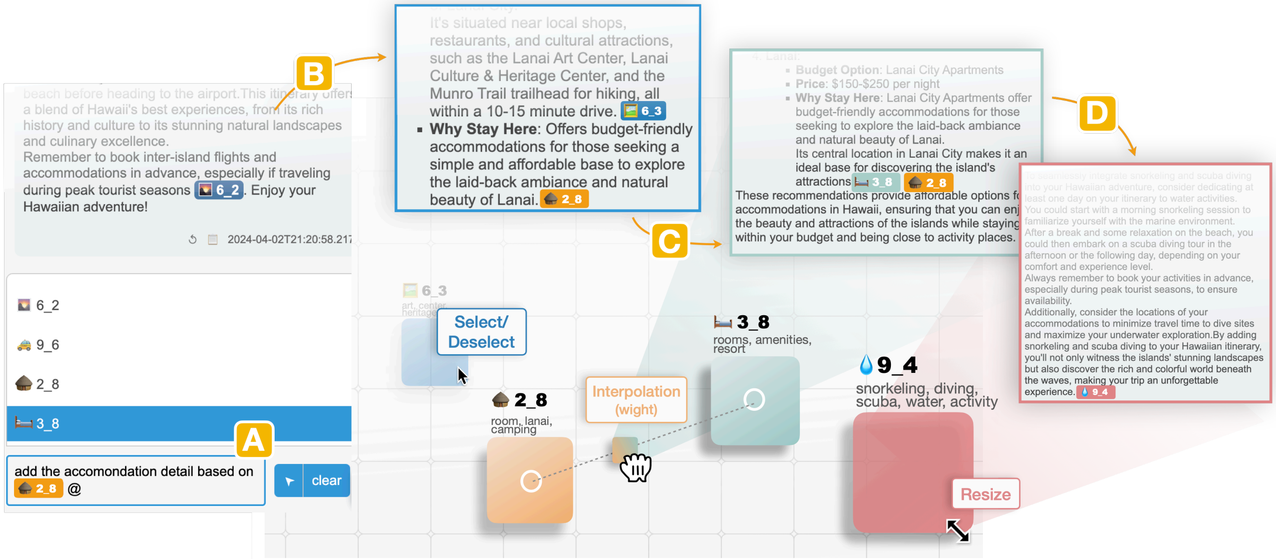
Grouping Similar Memories. In this curated memory sandbox, the background is partitioned and coloured according to groups determined by the similarity of these *Memolet*s using the Voronoi diagram. For example, if Celine selects *Memolet*s about tourist spots, restaurants, and traffic, the background colour will display three different colours, separating the *Memolet*s. When users drag the *Memolet*s, the background colour and partition dynamically update. In Figure 4, Celine is dragging a *Memolet$_{6\_0}$* towards another *Memolet$_{9\_3}$*, which is indicated by a feedforwarded white border and glow effect. When Celine drops the *Memolet$_{6\_0}$*, both *Memolet*s are then partitioned into another group, indicated by a different background colour. A summarization of all *Memolet*s in this group is then generated by GPT-3.5-turbo (Appendix A.7.4) and displayed beside. The ungrouping mechanism is activated when a *Memolet$_{6\_0}$* is dragged away beyond a threshold from other *Memolet*s, the *Memolet$_{6\_0}$* will automatically be removed from its original group and assigned the background colour of the original group. This grouping feature is helpful when Celine wants to create subgroups, such as separating water sports from tourist spots.

Extracting Memories at Different Granularity. The user can extract a child *Memolet*—a pair of prompt/response from a *Memolet* (Figure 4, *Memolet$_{3\_8}$*). Users can drag this child *Memolet* listed beside the parent *Memolet* and then drop it onto any active grid. Clicking on the *Memolet* also displays the associated conversation beside the memory sandbox, allowing users to extract sentences or code snippets from original prompts/responses to create a new *Memolet*. For instance, if Celine clicks on a *Memolet$_{6\_2}$* containing multiple conversations about Hawaii's tourist spots, she can pick one child *Memolet* about the resort and turn it into a new *Memolet$_{3\_8}$*. Later, she might want AI to provide the accommodation details in the resort, so she clicks on *Memolet$_{3\_8}$*, selects and drags related text to the sandbox and create a new *Memolet$_{3\_8\text{-}1}$* (Figure 4.D).

4.5 Generating with Memories [Layer3/DG4]

After organizing *Memolet*s based on users' interpretation of the usage of these memories, users can begin using *Memolet* as contexts provided for conversational agents (Figure 5).

Articulating a Prompt with Memories. When sent a prompt, the system will retrieve related contexts from all memories in the sandbox for generation (Section 4.6.1). Users can type in *"@"* and traverse through the *Memolet*s among the curated space to refer to them inside the prompt (Figure 5.A). For example, Celine can



beach before heading to the airport.This itinerary offers a blend of Hawaii's best experiences, from its rich history and culture to its stunning natural landscapes and culinary excellence.
Remember to book inter-island flights and accommodations in advance, especially if traveling during peak tourist seasons 🌄6_2. Enjoy your Hawaiian adventure!

2024-04-02T21:20:58.217

add the accomondation detail based on 🛖2_8 @

It's situated near local shops, restaurants, and cultural attractions, such as the Lanai Art Center, Lanai Culture & Heritage Center, and the Munro Trail trailhead for hiking, all within a 10-15 minute drive. 🖼6_3
- **Why Stay Here**: Offers budget-friendly accommodations for those seeking a simple and affordable base to explore the laid-back ambiance and natural beauty of Lanai. 🛖2_8

4. **Lanai:**
- **Budget Option**: Lanai City Apartments
- **Price**: $150-$250 per night
- **Why Stay Here**: Lanai City Apartments offer budget-friendly accommodations for those seeking to explore the laid-back ambiance and natural beauty of Lanai.
Its central location in Lanai City makes it an ideal base for discovering the island's attractions 🛏3_8 🛖2_8

These recommendations provide affordable options fo accommodations in Hawaii, ensuring that you can enj the beauty and attractions of the islands while staying within your budget and being close to activity places.

To seamlessly integrate snorkeling and scuba diving into your Hawaiian adventure, consider dedicating at least one day on your itinerary to water activities.
You could start with a morning snorkeling session to familiarize yourself with the marine environment.
After a break and some relaxation on the beach, you could then embark on a scuba diving tour in the afternoon or the following day, depending on your comfort and experience level.
Always remember to book your activities in advance, especially during peak tourist seasons, to ensure availability.
Additionally, consider the locations of your accommodations to minimize travel time to dive sites and maximize your underwater exploration.By adding snorkeling and scuba diving to your Hawaiian itinerary, you'll not only witness the islands' stunning landscapes but also discover the rich and colorful world beneath the waves, making your trip an unforgettable experience. 💧9_4

Figure 5: The memory buffer considers the *Memolet* during generation. (A) Users can refer to *Memolet*s in the sandbox while articulating prompts; (B) The generated results will cite the *Memolet*; (C) Users first attempt to refine the generation by interpolating two *Memolet*s, deselecting one *Memolet* from citations; (D) Users regenerate results by resizing a *Memolet* to highlight context contained.

select a *Memolet2_8* about accommodation while articulating the instruction about this memory. She can construct a prompt such as *"adding the accommodation details based on* 🏠2_8 *at the end of each day in my trip itinerary."* By referencing these *Memolet*s directly in prompts, users can directly control the articulation of prompts and convey how they expect these memories to be used.

Generation with Memory Citations. To facilitate users' evaluation of whether the AI utilizes the provided memories rather than hallucinating, the model is instructed to "cite" these *Memolet*s inside the generated results (Figure 5.B; Appendix A.7.1); users can hover over them to see the corresponding *Memolet* highlighted in the sandbox. Continuing the previous example, Celine observes that the generation added accommodation details inside the itinerary, citing both 🏠2_8 and details of sightseeing spots citing 🖼6_3 She clicks on the 🖼6_3 and the *Memolet6_3* in the sandbox is highlighted with a glow effect, indicating that this memory is in the tourist spots group about a culture center.

Refining Generation through Direct Manipulation. Consider that Celine is dissatisfied with the generated results because she prefers to add more details about accommodations. Our system provides a direct way of expressing these adjustments in how memories should be used by enabling users to manipulate the *Memolet*s. For instance, Celine can click the regeneration button and remove *Memolet6_3*, related to tourist spots, and add *Memolet3_8* about hotel room, and *Memolet9_4* that were not originally used in the generation (Figure 5.C). This **selection/deselection** instructs the AI to include or exclude context from these *Memolet*s, giving users more control over the required memories.

We also provide manipulation such as **interpolation**, allowing users to combine multiple *Memolet*s to create a new *Memolet* that summarizes these memories with different weights. Users can hold onto the circle control centered at *Memolet2_8* and drag a line towards *Memolet3_8*, with feedforward showing which *Memolet* is being connected before the cursor is dropped. A smaller-sized *Memolet* is generated in the middle of the line, and users can drag it towards connected *Memolet*s to determine the summarization's leaning. The model will consider this interpolation and generate content that combines these memories instead of illustrating them separately with citations 🏠2_8 🛏3_8 (Figure 5.C).

Lastly, users can **resize** the *Memolet*s to convey the importance of certain memories. For instance, Celine can resize *Memolet9_4* to specify the need to highlight more context about water sports in the generation. The regenerated results will add a section about water sports citing 💧9_4 that related to water activities (Figure 5.D). These instruction-based generation refinements are accomplished by adapting the RAG process described in Section 4.6.2.

Encoding New Conversations. After users complete a conversation session, the system will encode all pairs of prompts/responses to the long-term memory repository for future use. Users can also extract text from a response and create a new *Memolet* during the conversation, supporting in-session memory reusing.

4.6 System Implementation

We detail retrieval-augmented generation used for both the *Baseline* and our system, along with instructed generation for our system, and the overall system architecture (Figure 12).

4.6.1 Retrieval Augmented Generation. We employ a retrieval augmented generation (RAG) process to integrate context retrieval with text generation. The adapted RAG consists of several steps: generating queries to capture various aspects of the context, retrieving similar context using vector similarity search, fusing retrieved results using reciprocal rank fusion, determining top-k results based on fused scores, and utilizing these results as context for the generation model to ensure the generated responses are grounded in relevant information [73] (Appendix A.2.2).

