OmniQuery: Contextually Augmenting Captured Multimodal Memory to Enable Personal Question Answering

Jiahao Nick Li
UCLA
Los Angeles, USA
ljhnick@ucla.edu

Zhuohao (Jerry) Zhang
University of Washington
Seattle, USA
zhuohao@uw.edu

Jiaju Ma
Stanford University
Palo Alto, USA
jiajuma@stanford.edu

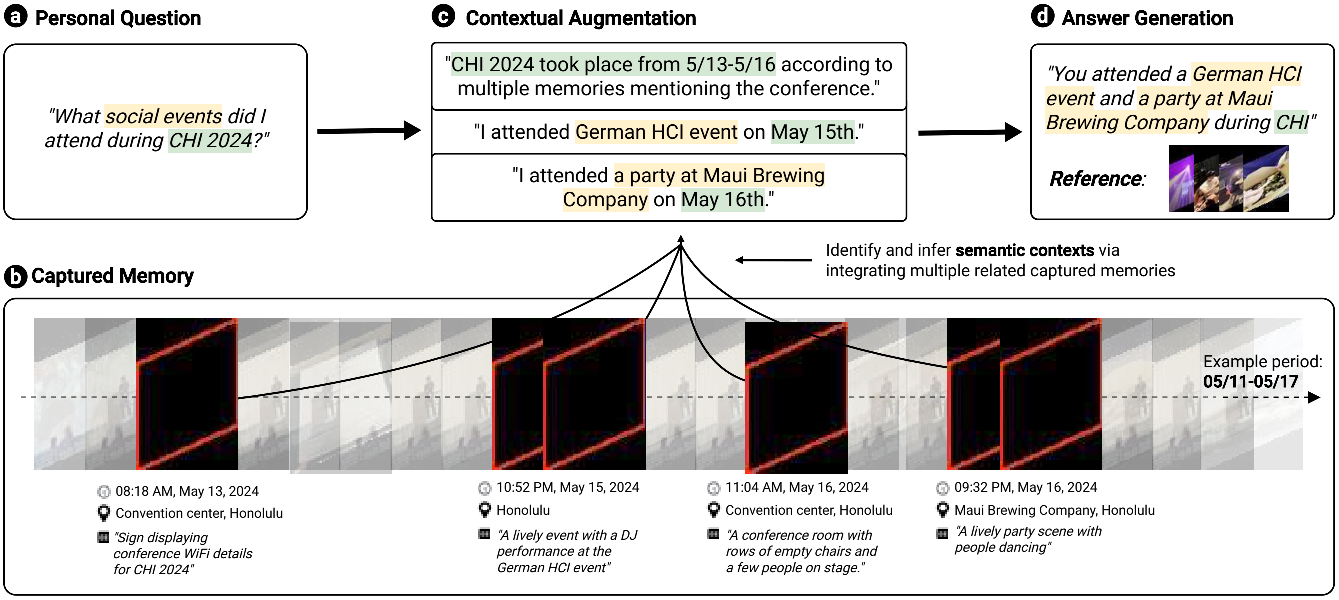

a Personal Question
"What social events did I attend during CHI 2024?"
c Contextual Augmentation
"CHI 2024 took place from 5/13-5/16 according to multiple memories mentioning the conference."
"I attended German HCI event on May 15th."
"I attended a party at Maui Brewing Company on May 16th."
d Answer Generation
"You attended a German HCI event and a party at Maui Brewing Company during CHI"
Reference:
Identify and infer semantic contexts via integrating multiple related captured memories
b Captured Memory
Example period: 05/11-05/17
08:18 AM, May 13, 2024
Convention center, Honolulu
"Sign displaying conference WiFi details for CHI 2024"
10:52 PM, May 15, 2024
Honolulu
"A lively event with a DJ performance at the German HCI event"
11:04 AM, May 16, 2024
Convention center, Honolulu
"A conference room with rows of empty chairs and a few people on stage."
09:32 PM, May 16, 2024
Maui Brewing Company, Honolulu
"A lively party scene with people dancing"

Figure 1: OmniQuery is able to answer complex personal questions (a) on individuals' captured memories (b), such as captured photos, saved screenshots, and recorded videos. It augments the captured memories by identifying and integrating contextual information scattered across multiple interconnected memories (c). OmniQuery then uses this information to retrieve relevant memories and leverages an LLM to generate a comprehensive answer with reference memories (d).

Abstract

People often capture memories through photos, screenshots, and videos. While existing AI-based tools enable querying this data using natural language, they only support retrieving individual pieces of information like certain objects in photos, and struggle with answering more complex queries that involve interpreting interconnected memories like sequential events. We conducted a one-month diary study to collect realistic user queries and generated a taxonomy of necessary contextual information for integrating with captured memories. We then introduce OmniQuery, a novel system that is able to answer complex personal memory-related questions that require extracting and inferring contextual information. OmniQuery augments individual captured memories through integrating scattered contextual information from multiple interconnected memories. Given a question, OmniQuery retrieves relevant augmented memories and uses a large language model (LLM) to generate answers with references. In human evaluations, we show the effectiveness of OmniQuery with an accuracy of 71.5%, outperforming a conventional RAG system by winning or tying for 74.5% of the time.

CCS Concepts

• **Human-centered computing → Human computer interaction (HCI)**; **Natural language interfaces**; User studies.