4.6.2 Instructed Generation. The *"refining generation with manipulation"* feature follows most of the steps described above but uses different prompts to instruct model generation based on the provided instructions. The system first modifies the user's prompt according to the instructions, constructing a new query prompt that let the AI to extend the user's question as specified, such as adding or removing context, highlighting or obscuring context, or merging context with relative weights (Appendix A.7.2). Contexts relevant to the user's question are then retrieved from the data store based on the provided instructions and separated into different sections of the prompt (e.g., Context to add; Context to highlight) for incorporation into the AI's response (Appendix A.7.3). This approach ensures that the response adheres to the instructions and incorporates the relevant contexts.

4.6.3 System Architecture. Both our system and the *Baseline* are implemented in Typescript using the Svelte framework [17]. They utilize Python as the backend server for handling the RAG process and Firebase Firestore for event logging. We utilized the state-of-the-art NLP models (i.e., GPT-4) for generation in both the *Baseline* and our system [60]. Additionally, GPT-3.5 is utilized for generating summarizations when grouping *Memolet*s and generating queries in the RAG process within our system [59]. Through pilot testing, we found that GPT-4 could better adhere to users' instructions when refining generation without repeating responses [3]. However, our main contribution lies in reifying users' intentions of reusing memories and supporting users in interacting with their memories throughout the reuse process. We do not claim contributions to our adapted RAG pipeline. As the algorithm advances, we believe this design of interaction with *Memolet* will remain applicable.

5 USER STUDY

The system aims to support the comprehensive memory reuse process by enabling users to interact with the *Memolet*. To investigate the usefulness of the system for memory reuse, we conducted a within-subject study and tested its flexibility in three different scenarios. The study was divided into two phases, with the first phase requiring participants to perform tasks using an LM-driven conversational interface. A day later, the same participants were invited to perform tasks using assigned systems that followed up the previous task, which required them to reuse the knowledge and conversations gained from the first phase. The study investigated four different aspects aligned with four design guidelines:

1. User interactions with *Memolet* across stages of memory reusing.
2. How users recall and extract memories for reuse.
3. How users organize *Memolet*s to externalize their sensemaking.
4. Alignment of users' intention of reusing memories with AI.

5.1 Participants

We recruited 12 participants through convenience sampling via a university email list (7 women and 5 men; age: 21-38, *M*=26.67, *SD*=4.75). Participants reported frequent use of AI-driven conversational agents (*M*=5.16, *SD*=1.72 days/week) and familiarity with AI conversational agents (*M*=4.33, *SD*=0.74 on a 5-point scale). Participants were asked in advance about their familiarity with programming and expository writing to avoid assigning them to unfamiliar scenarios (Appendix A.5).

5.2 Scenarios and Tasks

To demonstrate the versatility of *Memolet*, we selected three distinct scenarios for participants utilizing AI-driven conversational agents. Participants engaged in tasks spanning expository writing, programming, and trip planning (Appendix A.4). Tasks for each scenario in phase one involved seeking information, synthesizing a report based on provided context, and providing a comparison table to compare different options. In phase two, participants were required to reuse information conversed with agents from phase one and generate a report, program, or a thorough plan. To mitigate carry-over effects from learning and order effects, participants were assigned two different tasks, ensuring counterbalancing such that they encountered a different task with each condition.

5.3 Phase One Study Setup

To contextualize users' memory reusing process, participants in each scenario were assigned identical tasks. The purpose of this phase is to motivate the user to actively converse with AI to remember the context of what is being discussed.

5.3.1 Study Procedure. Participants were required to fill in a consent form and complete a pre-study questionnaire regarding their demographics before the study. During the study, participants assigned to the same scenario were tasked with two different tasks. They were asked to use our *Baseline*, similar to ChatGPT (Figure 13), for each task within 20 minutes. During a total of $20 \times 2 = 40$ minutes, participants freely interacted with the system and wrote the synthesized report in a Google Doc. They were instructed to think aloud about their thought processes throughout the session [39].

5.3.2 Collected Conversations. All participants were able to complete the assigned tasks, with a total of 38 conversations (i.e., new chats created) (*M*=3.16, *SD*=0.98) and 347 pairs or prompts/responses (*M*=28.92, *SD*=9.54) collected in the first phase.

5.4 Phase Two Study Setup

We compared our system to a *Baseline* system simulating a standard conversational agent in a within-subject study design.

5.4.1 Procedure. During the study, each participant used both systems to conduct the two assigned tasks designed to nudge them to recall and reuse past conversational memories from the first phase. To control for individual differences and learning behaviour, we counterbalanced tasks and conditions to reduce order effects. Each task lasted 20 minutes, and participants were instructed to think aloud. Before using our system, participants were given a 5-minute tutorial and another 5 minutes to explore its features. Following the completion of each task, participants were asked to fill out the same post-study questionnaire. The study concluded with a 20-minute semi-structured interview, bringing the total duration to approximately 75 minutes. Participants received $35 compensation for their time.

5.4.2 Baseline System. The *Baseline* condition utilized a conversational agent that simulated ChatGPT, which is currently the most prevalent LLM-driven conversational agent. We developed this *Baseline* because the specific methods used by ChatGPT to handle chat memories remain a black box. To ensure a fair comparison, we employed the same techniques to handle conversational memories and used the same prompt and technique for retrieval augmented generation for both our system and the *Baseline*. Additionally, we implemented a semantic search feature on *Baseline* to provide both systems with the same starting point, allowing participants to focus on the subsequent procedure of memory reuse. All user actions, such as copy/paste, switching chats, scrolling through conversations in a chat, and writing prompts, were logged for further data analysis. The detail of the *Baseline* is in Appendix A.6. We do not consider the concurrent 2D prompting interfaces (e.g., Graphologue [33]) as a baseline because they mostly focus on exploring AI outputs in a two-dimensional space, whereas our study focuses on past memories for contextualizing AI's input.

5.4.3 Measures. Usability was measured using the UMUX-LITE scale, which is directly related to the SUS score [40], and the NASA-TLX scale for perceived cognitive load [29] (Appendix A.3.1). Utility was measured using self-defined Likert scale items (Appendix A.3.3). Both systems logged various types of events based on participants' interactions during the study, including written prompts.

5.4.4 Data Analysis. We transcribed the think-aloud data and post-study interviews for all participants by Otter.ai [1]. Subsequently, we analyzed these transcriptions using reflexive thematic analysis [12]. Our approach combined inductive and deductive methods to identify codes and themes, with a particular emphasis on participants' interactions with *Memolet* across stages of the memory reusing process. We conducted statistical analysis on the comparative survey data by comparing responses between the *Baseline* and *Memolet* conditions using the Wilcoxon signed-rank test, given the ordinal nature of Likert-scale responses and the small sample size. In the upcoming sections, we will present the results in the following format: for questionnaire data, ($Q_{\text{question \#}}$: $\text{Median}_{Memolet}$ vs. $\text{Median}_{Baseline}$, p=p-value, r=effect size), and for other quantities, ($\text{Mean/Median}_{Memolet}$ vs. $\text{Mean/Median}_{Baseline}$, p=p-value). Additionally, prompts collected from the system log were categorized by whether or not referred to past memories and the type of prompt (Figure 9). Two researchers coded the data collaboratively, achieving an initial inter-coder agreement of 92%, which was iteratively refined to 100%.

6 FINDINGS

In this section, we present findings from our analysis of participants' survey responses, think-aloud protocols, interviews, and system usage logs. Our overarching goal was to explore how users interact with *Memolet* during the memory reuse process, methods of recalling and extracting memories, organization of *Memolet*s to externalize their thought process, and alignment of users' intentions of reusing memories with AI.

6.1 Overall Usage of *Memolet* in Memory-Reusing Process

Participants were able to complete all assigned tasks in both conditions without a significant difference in task completion time (M_M=12.62 min vs. M_B=14.37 min, p=0.072).

Our system supports different stages of memory-reusing process. The average system usability scores computed from UMUX-LITE were significantly greater (p = .003) for our system (Mdn = 91.67), compared to the *Baseline* (Mdn = 41.67). Participants consistently reported significantly better results with our system on all subjective metrics (Figure 6) from searching memory (Q_7: Mdn_M=4.5 vs. Mdn_B=2.5, p=0.0023, r=0.204), extracting related memories (Q_8: Mdn_M=4.0 vs. Mdn_B=2.5, p=0.0021, r=0.204), organizing memories (Q_9: Mdn_M=4.5 vs. Mdn_B=2.0, p=0.002, r=0.204), articulating prompts to specify how the memories should be reused (Q_{10}: Mdn_M=5.0 vs. Mdn_B=2.5, p=0.005, r=0.541), to refining generation which contained memories (Q_{11}: Mdn_M=5.0 vs. Mdn_B=3.0, p=0.0032, r=0.353).

Participants recall and extract memories first before conversing with AI. Based on the observations in Figure 7, participants using our system tended to extract memories first and then engage in conversation with the AI in relatively later stages, whereas with *Baseline*, participants tended to extract memories later in the process. P2 explained, *"I initially trusted that ChatGPT [Baseline] can remember my memories if I am continuing on the same chat, but it turns out not."* Other participants mentioned similar reasons, such as feeling the need to extract related memories after *"cannot validate the generation"* [p3] and *"if AI could not understand what memories refer to"* [p8], as it required too much effort to *"find related memories to reuse"* [p10]. Most participants (N=10) felt that our system reminded and assisted them in extracting and organizing memories, which proved it helpful when articulating prompts and further evaluating and refining the generated results.

Using our system helps focus on participants' current tasks. Overall, participants had more conversations using *Baseline* compared to that using our system (Mdn_M=26.0 vs. Mdn_B= 10.5, p=0.0049, r=0.309). To understand the reason, we further analyzed both prompts and generated results, schematizing them based on participants' types of prompts. From Figure 9, we observed that 82% of prompts from the *Baseline* referred back to memory, where participants primarily aimed to summarize multiple memories (26%), acquire more detailed information about memories (28%), and clarify their prompts (20%). We found that most prompts in the *Baseline* were about 'get info', which included finding related memories by conversation, validating if the generation correctly attributes to memories, and acquiring what memories *Baseline* remembers. We also observed that participants using *Baseline* tended to start from summarizing, aggregating, and getting information about the memories by continuing on their previous chats. Most participants (N=9) preferred this approach due to concerns about feeling *"lost while*

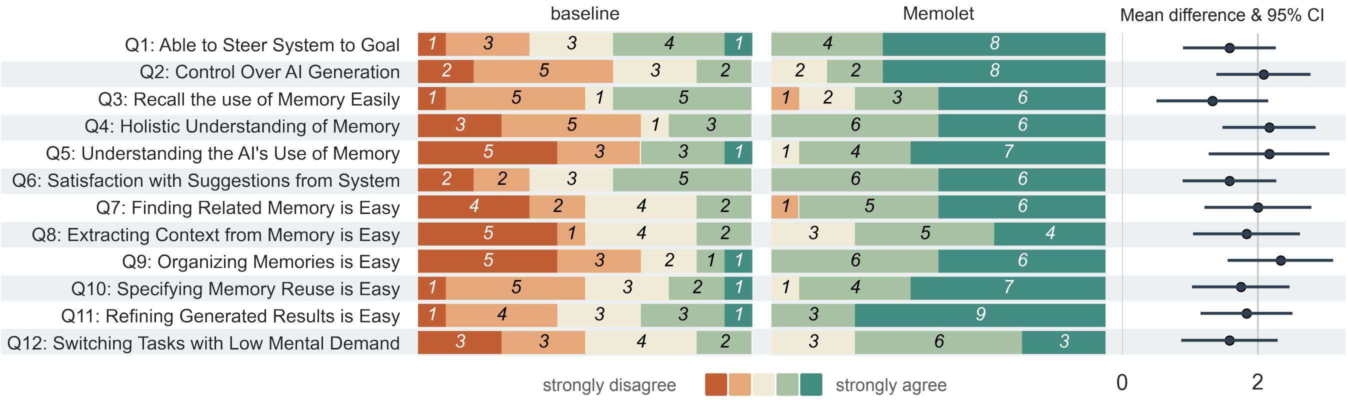


Figure 6: Participants’ responses when rating the 5-point self-defined Likert scale questionnaire for both our *Baseline* and our system. Dots represent the mean differences of our system compared to the *Baseline*. Bars indicate the 95% CI calculated using the studentized bootstrap method.

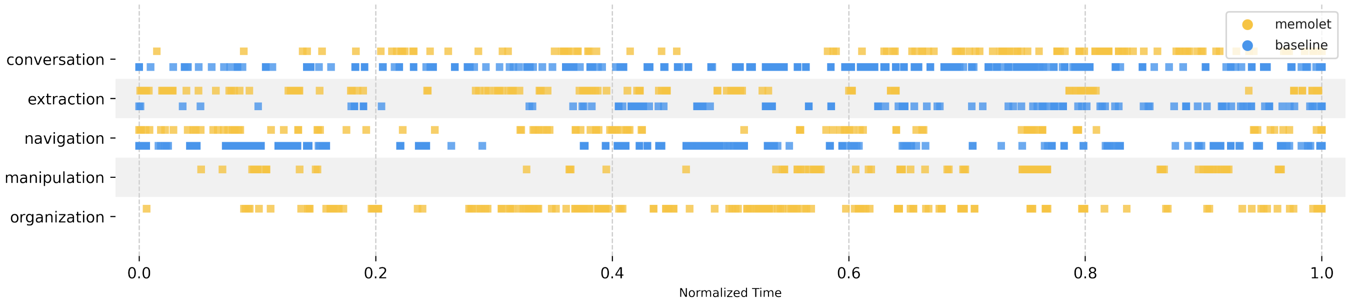


Figure 7: Distribution of logged events across normalized time aggregated by 12 participants, comparing the *Baseline* and our system. The graph indicates an increase in conversations (with AI) during the later stages of the study for our system, with participants extracting more in the early stage compared to the *Baseline*.

scrolling up and down” [p4] and *“copying and pasting previous conversations into the current chat”* [p10]. While using our system, participants tended to provide prompts that move on to the ‘next step’ toward the goal of the task (29%). Participants explained that using our system helped *“not wasting the time on engineering the prompt”* [p2] or *“ask GPT [AI] to clarify where the memories came from.”* [p7]

6.2 Recalling and Extracting Memories

Participants recalled the holistic view of memories (Q_4: Mdn_M=4.5 vs. Mdn_B=2.0, p=0.0031, r=0.352) as well as the use of single memory more easily using our system than those with the *Baseline* (Q_3: Mdn_M=5.0 vs. Mdn_B=2.0, p=0.0031, r=0.353).

Our system assists them to recall what the memory was. Most participants (N=10) expressed that our system helped them recall memories throughout the stages of reusing, including through keywords, summarization, clustering, and grouping mechanisms. P2 elaborated that elaborated that *Memolet* and the clustering better helped recall the context compared to the *“scrolling and reading text-heavy conversations in ChatGPT [Baseline].”* However, two participants mentioned that the *Baseline* could help them recall a single memory better because the way they encoded the memory in the phase one study was the same as they decoded it when using *Baseline* in phase two. Despite this, they mentioned the most challenging aspect is locating a single memory across numerous conversations. Without a latent space for them to explore and recall, participants using the *Baseline* indicated that the semantic search function was inadequate because they sometimes could not recall any keywords.

Extracting memories using our system is easy. While there is no significant difference in terms of the amount of extraction comparing the two conditions (see Figure 8), participants overall find using our system to extract needed memories easier and more intuitive (see Figure 6, Q8). The representations of *Memolet* in the long-term memory repository also make it easier for them to extract *“memories that are similar”* [p12]. The clustering helps participants understand *“whether enough context has been extracted to tailor to the current need”* [p11]. Participants mentioned that when they reused the same context multiple times, it becomes cumbersome in *Baseline* where they decide not to create new chats but extend their prior conversations. However, this approach hinders them when they need to *“synthesize results from multiple different sources”* [p8].

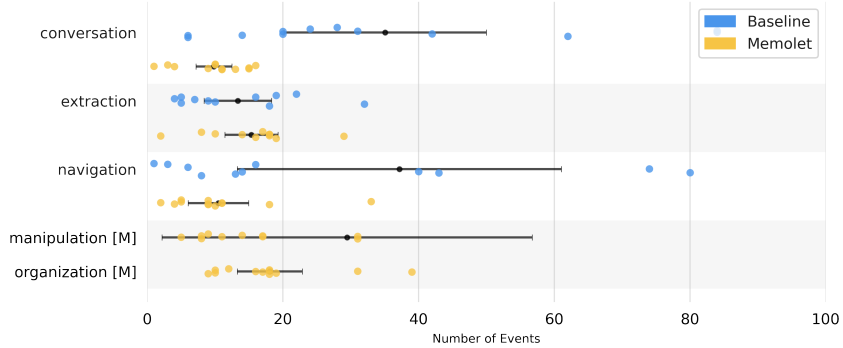


Figure 8: Distribution of system log events comparing the *Baseline* System and our system. Black dots represent means, and the bars denote 95% CI.

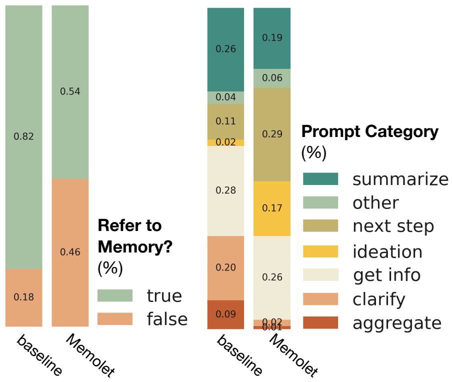


Figure 9: Left: Whether prompts referring to memory. Right: Categories of prompts.

Focusing on the gist and reinterpret the Memolet during extraction. Several participants (N=7) preferred how our system presents memories in the long-term memory repository, which they found intuitive when finding related memories for new interactions with the AI. We noticed that participants were more focused on the high-level gist about each memory provided by *Memolet*s when providing the context for AI, rather than the low-level details of the content. P1 highlighted the usefulness of keywords on *Memolet*, stating that *"the keywords are enough to recall without the need to look into the original conversation."* Additionally, participants using our system tended to reinterpret memories based on their current needs due to the unified design of *Memolet*s. P6 explained, *"I can reuse the same Memolet for different purposes since it might provide context for different prompts in different ways."*

6.3 Externalizing Users' Sensemaking Results to Lower the Cognitive Load

We used NASA-TLX to measure the perceived workload. Compared to *Baseline*, our system required lower mental (Mdn_M=2.5 vs. Mdn_B=5.5, p=0.020), physical (Mdn_M=2.0 vs. Mdn_B=3.0, p=0.05), and temporal (Mdn_M=2.0 vs. Mdn_B=5.0, p=0.002) demand, required less effort (Mdn_M=2.5 vs. Mdn_B=5.0, p=0.002), and led to better performance (Mdn_M=6.0 vs. Mdn_B=4.0, p=0.002) and statistically significantly less frustration (Mdn_M=1.0 vs. Mdn_B=3.5, p=0.032). The overall perceived workload, obtained by averaging all six raw NASA-TLX scores, was also lower for our system compared to that for *Baseline* (Mdn_M=2.083 vs. Mdn_B=4.16, p=0.002).

Organizing Memolets helps planning their next step. Participants utilized the curated memory sandbox for various purposes. Most participants (N=7) used it to externalize their sense-making results of *Memolet* and its corresponding memories. P6 and P7 used this space to recall memories; P3, P5, and P10 used it to plan for the next step and how to approach the task. P10 also mentioned that organizing these *Memolet* helped to identify what context was still needed, for example, *"I just found that I haven't planned for emergencies."* In contrast, most participants (N=9) using *Baseline* expressed difficulties organizing memories within the input box. Some participants (N=4) also requested to use Google Docs to record their copied text before sending it for generation. P11 mentioned that *"when finding related memories, I do not have enough bandwidth to think about what I have extracted already."*

Our system reduces the need for context switching. From Figure 8, we observed that participants navigated (i.e., going back to prior conversations) significantly less when using our system (Mdn_M=9.0 vs. Mdn_B=17.0, p=0.031). Participants also reported that context switching in our system required lower mental demand (Q_{12}: Mdn_M=4.0 vs. Mdn_B=2.5, p=0.006, r=0.61). This reduction in context switching can be attributed to several factors: firstly, the visual cues designed for *Memolet* required participants to recall from conversations; secondly, participants possessed more trust in our system, eliminating the need to validate results from original conversations; and lastly, they could extract all required *Memolet*s at once without going back and forth when writing new prompts.