Keywords

personal memory, contextual augmentation, diary study, multimodal question answering, RAG

ACM Reference Format:
Jiahao Nick Li, Zhuohao (Jerry) Zhang, and Jiaju Ma. 2025. OmniQuery: Contextually Augmenting Captured Multimodal Memory to Enable Personal Question Answering. In *CHI Conference on Human Factors in Computing*

CHI '25, April 26-May 1, 2025, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713448

Systems (CHI '25), April 26-May 1, 2025, Yokohama, Japan. ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3706598.3713448

1 INTRODUCTION

People often record their everyday life by taking photos, screenshots, and videos for saving important information, documenting special occasions, or simply capturing a funny moment [39]. These recorded instances, referred to as **captured memories**, collectively represent subsets of an individual's *episodic memories* [61], a type of long-term memory that contains both specific past experiences and associated contextual details. These episodic memories are essential for answering higher-level memory-related personal questions like "*What social events did I attend during CHI 2024?*" (Figure 1a). Being able to do so could help users reflect on past experiences and make informed decisions in daily tasks.

However, these raw captured memories by themselves are insufficient to answer personal questions, as they lack contextual details that are typically implicit and scattered across multiple pieces of data. As shown in Figure 1b, memories of attending parties during CHI 2024 are not explicitly annotated as occurring during the event. Answering such personal questions requires extracting and integrating contextual information not contained within a single captured instance. For example, by integrating multiple memories that mention "CHI 2024" in their content and extracting their metadata, it is possible to determine when the users attended the conference and connect related social events memories from that period to CHI 2024 (Figure 1c), enabling the answering of the query (Figure 1d).

Advancements in AI have enable question answering (QA) on long documents [4, 63], knowledge graph [30, 68], multimodal databases [13, 59], and egocentric videos [29, 47]. These methods typically rely on data-driven approaches to train powerful models for the target task. However, the private nature of captured memories makes it difficult to curate large datasets, posing challenges for training models specifically for QA on personal data. Recent LLM-based work has adopted retrieval augmented generation (RAG) workflows to handle external databases without specific training [37]. However, such methods depend on explicit connections between queries and relevant external data [17]. In contrast, captured memories are often unstructured and lack contextual annotations, making it difficult to establish explicit links between queries and scattered memories.

To facilitate QA on personal captured memories, we propose OmniQuery, a novel approach designed to robustly and comprehensively answer users' queries on their captured memories. OmniQuery has two key components: *(i)* a question-agnostic pipeline to augment captured memories with contextual information extracted from other related memories to produce *context-augmented memories*, and *(ii)* a natural language QA system that retrieves these processed memories and generates comprehensive answers with referenced captured memories as evidence. The design of OmniQuery is informed by a taxonomy of contextual information that we generated from a one-month diary study with 29 participants. Specifically, we collected and analyzed 299 user queries to identify three types of personal questions (direct content queries, contextual filters, and hybrid queries) and three categories of contextual information (atomic context, composite context, and semantic knowledge). For *(ii)* , OmniQuery employs a retrieval-augmented architecture: given a user input query, it augments the query via a rewriting strategy, retrieves related memories from the augmented data, and generates the final answer with referenced memory instances via an LLM.

To evaluate OmniQuery, we conducted a user evaluation with 10 participants against a generic RAG-based baseline. The participants tested queries both logged during the diary study and generated during the evaluation session on a subset of their own captured memories. For each tested query, participants rated the user perceived correctness and completeness of the answers generated by both systems in a blinded manner. The results show that OmniQuery effectively answers different types of queries on users' personal memories, outperforming the baseline with higher accuracy (71.5%, exceeding the baseline by 27.6%) and winning or tying 74.5% of the time in direct comparisons.

In summary, we contribute:

- A taxonomy of contextual information for augmenting captured memories, derived from queries collected in a one-month diary study with 29 participants.
- A taxonomy-based pipeline of augmenting captured memories that leverages temporal-based reasoning to extract and infer missing contextual information from other related memories.
- The design and implementation of an end-to-end taxonomy-informed system for personal QA[1] .
- A user evaluation of OmniQuery against a baseline system, demonstrating OmniQuery's effectiveness with 71.5% accuracy and outperforming the baseline (winning or tying 74.5% of the time).

2 Related Work

2.1 Personal Memory Augmentation

A large body of work in human-computer interaction (HCI) has explored how to augment users' memories. This includes developing reminder tools for elderlies or people with memory impairments [8, 33, 34, 56], providing proactive support in daily tasks [11, 72], or manipulating users' memory focus in extended reality [5]. These works typically focus on the "capturing" stage of the memory augmentation, where researchers develop wearable devices that continuously capture data using designated sensors, which record various modalities such as videos [16, 26, 48], audios [25, 62], or bio-signals [10], to augment the memory database. For example, recent work such as Memoro developed a wearable, audio-based device that continuously records users' conversations and enables memory suggestions in real-time, either through explicit queries or query-less contextual cues [72]. Differently, OmniQuery focuses on the "post-capturing" stage, utilizing already-existing memory data (e.g., photos and videos users have already captured). It addresses challenges in processing, annotating, and augmenting captured memories with contextual information.

Prior work in natural language processing (NLP), computer vision (CV), and information retrieval (IR) has studied methods of augmenting people's memory. Perhaps the most related is QA on

[1] OmniQuery is open-source at: https://github.com/ljhnick/omniquery

egocentric videos, which are also a form of personal data. Representative tasks include episodic memory retrieval [18, 23, 24], where the system, given a long egocentric video and a query, localizes the answer within the video. However, these datasets differ from the captured data targeted by OmniQuery. The main challenge in egocentric videos is filtering through large, often noisy data, using data-driven approaches to train models for feature extraction. In contrast, captured memories represent a smaller, intentionally collected dataset, where the challenge lies in integrating scattered contextual information across multiple implicitly related memories. Therefore, OmniQuery employs a taxonomy-based method to augment existing data without the need for specific model training, improving QA performance.