6.4 Aligning Users' Memory Reuse Intention

Participants reported a significantly better understanding of how AI reused the provided memories (Q_5: Mdn_M=5.0 vs. Mdn_B=2.0, p=0.003, r=0.35) and higher satisfaction with the generation results (Q_6: Mdn_M=4.5 vs. Mdn_B=3.0, p=0.003, r=0.35) from our system compared to those from the *Baseline*.

Using our system requires less prompt engineering and articulate more precise prompt. All participants noted that using our system required less prompt engineering than using *Baseline*. We observed from Figure 9 that participants required more 'clarification' (20% among all prompts) when using *Baseline* compared to using our system (2%). This indicates that participants had to clarify their intentions to the AI more frequently when using *Baseline*. One reason cited was that participants could refer to and select/deselect memories with control, allowing them to *"understand what context [memories] were provided"*p8. P1 mentioned that the generation process tended to *"understand what to do based on my [their] provided context [memories]."* These mechanisms also provide users more control in defining the scope of memory reused. Another advantageous feature mentioned was the use of '@', which referred to specific *Memolet*s participants preferred to use. For instance, P6 utilized this feature extensively, stating that *"the generation would not hallucinate and stick to my provided Memolets."* This feature helped participants express their intentions clearly, leading to *"more precise answers using memories correctly although it can serve multiple purposes"* [p11].

Refining generated results by manipulating Memolet is intuitive. All participants attempted to regenerate results by manipulating the cited *Memolet*s at a later stage of the study (see Figure 7). They found the manipulation, including add/remove, interpolate, and resize *Memolet*s, to be *"intuitive"* and *"convey their intention without the need for writing prompts"* [p11]. Some participants (N=3) mentioned that these manipulation features helped *"disambiguate"* [p3] the AI and *"enhance the explainability"* [p11] since they progressively match their intention with AI's understanding. Participants using *Baseline* expressed frustration when attempting to rectify errors, as they did not know *"how to specify the required changes"*p6.

Our system enhances the controllability over the generation. Most participants (N=10) reported that the AI-generated results not only better aligned with their intentions but also provided them with more control over the generation process itself (Q_2: Mdn_M=5.0 vs. Mdn_B=2.0, p=0.002, r=0.34). All participants expressed that the citation included in the generation and the direct reference (i.e., '@') to the *Memolet* helped them evaluate *"whether the generation followed their instruction clearly"* [p2]. P7 further explained that the visual representation of memory (i.e., *Memolet*) provided the feedback on what they were trying to reuse the memories, *"I know that when I interpolated two memories together, the generated results will combine them."* In comparison to *Baseline*, participants expressed the loss of control when the generation does not match their intention and did not know *"how to repair from the failure"* [p6].

Users develop custom memory reusing strategies through interaction with Memolet. Compared with the *Baseline*, participants were able to guide our system more effectively towards the goal of their task (Q_1: Mdn_M=5.0 vs. Mdn_B=3.0, p=0.003, r=0.35). P12 explained that the curated sandbox provided a canvas to express their own *"strategy of how to reuse them [Memolet],"* enabling them to steer the system towards their goal based on their plan. We observed that some participants (N=4) carefully planned their approach to synthesizing the report while organizing *Memolet*s in the sandbox. Figure 10 demonstrates how P1 and P3 organized *Memolet*s in the sandbox to synthesize their final report for the task. P10 elaborated on the reason for spending time organizing the curated space: to understand what *"memories are needed or not being covered before."* By doing so, participants can control their own strategy of memory reusing towards their goal, *"step by step"* [p2].

7 DISCUSSION

We discuss the trade-off between the need for sensemaking spaces and designing spatial effects, the differences between tasks, the balance between trust and over-reliance, and the applicability of our concepts to media other than conversational memory.

7.1 The Design of Space

Two participants mentioned that the curated space might not be necessary for them when not many contexts need to be reused; however, they indicated that organization *"still happens, but in my mind."* [P3]. We acknowledge that the current system is a proof-of-concept prototype designed to support stages of interacting with *Memolet*. Further studies are needed to understand its applicability to tasks that may not be knowledge-intensive. In such cases, the need for externalizing sensemaking results may be less critical, as we discovered that participants sometimes organize memories within the input box. Nonetheless, we argue for the necessity of the sensemaking space to reduce cognitive load, organize information, and carry information across stages of information seeking [65, 81].

7.2 Scenarios and Tasks

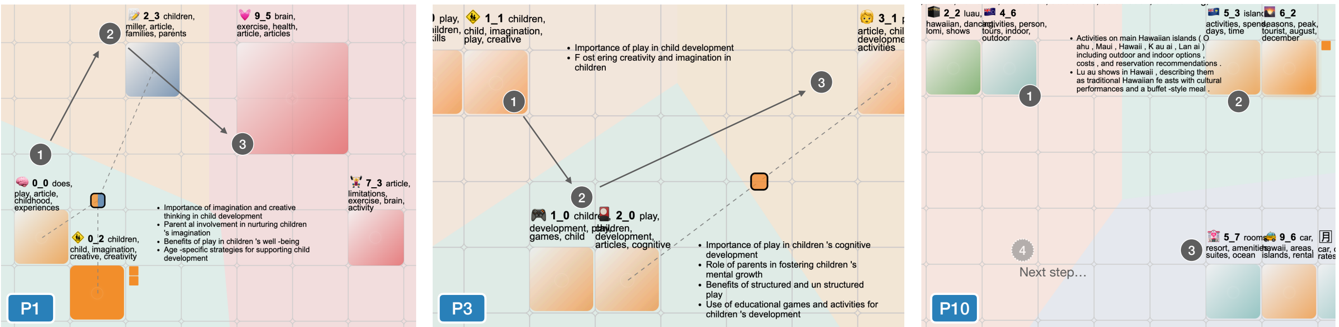
The inclusion of three different scenarios is not meant for comparison but rather to demonstrate the flexibility of the concept of *Memolet*. However, interesting insights have arisen due to the distinct nature of these scenarios. For instance, participants in scenario three (i.e., trip planning) tend to refer to memories in prompts less frequently (37.5%) compared to the other two scenarios (60%, 68.6%) when using our system. This might be attributed to the fact that less factual accuracy is required for trip planning compared to expository writing and programming scenarios. We also noted that participants in the first two scenarios navigated back to the original conversations more frequently. They required more detailed information rather than just the overall gist of each *Memolet*. Nonetheless, participants in scenario three still reused memories to *"reduce the time required to look for history [memories]"* [P10]. There is also no significant difference among subjective self-defined questionnaire questions across the three scenarios after conducting a one-way ANOVA test (p=0.12).

7.3 Trust and Overreliance

Some participants (N=5) reported in the interview that they trusted the generation from our system more, because of the alignment of generated results to their intention of memory reuse. However, two participants placed excessive trust in the system, as evidenced by P6's stress upon discovering missing conversations not covered in phase one (e.g., the web-augmented generation) and the need for organizing *Memolet*s in the sandbox. P6 explained, *"I kinda panic when found that I did not have that memory to reach the goal [in phase two] and had to put down my current work first."* Similarly, we observed that many participants N=8 tended to skip the validation (i.e., click on the references) in the later stage of conversations. Although this might be attributed to that they already grasp what that *Memolet* is for, P2 explained, *"I found the system understands me all the time, so I skip the validation part."* This further raises the concern if the generation cited *Memolet*s but still hallucinates. Future improvement can adopt techniques such as automatic evaluation [95] and explain the AI-generated results.

7.4 Polymorphism and Reuse of the *Memolet*

Further, *Memolet* represents the reification of how users reuse conversational memories, embedding a concept that can exhibit polymorphism and be repurposed to various types of memories [10]. For


P1
1
2
3
0_0 does, play, article, childhood, experiences
2_3 children, miller, article, families, parents
9_5 brain, exercise, health, article, articles
7_3 article, limitations, exercise, brain, activity
0_2 children, child, imagination, creative, creativity
P3
1_1 children, child, imagination, play, creative
3_1 article, chil developme activities
1_0 children, development, play, games, child
2_0 play, children, development, articles, cognitive
P10
2_2 luau, hawaiian, dancing, lomi, shows
4_6 activities, person, tours, indoor, outdoor
5_3 islands, activities, spend, days, time
6_2 seasons, peak, tourist, august, december
5_7 rooms, resort, amenities, suites, ocean
9_6 car, hawaii, areas, islands, rental
car, rates
4
Next step…

- Importance of imagination and creative thinking in child development
- Parent al involvement in nurturing children 's imagination
- Benefits of play in children 's well -being
- Age -specific strategies for supporting child development

- Importance of play in child development
- F ost ering creativity and imagination in children

- Importance of play in children 's cognitive development
- Role of parents in fostering children 's mental growth
- Benefits of structured and un structured play
- Use of educational games and activities for children 's development

- Activities on main Hawaiian islands (O ahu , Maui , Hawaii , K au ai , Lan ai) including outdoor and indoor options , costs , and reservation recommendations .
- Lu au shows in Hawaii , describing them as traditional Hawaiian fe asts with cultural performances and a buffet -style meal .

Figure 10: The examples of participants' curated memory sandbox after organizing and manipulating *Memolet*s. The arrow indicates the order in which they synthesized their report.

instance, it could be reused to accommodate the embedded text of a corpus of papers' abstracts in a scholarly setting, enabling scholars to ideate and engage in expository writing [4]. Similarly, it could be utilized in programmers' codebases, facilitating exploratory programming [34, 42]. Within our system, we also exemplify this concept's reuse by empowering users to transform extracted text from documents or ongoing conversations into new *Memolet*s. Users could manipulate those *Memolet*s using the same concept of expressing their intentions of memory reuse. We believe that future designs can provide users with more customized features based on their own datasets.