2.2 Multimodal Question Answering

Over time, natural language QA research has shifted to more complex settings, including QA across different modalities (e.g., images [2, 22], videos [45, 66, 70], tables [71] or knowledge graph [31, 69]), QA on large datasets [12, 37] and tasks that require multi-hop reasoning [49, 67]. Recent advancements in large language models (LLMs) and multimodal foundation models (e.g., [41–43]) have enabled improved reasoning and answer generation over large, multimodal datasets. This is similar to OmniQuery's use case as answering personal questions requires handling large amounts of captured memories and performing complex reasoning. Prior work has used retrieval-augmented generation (RAG) workflow [37], which retrieves relevant information from external datasets based on a query and then generates output using the retrieved results. For example, MuRAG leverages RAG to answer open questions via retrieving related information from databases of images and text [13]. VideoAgent leverages structured memories processed from long videos to accomplish video understanding tasks [19]. However, these methods rely on datasets already rich in context (e.g., Wikipedia[2]) and improvements are often achieved by designing new query augmentation [9] and retrieval workflows such as Self-RAG [3] and tree-based retrieval [54].

More recently, GraphRAG introduced a data augmentation approach that extracts a knowledge graph from raw data to tackle tasks requiring higher-level understanding, such as query-focused summarization [17]. While we do not explicitly employ a graph data structure in OmniQuery, we adopt GraphRAG's *structured, hierarchical* approach for RAG-based tasks and extend it with taxonomy-based augmentation informed by insights from a diary study to enhance retrieval results on personal captured memories. Finally, when it comes to QA system design, Jim Gray proposed the "20 queries" heuristic that optimizes for answering a core set of questions to address the long tail distribution of potential queries [58]. We adopt the same design principle and replace the specific rules with our contextual information taxonomy.

2.3 Applications Utilizing Contextual Information

Contextual information has long been important in HCI research from early mixed-initiative systems [28] to recent agentic workflows [32]. Over the past few years, there has been a surge in the usage of AI and LLMs in the HCI community to extract contextual information from processing raw multimodal information. For example, Li *et al.* studied how visually impaired people cook and emphasized the importance of conveying contextual information to users through multimodal models [38]. Additionally, Human I/O leverages egocentric perceptions of users and detect situational impairments through reasoning on the multimodal sensing data [44]. GazePointAR develops a context-aware voice assistant to disambiguate users' intent when interacting with real-world information [36]. OmniActions categorizes digital follow-up actions on real-world information and provides proactive action prediction based on perceived context [39]. These system utilized off-the-shelf multimodal models to process raw sensory data and leverage the reasoning capabilities of LLMs to infer the semantic context. OmniQuery builds on this approach by applying these AI techniques to extract and integrate semantic context scattered across various unstructured, raw captured memories. This augmentation enhances users' memory databases, enabling them to answer personal questions about their memories through natural language queries.

3 DIARY STUDY: UNDERSTANDING USER QUERIES

While single captured memory often lacks essential contextual information, OmniQuery proposes to augment such memories by extracting and inferring semantic context from other explicitly or implicitly related memories. To understand how to effectively augment captured memories, we need to answer the following research question:

RQ: What contextual information is essential to integrate with captured memory instances to ensure accurate retrieval in response to user queries?

This question is important as "context" is a broad term, and thus the focus should be on categorizing and identifying the most effective contextual information that enables accurate and meaningful responses to the types of queries users generate when reflecting on past experiences.

3.1 Method

To answer the research question above, we conducted a diary study, a methodology that enables participants to log data whenever need arose [57]. Specifically, we adopted the *snippet-based technique* proposed by Brandt *et al.* [6]. We asked participants to log queries on their past memories only when they had real intent under a genuine context, rather than brainstorming potential questions they might ask to retrieve specific past memories. This approach enabled us to collect spontaneous, authentic queries that users have in real-world scenarios.

We collect the data including: *(i)* the queries participants would use to retrieve or ask about their past memories, *(ii)* the reasons and contexts of these queries (e.g., wanting to show a past experience while chatting with a friend) and *(iii)* (optional) whether they were able to retrieve the corresponding memories from their album, and if so, how they did it (e.g., by scrolling through the photo album).

[2] https://www.wikipedia.org/

3.2 Participants

32 participants (14 male, 17 female, and one non-binary) were initially recruited through an online RSVP form distributed via the X platform[3]. Participants came from North America and Asia. 11 participants reported using Android devices, while the remainder used iOS devices in their daily life. Additionally, 16 participants reported actively logging their daily lives, 13 regularly logged important events and memorable experiences, two logged only essential information, and one seldom logged their lives. While participants were compensated based on their participation ($50 for full participation), they were not required to log a specific number queries each day or over the entire study period. This approach was intentional, as we did not want to require them to generating queries artificially.

3.3 Data Summary

During the diary study, one participant opted out during the first week, and two participants did not log any queries. Of the remaining participants, seven stopped logging after the first week. The rest remained active until the end of the study. As a result, we collected a total of 299 queries. On average, each participant contributed 10.27 queries (SD = 6.09). The highest number of queries from a single participant was 25 and the lowest was 3.

From the collected queries, we identified three types of query: (1) direct content queries (75 queries), (2) context-based filters (28 queries), and (3) hybrid queries (191 queries). The remaining five queries fell outside of these categories as participants attempted non-memory-related tasks like“Mark yesterday pictures as favorites”.

Direct content queries: These queries aim to get direct answers that can be retrieved by searching for memories via description (e.g., “*skateboarding in a tie-dye shirt*”) or rely on information explicitly contained within a single captured instance (e.g., “*What is my driver's license number?*”). This type of query **does not** require additional context not contained in a single captured memory.