8 LIMITATIONS AND FUTURE WORK

To contextualize the results, we designed three knowledge-intensive tasks that required participants to reuse the memory they acquired in phase one. However, we did not assess whether our system performs equally well in real-world scenarios where tasks are more varied. For instance, our system might not be useful for tasks that do not prioritize context, such as rewriting or grammar fixing, or tasks where only the prompt template matters, such as image generation. Future designs could consider the variations between tasks and validate the usefulness of reusing the concept of *Memolet* in specific scenarios. Furthermore, our study was divided into two phases with a one-day gap in between, during which participants might recall memories not solely based on the visual cues provided by *Memolet*. Future studies could extend to long-term deployment studies to understand the effectiveness of our system as memory degradation occurs over time. Additionally, the current memory reusing process described in Figure 2 was synthesized from existing knowledge in externalization, information foraging, and information reusing theory or frameworks. Future work could build upon our research by further investigating the validity of this memory-reusing process, which might vary over time, such as differences observed with long-term usage.

In terms of system design, we believe future work could extend the memory-reusing concept beyond conversational interfaces. For example, it could be integrated into in-IDE code generation [91] or in-document text generation. This notion involves a trade-off between the need for sensemaking spaces and the design for space effectiveness as previously discussed in Section 7.1. We anticipate the future design could be condensed and minimal, allowing users to utilize it as an extension for reusing memories while retaining the concept of organization and manipulation of *Memolet* when needed. While our evaluation used relatively small datasets to contextualize results, *Memolet* is designed to handle larger data. Key features include merging and extracting memories across different granularities; dynamic sizing of text and *Memolet* squares based on data volume; and adjustable binning to change the amount of data in a single bin (Figure 3b), proven effective for conveying both global patterns and local features in literature and production systems [52, 86]. These mechanisms, along with semantic search and zooming/panning, support users in exploring larger datasets. However, issues such as text overlap can potentially be addressed using text exclusion techniques from [36]. Lastly, we acknowledge the importance of privacy concerns, which were not the main focus of this paper. Future deployment of these memory management tools should consider techniques such as data encryption, access control, and data anonymization, which are being studied in software development.

9 CONCLUSION

In this paper, we explore novel ways of interacting with memories from past conversations with generative AI. We propose a memory-reusing process and four design guidelines derived from prior theories. We introduce *Memolet* as a reification of 'memory reusing' for users to manipulate their conversation memories with AI directly. We demonstrate *Memolet*'s utility across multiple memory-reusing stages with a novel system and evaluate its effectiveness through a two-phase study. Our findings suggest improved memory recall, reduced cognitive load, and enhanced control over the generative process. We believe *Memolet* offers valuable insights for enabling the intuitive and controlled reuse of conversational memories.

ACKNOWLEDGMENTS

We thank all our participants for their time and valuable input. We would also like to thank our reviewers whose insightful comments have led to a great improvement of this paper. This work is supported in part by the NSERC Discovery Grant and the University of Waterloo faculty start-up fund.

REFERENCES

[1] 2023. Otter.AI -AI Meeting Note Taker and Real-time AI Transcription. https://otter.ai/

[2] Lynne M. Markus . 2001. Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success. *Journal of Management Information Systems* 18, 1 (May 2001), 57–93. https://doi.org/10.1080/07421222.2001.11045671 Publisher: Routledge _eprint: https://doi.org/10.1080/07421222.2001.11045671.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[4] Griffin Adams, Emily Alsentzer, Mert Ketenci, Jason Zucker, and Noémie Elhadad. 2021. Beyond Summarization: Designing AI Support for Real-World Expository Writing Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 4794–4811. https://doi.org/10.18653/v1/2021.naacl-main.382

[5] Maryam Alavi et al. 2000. Managing organizational knowledge. *Framing the domains of IT management: Projecting the future through the past* 15 (2000), 28.

[6] Rana Alkadhi, Teodora Lata, Emitza Guzmany, and Bernd Bruegge. 2017. Rationale in development chat messages: an exploratory study. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. IEEE, 436–446.

[7] Richard C Atkinson and Richard M Shiffrin. 1968. Human memory: A proposed system and its control processes. In *Psychology of learning and motivation*. Vol. 2. Elsevier, 89–195.

[8] Sanghwan Bae, Donghyun Kwak, Soyoung Kang, Min Young Lee, Sungdong Kim, Yuin Jeong, Hyeri Kim, Sang-Woo Lee, Woomyoung Park, and Nako Sung. 2022. Keep Me Updated! Memory Management in Long-term Conversations. https://doi.org/10.48550/arXiv.2210.08750 arXiv:2210.08750 [cs].

[9] Michel Beaudouin-Lafon. 2000. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 446–453. https://doi.org/10.1145/332040.332473

[10] Michel Beaudouin-Lafon and Wendy E. Mackay. 2000. Reification, polymorphism and reuse: three principles for designing visual interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (Palermo, Italy) *(AVI '00)*. Association for Computing Machinery, New York, NY, USA, 102–109. https://doi.org/10.1145/345513.345267

[11] Krishna Bharat. 2000. SearchPad: Explicit capture of search context to support web search. *Computer Networks* 33, 1-6 (2000), 493–501.

[12] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in neural information processing systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[14] Joseph Chee Chang, Nathan Hahn, and Aniket Kittur. 2020. Mesh: Scaffolding Comparison Tables for Online Decision Making. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 391–405. https://doi.org/10.1145/3379337.3415865

[15] Michelene TH Chi. 2009. Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in cognitive science* 1, 1 (2009), 73–105.

[16] Daniel T Citron and Paul Ginsparg. 2015. Patterns of text reuse in a scientific corpus. *Proceedings of the National Academy of Sciences* 112, 1 (2015), 25–30.

[17] Svelte contributors. 2023. Svelte: cybernetically enhanced web app. https://svelte.dev/

[18] Richard Cox. 1999. Representation construction, externalised cognition and individual differences. *Learning and instruction* 9, 4 (1999), 343–363.

[19] Thomas Davenport, Sirkka Jarvenpaa, and Michael Beers. 1996. Improving knowledge work processes. *MIT Sloan Management Review* (1996).

[20] Thomas H Davenport and Morten T Hansen. 1999. *Knowledge management at Andersen consulting*. Harvard Business School Pub.

[21] Paul Denny, Viraj Kumar, and Nasser Giacaman. 2023. Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1136–1142. https://doi.org/10.1145/3545945.3569823

[22] Nancy M Dixon. 2000. *Common knowledge: How companies thrive by sharing what they know*. Harvard Business School Press.

[23] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling Large Language Models to Generate Text with Citations. https://doi.org/10.48550/arXiv.2305.14627 arXiv:2305.14627 [cs].

[24] Camille Gobert and Michel Beaudouin-Lafon. 2023. Lorgnette: Creating Malleable Code Projections. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–16.

[25] Jonathan Grudin and Richard Jacques. 2019. Chatbots, humbots, and the quest for artificial general intelligence. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–11.

[26] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[27] Han L. Han, Miguel A. Renom, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2020. Textlets: Supporting Constraints and Consistency in Text Documents. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376804

[28] Han L. Han, Junhang Yu, Raphael Bournet, Alexandre Ciorascu, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2022. Passages: Interacting with Text Across Documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3491102.3502052

[29] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[30] Lucas Torroba Hennigen, Shannon Shen, Aniruddha Nrusimha, Bernhard Gapp, David Sontag, and Yoon Kim. 2023. Towards Verifiable Text Generation with Symbolic References. https://doi.org/10.48550/arXiv.2311.09188 arXiv:2311.09188 [cs].

[31] Ziheng Huang, Sebastian Gutierrez, Hemanth Kamana, and Stephen Macneil. 2023. Memory Sandbox: Transparent and Interactive Memory Management for Conversational Agents. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 1–3. https://doi.org/10.1145/3586182.3615796

[32] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.

[33] Peiling Jiang, Jude Rayan, Steven P. Dow, and Haijun Xia. 2023. Graphologue: Exploring Large Language Model Responses with Interactive Diagrams. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 3, 20 pages. https://doi.org/10.1145/3586183.3606737

[34] Mary Beth Kery, Amber Horvath, and Brad Myers. 2017. Variolite: Supporting Exploratory Programming by Data Scientists. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1265–1276. https://doi.org/10.1145/3025453.3025626

[35] David Kirsh. 2010. Thinking with external representations. *AI & society* 25 (2010), 441–454.

[36] Kyle Koh, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. 2010. Maniwordle: Providing flexible control over wordle. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1190–1197.

[37] Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. *arXiv preprint arXiv:2205.09726* (2022).

[38] Andrew Kuznetsov, Joseph Chee Chang, Nathan Hahn, Napol Rachatasumrit, Bradley Breneisen, Julina Coupland, and Aniket Kittur. 2022. Fuse: In-Situ Sensemaking Support in the Browser. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3526113.3545693

[39] Clayton Lewis. 1982. *Using the" thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.

[40] James R. Lewis, Brian S. Utesch, and Deborah E. Maher. 2013. UMUX-LITE: When There's No Time for the SUS. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France). Association for Computing Machinery, New York, NY, USA, 2099–2102. https://doi.org/10.1145/2470654.2481287

[41] Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155* (2016).

[42] Xingjun Li, Yizhi Zhang, Justin Leung, Chengnian Sun, and Jian Zhao. 2023. EDAssistant: Supporting Exploratory Data Analysis in Computational Notebooks with In Situ Code Search and Recommendation. *ACM Trans. Interact. Intell. Syst.* 13, 1, Article 1 (mar 2023), 27 pages. https://doi.org/10.1145/3545995

[43] Brian Y. Lim and Anind K. Dey. 2010. Toolkit to support intelligibility in context-aware applications. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing* (Copenhagen, Denmark) *(UbiComp '10)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/1864349.1864353

[44] Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. 2023. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719* (2023).

[45] Michael Xieyang Liu, Jane Hsieh, Nathan Hahn, Angelina Zhou, Emily Deng, Shaun Burley, Cynthia Taylor, Aniket Kittur, and Brad A. Myers. 2019. Unakite: Scaffolding Developers' Decision-Making Using the Web. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 67–80. https://doi.org/10.1145/3332165.3347908

[46] Michael Xieyang Liu, Aniket Kittur, and Brad A. Myers. 2021. To Reuse or Not To Reuse? A Framework and System for Evaluating Summarized Knowledge. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021), 166:1–166:35. https://doi.org/10.1145/3449240

[47] Michael Xieyang Liu, Aniket Kittur, and Brad A. Myers. 2022. Crystalline: Lowering the Cost for Developers to Collect and Organize Information for Decision Making. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3491102.3501968

[48] Michael Xieyang Liu, Advait Sarkar, Carina Negreanu, Benjamin Zorn, Jack Williams, Neil Toronto, and Andrew D. Gordon. 2023. "What It Wants Me To Say": Bridging the Abstraction Gap Between End-User Programmers and Code-Generating Large Language Models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–31. https://doi.org/10.1145/3544548.3580817

[49] Michael Xieyang Liu, Tongshuang Wu, Tianying Chen, Franklin Mingzhe Li, Aniket Kittur, and Brad A. Myers. 2024. Selenite: Scaffolding Online Sensemaking with Comprehensive Overviews Elicited from Large Language Models. https://doi.org/10.1145/3613904.3642149 arXiv:2310.02161 [cs].

[50] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173.

[51] Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848* (2023).

[52] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. 2013. imMens: Real-time visual querying of big data. In *Computer graphics forum*, Vol. 32. Wiley Online Library, 421–430.

[53] Elizabeth F Loftus. 1981. Reconstructive memory processes in eyewitness testimony. In *The Trial Process*. Springer, 115–144.

[54] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511* (2022).

[55] Marcello M. Mariani, Novin Hashemi, and Jochen Wirtz. 2023. Artificial intelligence empowered conversational agents: A systematic literature review and research agenda. *Journal of Business Research* 161 (2023), 113838. https://doi.org/10.1016/j.jbusres.2023.113838

[56] Richard E Mayer. 1984. Aids to text comprehension. *Educational psychologist* 19, 1 (1984), 30–42.

[57] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661* (2020).

[58] Carla O'dell and C Jackson Grayson. 1998. If only we knew what we know: Identification and transfer of internal best practices. *California management review* 40, 3 (1998), 154–174.

[59] OpenAI. 2024. GPT-3.5 Turbo fine-tuning and API updates. https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates

[60] OpenAI. 2024. GPT-4. https://openai.com/research/gpt-4

[61] OpenAI. 2024. Memory and new controls for ChatGPT. https://openai.com/blog/memory-and-new-controls-for-chatgpt

[62] OpenAI. 2024. Pioneering research on the path to AGI. https://openai.com/research/overview

[63] Margit Osterloh and Bruno S Frey. 2000. Motivation, knowledge transfer, and organizational forms. *Organization science* 11, 5 (2000), 538–550.

[64] Srishti Palani, Zijian Ding, Austin Nguyen, Andrew Chuang, Stephen MacNeil, and Steven P. Dow. 2021. CoNotate: Suggesting Queries Based on Notes Promotes Knowledge Discovery. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–14. https://doi.org/10.1145/3411764.3445618

[65] Sharoda A. Paul and Meredith Ringel Morris. 2009. CoSense: enhancing sensemaking for collaborative web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1771–1780. https://doi.org/10.1145/1518701.1518974

[66] Sharoda A Paul and Meredith Ringel Morris. 2011. Sensemaking in collaborative web search. *Human–Computer Interaction* 26, 1-2 (2011), 72–122.

[67] Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611* (2020).

[68] Peter Pirolli and Stuart Card. 1999. Information Foraging. *Psychological Review* 106, 4 (1999), 643–675. https://doi.org/10.1037/0033-295x.106.4.643

[69] Peter Pirolli and Stuart Card. 2005. *The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis.*

[70] Héctor R. Ponce, Richard E. Mayer, M. Soledad Loyola, and Mario J. López. 2020. Study Activities That Foster Generative Learning: Notetaking, Graphic Organizer, and Questioning. *Journal of Educational Computing Research* 58, 2 (2020), 275–296. https://doi.org/10.1177/0735633119865554 arXiv:https://doi.org/10.1177/0735633119865554

[71] Martin Potthast, Matthias Hagen, Michael Völske, and Benno Stein. 2013. Crowdsourcing interaction logs to understand text reuse from the web. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1212–1221.

[72] Guanghui Qin, Yukun Feng, and Benjamin Van Durme. 2022. The NLP task effectiveness of long-range transformers. *arXiv preprint arXiv:2202.07856* (2022).

[73] Zackary Rackauckas. 2024. Rag-Fusion: A New Take on Retrieval Augmented Generation. *International Journal on Natural Language Computing* 13, 1 (Feb. 2024), 37–47. https://doi.org/10.5121/ijnlc.2024.13103

[74] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics* 11 (2023), 1316–1331.

[75] Miguel A. Renom, Baptiste Caramiaux, and Michel Beaudouin-Lafon. 2022. Exploring Technical Reasoning in Digital Tool Use. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>New Orleans</city>, <state>LA</state>, <country>USA</country>, </conf-loc>) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 579, 17 pages. https://doi.org/10.1145/3491102.3501877

[76] Grega Repovš and Alan Baddeley. 2006. The multi-component model of working memory: Explorations in experimental cognitive psychology. *Neuroscience* 139, 1 (2006), 5–21.

[77] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large Language Models Can Be Easily Distracted by Irrelevant Context. arXiv:2302.00093 [cs.CL]

[78] Ben Shneiderman and Richard Mayer. 1979. Syntactic/semantic interactions in programmer behavior: A model and experimental results. *International Journal of Computer & Information Sciences* 8 (1979), 219–238.

[79] Hariharan Subramonyam, Christopher Lawrence Pondoc, Colleen Seifert, Maneesh Agrawala, and Roy Pea. 2023. Bridging the Gulf of Envisioning: Cognitive Design Challenges in LLM Interfaces. https://doi.org/10.48550/arXiv.2309.14459 arXiv:2309.14459 [cs].

[80] Hariharan Subramonyam, Colleen Seifert, Priti Shah, and Eytan Adar. 2020. texSketch: Active Diagramming through Pen-and-Ink Annotations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Honolulu</city>, <state>HI</state>, <country>USA</country>, </conf-loc>) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376155

[81] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling Multilevel Exploration and Sensemaking with Large Language Models. https://doi.org/10.1145/3586183.3606756 arXiv:2305.11483 [cs].

[82] Jason Swarts. 2010. Recycled writing: Assembling actor networks from reusable content. *Journal of Business and Technical Communication* 24, 2 (2010), 127–163.

[83] David Traum. 2017. Computational approaches to dialogue. *The Routledge Handbook of Language and Dialogue* 1 (2017), 143–161.

[84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[85] Qingyue Wang, Liang Ding, Yanan Cao, Zhiliang Tian, Shi Wang, Dacheng Tao, and Li Guo. 2023. Recursively summarizing enables long-term dialogue memory in large language models. *arXiv preprint arXiv:2308.15022* (2023).

[86] Zijie J Wang, Fred Hohman, and Duen Horng Chau. 2023. Wizmap: Scalable interactive visualization for exploring large machine learning embeddings. *arXiv preprint arXiv:2306.09328* (2023).

[87] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* (2023).

[88] Jules White, Sam Hays, Quchen Fu, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design. https://doi.org/10.48550/arXiv.2303.07839 arXiv:2303.07839 [cs].

[89] Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A brief overview of ChatGPT: The history, status quo and

potential future development. *IEEE/CAA Journal of Automatica Sinica* 10, 5 (2023), 1122–1136.

[90] Ziang Xiao, Michelle X Zhou, Q Vera Liao, Gloria Mark, Changyan Chi, Wenxi Chen, and Huahai Yang. 2020. Tell me about yourself: Using an AI-powered chatbot to conduct conversational surveys with open-ended questions. *ACM Transactions on Computer-Human Interaction (TOCHI)* 27, 3 (2020), 1–37.

[91] Frank F. Xu, Bogdan Vasilescu, and Graham Neubig. 2022. In-IDE Code Generation from Natural Language: Promise and Challenges. *ACM Trans. Softw. Eng. Methodol.* 31, 2, Article 29 (mar 2022), 47 pages. https://doi.org/10.1145/3487569

[92] Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567* (2021).

[93] Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. 2022. Long Time No See! Open-Domain Conversation with Long-Term Persona Memory. https://doi.org/10.48550/arXiv.2203.05797 arXiv:2203.05797 [cs].

[94] Ryan Yen, Nicole Sultanum, and Jian Zhao. 2024. To Search or To Gen? Exploring the Synergy between Generative AI and Web Search in Programming. arXiv:2402.00764 [cs.HC]

[95] Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic Evaluation of Attribution by Large Language Models. https://doi.org/10.48550/arXiv.2305.06311 arXiv:2305.06311 [cs].

[96] Amy X Zhang, Lea Verou, and David Karger. 2017. Wikum: Bridging discussion forums and wikis using recursive summarization. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 2082–2096.

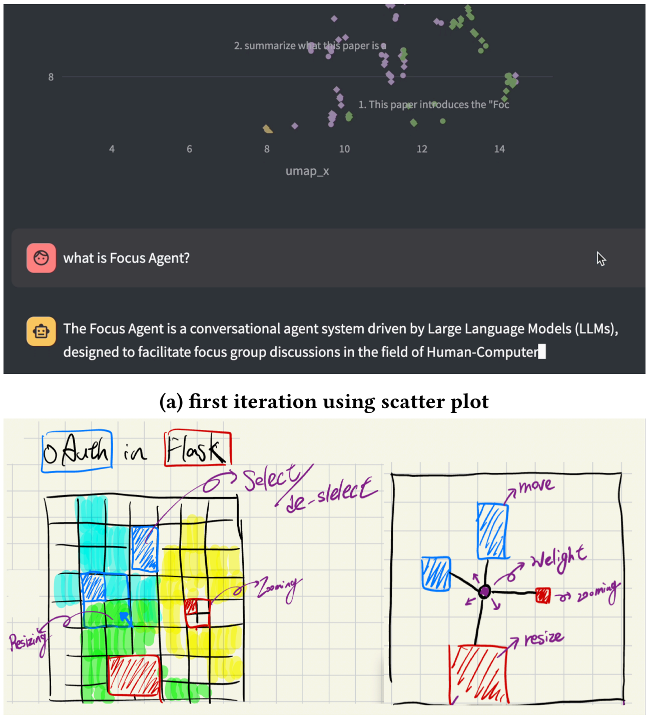
[97] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243* (2018).