Contextual filters: These queries focus on retrieving memories based on specific contexts, such as time, location, or event. For example, a query like “*All the photos in Hawaii*” might only require filtering based on metadata like location. However, for more complex queries such as “*All the photos from my graduation ceremony*”, it **does** require a deeper synthesis of multiple interconnected memories to reconstruct the context surrounding the event.

Hybrid queries: These queries are more complex, combining both direct content queries and contextual filters. For example, a participant asked “*Which meat did I order the last time I came to this Japanese BBQ restaurant?*” Answering such a query typically requires a **multi-hop** process: (1) filter all captured memories under the specific context (e.g., dining in this Japanese restaurant) and (2) analyze the filtered data to generate the final result.

3.4 Analysis

Inspired by the psychological memory theory [61], our data summary indicates that 74.4% of the queries (contextual filters + hybrid queries) require more than just querying the direct content. The complexity in these queries require integration of contextual information in captured memories for accurate processing and filtering. Therefore, we take a step further to build a taxonomy of contextual information in user queries to inform the design of OmniQuery.

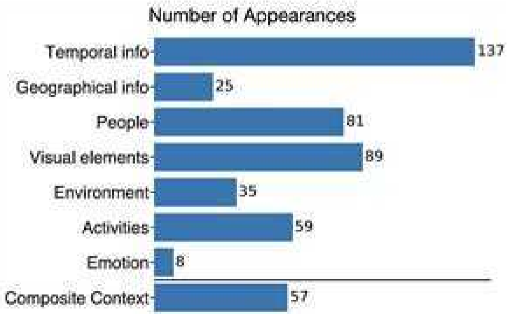


Figure 2: Number of appearances of each types of context (atomic and composite) in the logged queries. Note that a query may contain multiple types of categories, such as “*What boba tea did I drink last week?*”

To identify this essential contextual information, two researchers on the team independently analyzed the logged queries. They coded, filtered, and categorized the types of context required to filter captured memories and better answer the queries. Their results were compared, and discrepancies in categorizations, hierarchy, naming, and granularity were discussed and resolved.

4 TAXONOMY OF CONTEXTUAL INFORMATION

In this section, we present the taxonomy built from analyzing user queries. We identified three key types of contextual information that can be integrated with captured memories: (1) atomic context, (2) composite context, and (3) semantic knowledge.

4.1 Atomic Context

Atomic context refers to contextual information typically obtainable from a single captured memory. This includes data directly from metadata, sensed from visual and auditory content, or inferred from the content itself. Table 1 shows the seven types of atomic contexts categorized from the queries. Among them, temporal information and geographical info can be directly obtained from the memory media's metadata. People and visual elements typically require facial recognition or other vision models for detection. Environment, activity, and emotion are more implicit and require reasoning based on the content (e.g., a photo of a menu may suggest the person is in a restaurant). The number of appearances of each category is shown in Figure 2.

4.2 Composite Context

Composite context is how people remember and refer to past experiences, such as "*Who did I ski with in* ***the lab retreat*** *last year?*" These contexts can range from significant events like a wedding or a conference trip to smaller incidents like hanging out with a friend or a day trip to Seattle. Specifically, composite context is defined as **a combination of multiple atomic contexts**. For example,

[3] https://x.com/

Table 1: Categorization and examples of atomic and composite context

Category	**Definition**	**Exemplar queries** 🟪 refers to contextual cues
	Atomic context	
Temporal info	Specific time period or particular time of the day	"*What boba tea did I drink last week?*" "*What is my routine in the morning?*"
Geographical info	Location data such as city names or venue details	"*How many churches did I visit in Barcelona?*"
People	Individuals present in the captured memories	"*Find the photo of me and my grandpa last year.*"
Visual elements	Other directly sensible elements, including animals, physical objects, or specific visual features	"*My photo with short hair last year.*" "*Photo of my dog when he was a puppy.*"
Environment	Inferred environment based on the content	"*Gym selfies from last year.*"
Activities	Actions or activities inferred from the content	"*How many cardio session did I complete last month?*"
Emotion	Subjective emotion or emotional cues	"*My happiest moment last year*"
	Composite context	
-	Combination of multiple **atomic contexts**	"*Who did I ski with in the lab retreat last year*"

the composite context "lab retreat" encompasses atomic contexts including "February, 2024" (temporal), "Lake Tahoe, California" (geographical), and "hanging out with labmates" (activity).

While atomic context is typically available within a single captured memory, composite context requires integrating multiple memories to understand the connection between them. Since an individual's captured memories are linear on the timeline, memories related to a specific event tend to cluster closely together. We leveraged this **temporal proximity** to identify and extract various composite contexts from the raw captured memories. For a detailed discussion of this approach, please refer to Section 5.2.

4.3 Semantic Knowledge

In psychology theories, semantic knowledge refers to the general world knowledge that humans accumulate over time [50, 61], distinct from episodic memories that are tied to specific experiences and events. Similarly, we can generate semantic knowledge from a user's captured memories, providing broader insights of the user's past experiences. For example, patterns like "Jason has a habit of going to the gym 3-4 times a week" can be inferred from multiple captured memories. Such patterns are helpful in answering queries that not necessarily require specific knowledge such as "*How often do I go to the gym in April?*"

5 OMNIQUERY: AUGMENTING CAPTURED MEMORIES

Informed by the generated taxonomy, OmniQuery employs a **query-agnostic** preprocessing pipeline to augment existing captured memories. The pipeline extracts scattered contextual information from interconnected captured memories, synthesizes it, and augment each memory with the enhanced context. Specifically, the augmentation pipeline involves three steps (as shown in Figure 3): (1) structuring individual captured memories via processing their content and annotating with atomic contexts, (2) identifying and synthesizing composite contexts from multiple captured memories using sliding windows, and (3) inferring semantic knowledge from multiple captured memories and the identified composite contexts.