[98] Xiangyu Zhao, Longbiao Wang, and Jianwu Dang. 2022. Improving dialogue generation via proactively querying grounded knowledge. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6577–6581.

[99] Hanxun Zhong, Zhicheng Dou, Yutao Zhu, Hongjin Qian, and Ji-Rong Wen. 2022. Less is more: Learning to refine dialogue history for personalized dialogue generation. *arXiv preprint arXiv:2204.08128* (2022).

A APPENDIX

A.1 Design Iterations



what is Focus Agent?

The Focus Agent is a conversational agent system driven by Large Language Models (LLMs), designed to facilitate focus group discussions in the field of Human-Computer

(a) first iteration using scatter plot

(b) low fidelity prototype of the second iteration

Figure 11: Our system design process underwent several iterations based on feedback from four participants.

A.2 System Implementation Detail

A.2.1 Combining Consecutive Pairs of Prompts/Responses. To determine when to merge two consecutive conversations, we define a threshold τ based on the distribution of semantic similarity scores. This threshold governs whether to combine consecutive conversations into a single data point, ensuring that the merging process captures meaningful semantic similarities while avoiding the fusion of unrelated conversations. The equation for calculating the threshold τ is $\tau = \text{percentile}(S(R_i, R_{i+1}), p)$, where $\text{percentile}(S(R_i, R_{i+1}), p)$ denotes the p^{th} percentile of the distribution of semantic similarity scores between consecutive conversations, allowing for a data-driven determination of the threshold τ.

A.2.2 Adapted RAG. The steps involved in our adapted RAG are as follows:

1. Generate Queries: Based on the user's input, multiple related queries are generated to capture various aspects of the context.
2. Retrieve Similar Context: Using a vector similarity search, similar context is retrieved based on the generated queries. This step aims to identify relevant information that can enrich the response generation process.

$$\text{similarity}(\text{q}, \text{d}) = \frac{\text{q\_embedding} \cdot \text{d\_embedding}}{\|\text{q\_embedding}\| \cdot \|\text{d\_embedding}\|}$$

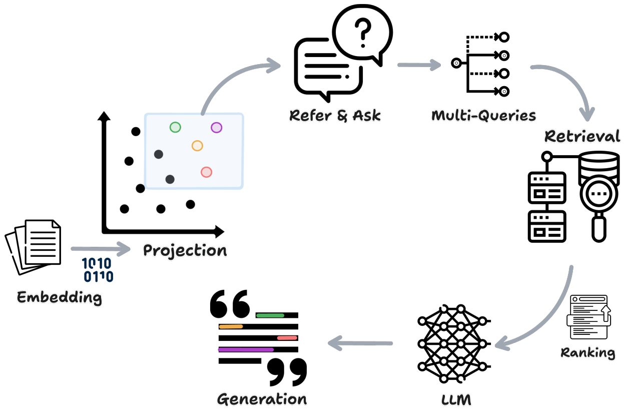


Figure 12: The implementation of adopted RAG for the generation with memories.

where d is the document and q is the user's query.

3. Reciprocal Rank Fusion: The retrieved results are fused using the reciprocal rank fusion algorithm [73]. We aggregate the relevance scores of search results across multiple queries, prioritizing documents that are consistently highly ranked.

$$\text{RRF\_score}(d) = \sum_{q=1}^{Q} \frac{1}{\text{rank}(d, q) + k}$$

- d is the document
- Q is the total number of queries
- $\text{rank}(d, q)$ is the rank of document d in response to query q
- k is a constant to mitigate the effect of small reciprocal ranks

4. Determine Top-k Results: The top-k results are determined based on the fused scores. This step selects a subset of the most relevant context for further processing. It determines the number of top results to select, considering the distribution of scores and identifying a suitable threshold.

$$k = \text{next}\ (i \mid \text{diff} > \text{threshold})$$

- diff represents the differences between consecutive scores
- threshold is the standard deviation of the differences multiplied by 0.8, determining the cutoff point

5. Utilize Retrieved Context for Generation: The selected top-k results are provided as context for the generation model, informing the generation process and ensuring that the generated responses are grounded in relevant information.

A.2.3 Architecture.

A.3 Questionnaire

Below we list the questions we used in the evaluation study questionnaire.

A.3.1 UMUX-LITE.

1. This system's capabilities meet my requirements.
2. This system is easy to use.

A.3.2 NASA-TLX.

1. How mentally demanding was the task?
2. How physically demanding was the task?
3. How hurried or rushed was the pace of the task?
4. How successful were you in accomplishing what you were asked to do?
5. How hard did you have to work to accomplish your level of performance?
6. How insecure, discouraged, irritated, stressed, and annoyed were you?

A.3.3 Self-Defined Likert Scale Items.

1. I had a good understanding of why the system generates such results.
2. I could steer the system toward the task goal.
3. I had more control when managing the output of the AI.
4. I can recall what the memory is about easily.
5. I have a holistic understanding of all my memories.
6. I can see how the AI is using my memories.
7. I am satisfied with the overall suggestions from the system.
8. Finding related memory to reuse is easy.
9. Extracting needed context from memory is easy.
10. Organizing and schematizing memories is easy.
11. Specifying how the memory should be reused is easy.
12. Refinement and iteration of the generated results is easy.
13. Switching between searching memory, providing memory for AI, and chatting required low mental demand

A.4 Study Scenarios and Tasks

A.4.1 ***Scenario 1 (Expository Writing)—Phase One***.

Task 1 (20min). You are provided with four articles talking about education and children's cognitive development, you may copy the content to the chatGPT for reading quickly, and you can also ask GPT to provide more opinions on this topic. You need to write a paragraph (100 words) to demonstrate your understanding of these 4 articles.

- Showing emotional feeling on disparity of education: Strategies in Class Differences in Child Rearing-Are on the Rise
- Importance of Educational Games for Cognitive Development of Children
- Nurturing Creativity & Imagination for Child Development
- Power of Play

Task 2 (20min): Now you are provided with four articles talking about physical exercise and brain development, you may copy the content to the chatGPT for reading quickly, and you can also ask GPT to provide more opinions on this topic. You need to write a paragraph (100 words) to demonstrate your understanding of these 4 articles.

- 10 Benefits of Exercise on The Brain and Body — Why You Need Exercise
- How Exercise Protects Your Brain's Health
- 5 Ways To Improve Your Brain Health and Lower Your Risk of Alzheimer's
- Is exercise actually good for the brain?

A.4.2 ***Scenario 1 (Expository Writing)—Phase Two***.

- Condition A: Now, write a report based on all provided articles on "the effect of physical exercise on education" which should be no less than 5 paragraphs.
- Condition B: Now, write a report based on all provided articles on "the effect of games on children's cognitive development" which should be no less than 5 paragraphs.

A.4.3 ***Scenario 2 (Programming)—Phase One***.

Task 1 (20 min): You are developing a system that enables users to perform semantic searches in a corpus of summaries of ACL by entering search queries. We provide you with several different approaches, and your task is to find the best pipelines for accomplishing this task and provide how to implement these pipelines using Python code.

- Semantic Search
- Build a semantic search engine in Python
- Document Embedding Techniques

Task 2(20 min): You are now trying to find a way to prompt GPT to generate results without hallucinating. There are various processing methods available and you need to discuss and understand them in depth. You need to generate a comparison table that reports the techniques, algorithms/methods, advantages, disadvantages, and how you can roughly implement a simple generation pipeline.

- Advanced Prompt Engineering for Reducing Hallucination
- Retrieval-Augmented Generation (RAG) from basics to advanced
- Advanced Retrieval-Augmented Generation: From Theory to LlamaIndex Implementation
- GPT-4 Enhanced with Real-Time Web Browsing
- ReAct Prompting

A.4.4 ***Scenario 2 (Programming)—Phase Two***.

- Condition A: You now need to build a retrieval enhancement generation pipeline to help programmers solve problems by retrieving through a large code base.
- Condition B: You need to write Python code that enables a user to ask a question about a PDF from the web, the user can type in the question and the system will search the web for relevant PDFs and display the extracted relevant sentences.

A.4.5 ***Scenario 3 (Trip Planning)—Phase One***.

Task 1 (20 min): Engage in a conversation with GPT to gather information about Hawaii, including details about scenes, weather, visa requirements, and more.

Task 1 (20 min): Chat with GPT to arrange accommodation, tourist spots, activities, transportation, and other aspects, and synthesize a table comparing different approaches.

A.4.6 ***Scenario 3 (Trip Planning)—Phase Two***.

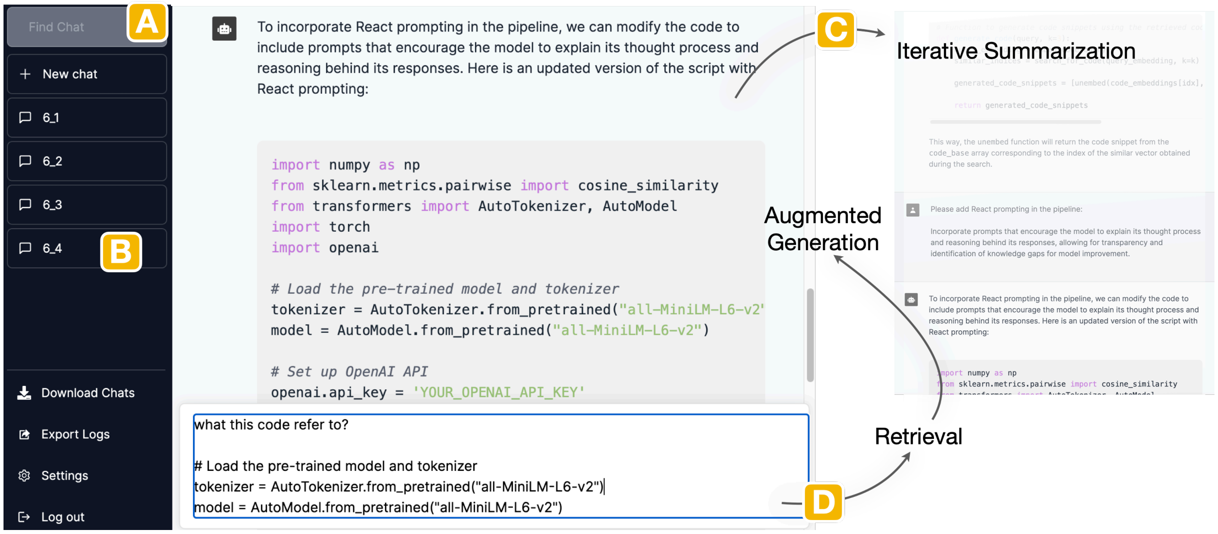
- Condition A: Come up with a five-day travel plan, including the spots plan to visit
- Condition B: Come up with transportation and hotel arrangements for a five-day round trip from Boston to Hawaii

A.5 Demographic Table

Participants included graduate students, research scientists, software engineers, and university students, with reported usage of AI-driven conversational agents for various tasks such as writing emails (4), reports (8), academic papers (8), coding (10), ideation (4), general question answering (12), and information seeking (8).