5.1 Step 1: Structuring Individual Captured Memories

Raw captured memories are often unstructured and lack contextual annotation [55]. In this step, OmniQuery structures each captured memory, making it easier to analyze and extract information. Figure 4 shows an example of structuring an single captured memory, which involves two key parts: (1) processing and understanding the content of the memory and (2) annotating the memory with atomic contexts.

Processing content. Content of a captured memory includes an overall description of the memory as caption, visible text in the image, and transcribed speech (for videos, not shown in Figure 4). Specifically, OmniQuery leverages multimodal models to process and generate image captions, performs optical character recognition (OCR) to recognize visible texts, and uses audio-to-text models to transcribe speech.

Annotating atomic contexts. With the content processed, OmniQuery annotates each captured memory with each type of atomic context. As shown in Figure 4b, OmniQuery extracts the temporal and geographical information from the metadata and uses multimodal models to detect people and other visual elements. Then OmniQuery synthesizes the processed information and infers the environment and activities. For example, based on a photo of a sign displaying conference Wi-Fi details, OmniQuery infers that the user is likely attending a conference (activity) and is at the conference venue (environment). Note that due to the subjective nature

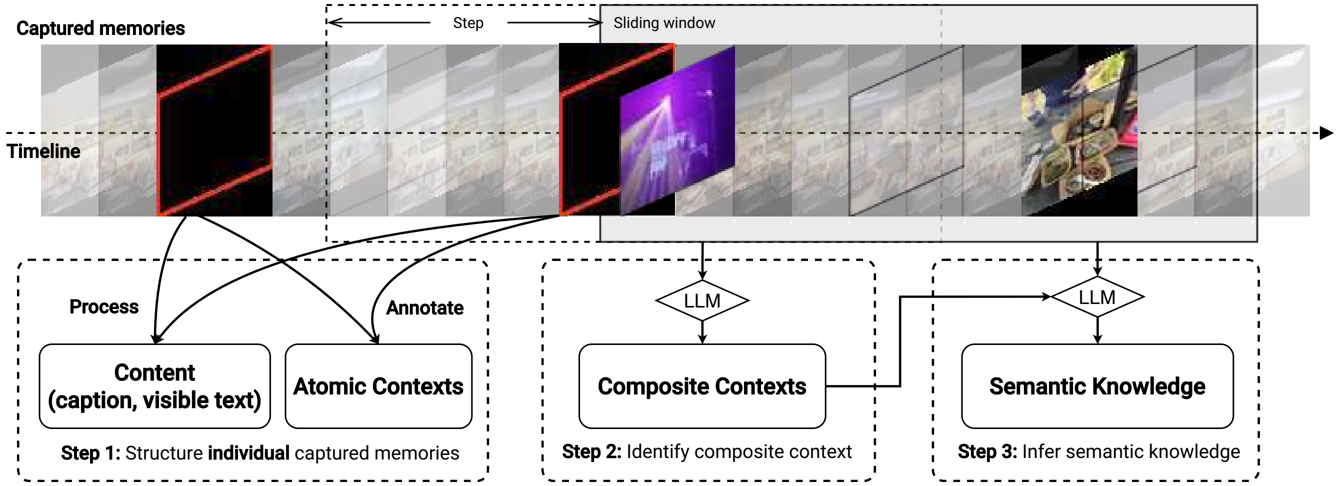

Captured memories
Step
Sliding window
Timeline
Process
Annotate
Content (caption, visible text)
Atomic Contexts
Step 1: Structure individual captured memories
LLM
Composite Contexts
Step 2: Identify composite context
LLM
Semantic Knowledge
Step 3: Infer semantic knowledge

Figure 3: Augmenting captured memories involves three steps: (1) structuring memories by processing content and annotating with atomic contexts; (2) identifying composite context through sliding windows; (3) inferring semantic knowledge from the structured memories and identified contexts.

of emotions that often requires user input, emotion inference is excluded from the current implementation.

5.1.1 Indexing. After each captured memory is structured, it is indexed and stored in a database. Additionally, the annotations in textual format are encoded into text embeddings to enable vector-based search during the retrieval process. In the database, each data entry corresponds to a captured memory with both the original media (e.g., photo, video), its structured annotations in natural language, and text embeddings.

5.2 Step 2: Identifying Composite Context

As captured memories are recorded in a linear manner along a personal timeline, those interconnected through semantic contexts often cluster closely together. For example, memories related to CHI 2024 are likely to occur during the event itself. Taking advantage of this **temporal proximity**, OmniQuery adopts a sliding window approach to analyze potentially interconnected memories scattered in segments for composite context identification.

As shown in Figure 5a, a static window size of seven days is used in our current implementation. The inference is performed via an LLM, in which the input is the structured annotations of these memories and the output is the identified composite contexts along with their start and end dates and the associated captured memories (Figure 5b). To account for cases where composite contexts are split in half, we use a step size (4 days in the current setup) smaller than the window size, allowing for overlap and comprehensive processing. For longer composite contexts (e.g., lasting more than two weeks), each segment of the context is identified separately within the sliding windows and then merged into a single composite context. Additionally, any duplicated composite contexts caused by the overlap between sliding windows are also merged to avoid redundancy (Figure 5c). Note that we determined the window size heuristically. A longer sliding window can better capture extended events or patterns, while it may underperform on shorter contexts due to redundant information. One way to optimize is to make the size dynamic, adjusting it based on the density of activities in a given period. This approach would require a fixed dataset for experiments. We further discuss this in Section 9.1.

Specifically, as opposed to including detailed predefined categories (as with atomic contexts) in the prompt for LLMs, we adopt the few-shot prompting technique [7], providing examples of composite contexts summarized from the collected questions in the prompt. For the detailed prompt, please refer to Appendix A.1.

Explicitly mentioned contexts. Some composite contexts are **explicitly** mentioned in the captured memories. For example, a screenshot of a flyer may reference the upcoming "CHI 2024" event happening next month, or a transcribed conversation might discuss a "Hawaii trip" that took place the previous year. We leverage LLMs' pretrained world knowledge to differentiate between atomic contexts and composite contexts. For example, “a workout session” is identified as an activity (atomic context) because, based on world knowledge, it is more likely to refer to this activity alone. In contrast, “CHI 2024” is recognized as a composite context, as it likely involves multiple interconnected atomic contexts. Such identified composite contexts are either merged with an existing composite context (e.g., if "CHI 2024" has already been identified) or directly added as a new composite context if it is unique.

5.3 Step 3: Inferring Semantic Knowledge

Different from composite contexts, semantic knowledge focuses on high-level general knowledge rather than specific memory details. In the scenarios of personal memory, semantic knowledge refers to personalized knowledge distilled from an individual's past, as opposed to general knowledge (e.g., “the capital of France is Paris.”). For example, if a person's captured memories contain photos of attending CHI 2024, the distilled semantic knowledge might be,

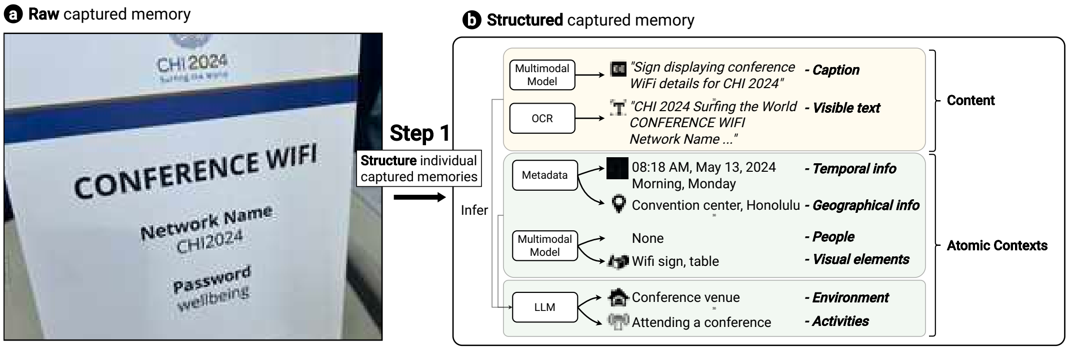


Figure 4: An example of structuring an individual captured memory (a photo of the Wi-Fi details of CHI 2024 conference).