Gender		Age		Education		ChatGPT Familarity		ChatGPT Usage		Python Experience		Writing Experience	
Men	5	20-29	9	Bachelor	4	Extremely	6	2 times/ week	1	Extremely	3	Extremely	4
Women	7	30-39	3	Master	6	Moderately	4	3 times/ week	1	Moderately	2	Moderately	3
				Doctoral	1	Somewhat	2	4 times/ week	3	Somewhat	3	Somewhat	4
				Professional	1	Slightly		5 times/ week	2	Slightly	1	Slightly	1
						Not at All		7 times/ week	5	Not at All	3	Not at All	0

A.6 Baseline Design



To incorporate React prompting in the pipeline, we can modify the code to include prompts that encourage the model to explain its thought process and reasoning behind its responses. Here is an updated version of the script with React prompting:

```
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from transformers import AutoTokenizer, AutoModel
import torch
import openai

# Load the pre-trained model and tokenizer
tokenizer = AutoTokenizer.from_pretrained("all-MiniLM-L6-v2'
model = AutoModel.from_pretrained("all-MiniLM-L6-v2")

# Set up OpenAI API
openai.api_key = 'YOUR_OPENAI_API_KEY'
```

```
what this code refer to?

# Load the pre-trained model and tokenizer
tokenizer = AutoTokenizer.from_pretrained("all-MiniLM-L6-v2")
model = AutoModel.from_pretrained("all-MiniLM-L6-v2")
```

```
# Function to generate code snippets using the retrieved coc
def generate_code(query, k=3):
    similar_indices = search_for_code(query_embedding, k=k)

    generated_code_snippets = [unembed(code_embeddings[idx],

    return generated_code_snippets
```

This way, the unembed function will return the code snippet from the code_base array corresponding to the index of the similar vector obtained during the search.

Please add React prompting in the pipeline:

Incorporate prompts that encourage the model to explain its thought process and reasoning behind its responses, allowing for transparency and identification of knowledge gaps for model improvement.

To incorporate React prompting in the pipeline, we can modify the code to include prompts that encourage the model to explain its thought process and reasoning behind its responses. Here is an updated version of the script with React prompting:

```
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
```

Figure 13: Our adopted *Baseline*, resembling the current prevalent AI-driven conversational agent ChatGPT, allows users to semantically search related context (A) and switch or create new conversations (B). To ensure a fair comparison with our system, we adopted the same iterative conversational history summarization methods (C) and the retrieval augmented generation approach (D). The generation is powered by the latest large language model, GPT-4 [60].

A.7 Prompt Template

A.7.1 Prompt for Retrieval Augmented Generation.

You are a large language AI assistant helping users to complete tasks or answer questions.

You are given a user question or prompt, and please write a clean, concise and accurate answer to the question or conduct the task. You will be given a set of related contexts to the question or prompt, each starting with a reference number like [[citation:x_x]], where x_x is a referenced number. Please use the context and cite the context at the end of each sentence if related.

Your answer must be correct, accurate and written by an expert using an unbiased and professional tone. Please limit to 1024 tokens. If users prompt for a task, complete the quest asked by users with related citations. and if the given prompt contains a citation, please complete the task based on the context from that citation and cite the context at the end sentence using information from that context.

Please cite the contexts with the reference numbers, in the format [citation:x_x] (for example: 6_8 => [citation: 6_8]). If a sentence comes from multiple contexts, please list all applicable citations, like [citation:3_2][citation:5_1]. Other than code and specific names and citations, your answer must be written in the same language as the question.

Here is the set of contexts: [context]

Remember, don’t blindly repeat the contexts verbatim. And here is the user question or prompt:

A.7.2 modify query with instructions.

You are a large language AI assistant. You are given a user question, and please rewrite the given question based on the instructions. Your modified question must be written in the same language as the user’s question.

The user might provide several instructions to modify the question, such as: (1) ADD_CONTEXT: where all these contexts are needed to be included to answer users’ questions; (2) REMOVE_CONTEXT: where the context should not be included in the answer; (3) HIGHLIGHT_CONTEXT: where the context should be included MORE PROMINENTLY in the answer; (4) OBSCURE_CONTEXT: where the context should be included LESS PROMINENTLY in the answer; (5) GROUP_CONTEXT: where the context should be included in the same sentence and cited together. (6) GENERAL_CONTEXT: cite the context if applicable.

You have to extend the user question based on the given instructions. For example, if the user question is "What are citations 6_1 and 6_2 about?" The instructions are: 1. ADD_CONTEXT: 6_3, 6_4 2. REMOVE_CONTEXT: 6_2 3. HIGHLIGHT_CONTEXT: 6_1 4. GROUP_CONTEXT: 6_1, 6_3; 3_2, 3_1

Reasoning:
The user question is asking context about 6_1 and 6_2
user wants to remove 6_2 and add 6_3 and 6_4
user wants to highlight 6_1
user wants to group 6_1 and 6_3 together
user wants to group 3_2 and 3_1 together

The example output should mention citation by [[citation: x_x]]: “What are citations [[citation: 6_1], [[citation: 6_3]], [[citation: 6_4]] about. Highlight more context from citation [[citation: 6_1]], remove [[citation: 6_2]] and merge the context from [[citation: 6_1]] and [[citation: 6_3]] together. Also, merge the context from [[citation: 3_2]] and [[citation: 3_1]] together.”

Here are the users’ instructions: {instructions}

And here is the user question:

A.7.3 instructed rag prompt.

```
You are a large language AI assistant. You are given a user question, and please write a clean, concise
and accurate answer to the question. You will be given a set of related contexts to the question, each
starting with a reference number like [[citation:x_x]], where x_x is a referenced number. Please use
the context and cite the context at the end of each sentence if applicable.

Your answer must be correct, accurate and written by an expert using an unbiased and professional tone.
Do not give any information that is not related to the question, and do not repeat. Say ïnformation
is missing onf̈ollowed by the related topic, if the given context does not provide sufficient information.

Please cite the contexts with the reference numbers, in the format [citation:x_x]. If a sentence comes
from multiple contexts, please list all applicable citations, like [citation:3_2][citation:5_9]. Other
than code and specific names and citations, your answer must be written in the same language as the
question.

There are six types of contexts with instructions:
(1) ADD_CONTEXT: where all these contexts are needed to be included to answer users' questions;
(2) REMOVE_CONTEXT: where the context should not be included in the answer;
(3) HIGHLIGHT_CONTEXT: where the context should be included MORE PROMINENTLY in the answer;
(4) OBSCURE_CONTEXT: where the context should be included LESS PROMINENTLY in the answer;
(5) GROUP_CONTEXT: where the context should be included in the same sentence and cited together.
(6) GENERAL_CONTEXT: cite the context if applicable.

Please answer the user question strictly based on the given context and the instructions.

ADD_CONTEXT: {add_context}

REMOVE_CONTEXT: {remove_context}

HIGHLIGHT_CONTEXT: {highlight_context}

OBSCURE_CONTEXT: {obscure_context}

GROUP_CONTEXT: {group_context}

GENERA_CONTEXT: {general_context}

Remember, don't blindly repeat the contexts verbatim, CITE the contexts, and DO NOT MISS ANY
INSTRUCTIONS! And here is the user question:
```

A.7.4 summarize prompt. This prompt summarizes all the context within a *Memolet* and also serves as a summary for grouped *Memolet*s during organization.

```
You are a large language AI assistant that describes what are the main points of the given con-
texts. You will be given a set of contexts from past conversations between users and AI, please
reason the usages of each context and aggregate them concisely. Start with: "These memories are re-
lated to the following topics:" and list the topics that are related to the contexts extremely concisely.

Related Contexts: {context}
```

A.7.5 summarize chat history prompt. This prompt summarizes the conversations from 1 to $n - 12$, and subsequently aggregates all new conversations into the initial summarization.

```
Progressively summarize the lines of conversation provided, adding onto the previous summary and returning a new summary.

EXAMPLE
Current summary:
The human asks what the AI thinks of artificial intelligence. The AI thinks artificial intelligence is a force for good.

New lines of conversation:
Human: Why do you think artificial intelligence is a force for good?
AI: Artificial intelligence will help humans reach their full potential.

New summary:
The human asks what the AI thinks of artificial intelligence. The AI thinks artificial intelligence is a force for good because it will help humans reach their full potential.
END OF EXAMPLE

Current summary:
{summary}

New lines of conversation:
{new_lines}

New summary:
```

A.7.6 generate more queries prompt.

```
You are a helpful assistant that helps the user to generate 4 6 search queries based on a single input query, based on the user's original question and your own knowledge. Please identify worthwhile topics that can be follow-ups, and write questions no longer than 20 words each.

Please make sure that specifics, like events, names, and locations, are included in follow-up questions so they can be asked standalone. For example, if the original question asks about "the Manhattan Project", in the follow-up question, do not just say "the project", but use the full name "the Manhattan Project". Your related questions must be in the same language as the original question.

And here is the user query, generate 4 to 6 related queries based on this query:
```