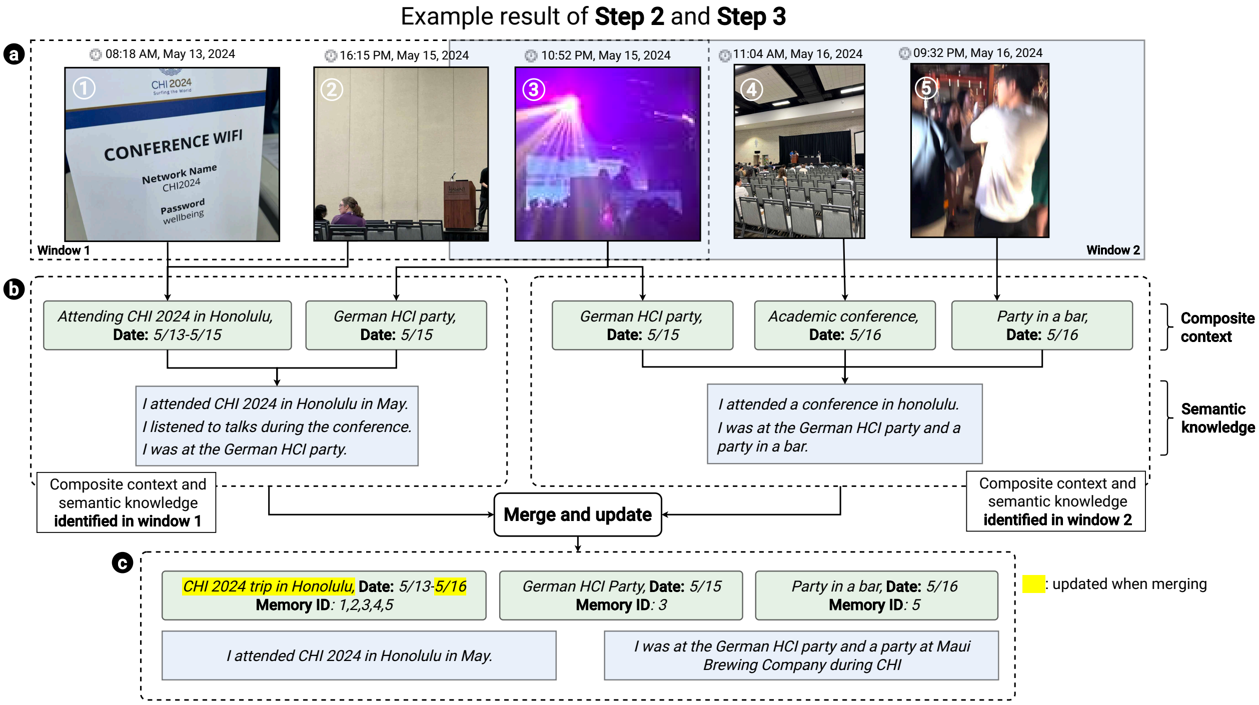

Example result of Step 2 and Step 3
a
08:18 AM, May 13, 2024
16:15 PM, May 15, 2024
10:52 PM, May 15, 2024
11:04 AM, May 16, 2024
09:32 PM, May 16, 2024
1
2
3
4
5
Window 1
Window 2
b
Attending CHI 2024 in Honolulu, Date: 5/13-5/15
German HCI party, Date: 5/15
German HCI party, Date: 5/15
Academic conference, Date: 5/16
Party in a bar, Date: 5/16
Composite context
I attended CHI 2024 in Honolulu in May.
I listened to talks during the conference.
I was at the German HCI party.
I attended a conference in honolulu.
I was at the German HCI party and a party in a bar.
Semantic knowledge
Composite context and semantic knowledge identified in window 1
Merge and update
Composite context and semantic knowledge identified in window 2
c
CHI 2024 trip in Honolulu, Date: 5/13-5/16 Memory ID: 1,2,3,4,5
German HCI Party, Date: 5/15 Memory ID: 3
Party in a bar, Date: 5/16 Memory ID: 5
🟨: updated when merging
I attended CHI 2024 in Honolulu in May.
I was at the German HCI party and a party at Maui Brewing Company during CHI

Figure 5: An example of using sliding windows to identify composite contexts and infer semantic knowledge: (a) two consecutive sliding windows; (b) composite contexts and semantic knowledge generated in each window; (c) merging results of the windows.

"The person attended the CHI conference in Honolulu in 2024." Additionally, semantic knowledge goes beyond summarizing past events. It also encompasses inferred patterns and facts about the individual's behavior or preferences. For example, a chat message mentioning Jason's birthday could infer that "Jason's birthday is on [SPECIFIC DATE]." Similarly, analyzing multiple grocery shopping receipts that consistently include lactose-free milk could lead to the inference that the user is possibly lactose intolerant. We would like to note that, while inspired by human semantic knowledge as defined in psychology [61], the semantic knowledge referenced here is slightly different. In OmniQuery's setting, the semantic knowledge is objective and derived solely from the content of the memories. In contrast, human semantic knowledge is more implicit, containing broader associations and longer-term effects that might go beyond just the observable data.

Semantic knowledge is inferred in each sliding window, while also taking into account the identified composite contexts (in Step 2) to gain higher-level understandings of the user's past and generalized information (Figure 5b bottom). The output is a list of inferred declarative semantic knowledge independent from specific memories. The instructions provided to the model are specifically tailored to guide the inference process toward overarching patterns and trends rather than specific event details. The detailed prompt for identifying semantic knowledge can be also found in Appendix A.2. Each inferred entry of semantic knowledge is either merged with existing entries or added to the knowledge list if new.

5.4 Implementation Details

To deduplicate images and videos as people tend to capture similar content multiple times, we use CLIP [53] to encode images (or the first frame of videos) into embeddings, calculate the similarities between images, and merge those with the similarity above 0.85. We use the Google Cloud Vision API[4] for OCR to detect text in images and OpenAI's Whisper model[5] for audio-to-text conversion. Note that Whipser is known for hallucination when there is no speech in the audio, thus we applied further data cleaning to validate the transcribed result using OpenAI's `GPT-4o-mini`. For other visual processing, We use `GPT-4o` handles multimodal sensing, including identifying people and visual elements in images and generating scene descriptions. For video processing, as a proof-of-concept, we consider only the first 10 seconds of each video, sampling 10 frames to be analyzed by `GPT-4o` for content understanding. Text is encoded into embeddings using OpenAI's `text-embedding-3-small` model. Currently, we utilize a custom vector database and matrix-based similarity search implemented with NumPy in Python. However, for real-world applications, more advanced vector databases (e.g., Pinecone[6]) would be necessary to handle larger volumes of personal data.

6 OMNIQUERY: QUESTION-ANSWERING SYSTEM

With captured memories augmented with contextual information, OmniQuery adopts a RAG architecture for the question answering system. RAG-based systems are effective in handling large datasets and mitigating hallucination issues by retrieving relevant content and grounding the generated results in this retrieved information. This approach ensures that the output is both relevant and accurate, leveraging specific data rather than relying solely on the model's internal knowledge. This approach is chosen because, on average, personal captured memories often exceed 30,000 photos and videos (as reported by participants in our diary study), which exceeds the limit of most foundation models nowadays.

As shown in Figure 6, given an input query, OmniQuery first applies a *taxonomy-based* augmentation of the query by disambiguating and decomposing it into specific contextual elements (Figure 6a). Then, it retrieves the relevant captured memories from the structured captured memories and the composite contexts, along with related knowledge from the list of semantic knowledge (Figure 6b). The retrieved memories and knowledge, along with the augmented query, are then sent to an LLM to generate a comprehensive answer (Figure 6c). We discuss detailed implementations of each step below.

6.1 Taxonomy-Based Query Augmentation

As mentioned in Section 3.3, most user queries tend to be hybrid in nature or require contextual information. This means that directly searching based solely on the content of captured memories often results in an incomplete or insufficient retrieval of relevant memories. To enhance the retrieval process, OmniQuery adopts the query refinement approach [9] to augment the queries. This query augmentation process is also informed by the taxonomy of contextual information and it involves

(1) **Rewriting the query** to declarative format to improve search accuracy of vector-based similarity matching;
(2) **Decomposing the query** to extract necessary contextual filters, such as time, location, or events, which are grounded in the taxonomy. Note that only explicitly mentioned temporal contexts like "... last week" will be recognized temporal filters. Phrases like "... during CHI 2024" are part of a composite context and thus not counted as a temporal filter;
(3) **Inferring potential related contexts** that may not be explicitly mentioned in the query but can enhance the filtering process also grounded in the taxonomy.

For example, as shown in Figure 6d-g, the query "*What social events did I attend during CHI 2024?*" is rewritten into a declarative format of "*The social events I attended during CHI 2024*". We leveraged an LLM to classify the mentioned contexts in the query as either atomic or composite. Detailed prompts are provided in Appendix A.3. Since "CHI 2024" is explicitly mentioned and identified as a composite context, it is extracted and labeled with the appropriate composite context tag. "Social events" is also extracted and identified as an atomic context (activities). Additionally, because "social events" might include various activities like parties, dancing, or casual conversations and involve multiple people, OmniQuery infers the relevant atomic contexts (people and activities) and annotates them in the corresponding context category.

6.2 Retrieving Relevant Augmented Memories

The decomposed augmented query is used to comprehensively retrieve relevant augmented captured memories. The augmented captured memories consist of the structured captured memories (with processed content and annotated atomic contexts), the list of identified composite contexts, and the list of semantic knowledge. OmniQuery uses the decomposed components from the augmented query to perform a multi-source retrieval, pulling related memories from each of these sources. The results are then consolidated into a comprehensive list of relevant memories, which are used to generate an accurate and detailed answer for the user's query.

Declarative query → Semantic knowledge & processed content. The declarative query is encoded into text embeddings to search for both the semantic memories and processed content (caption and visible text) of the captured memories. This initial search step focuses on finding knowledge and memories directly related to the input query, without incorporating additional contextual filters.

Decomposed atomic contexts → Annotated atomic contexts. Each element of the decomposed atomic contexts (both extracted or inferred) is encoded into text embeddings and searched through the corresponding categories in the structured captured memory database. For example, if the query involves activities like "party" and "dancing," OmniQuery searches for captured memories annotated with similar activities. Any memories that have been annotated with related or similar activities will be retrieved, ensuring that relevant memories are included in the results. Additionally, *temporal contexts* apply a **strict** filter, excluding memories outside the

[4] https://cloud.google.com/vision/docs/ocr
[5] https://github.com/openai/whisper
[6] https://www.pinecone.io/

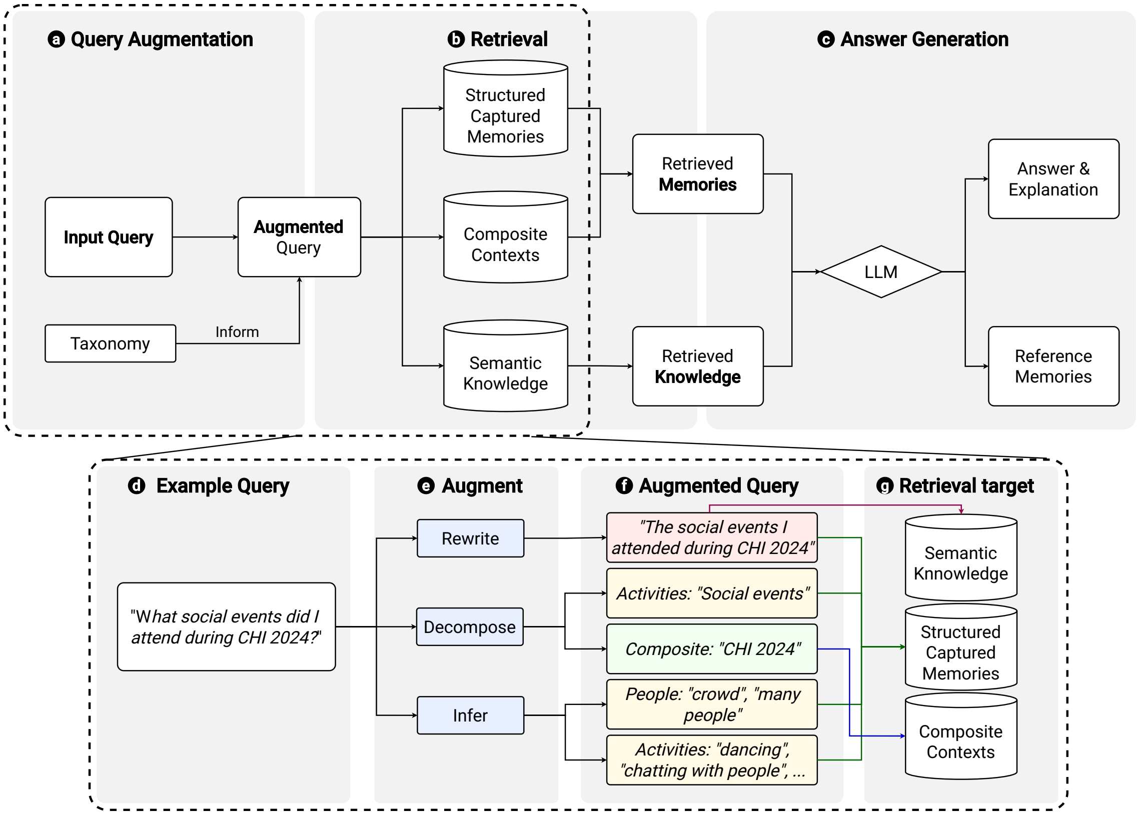


Figure 6: The question-answering system consists of: (a) taxonomy-based query augmentation by decomposing and inferring contextual information; (b) retrieving memories and semantic knowledge; (c) generating answers with an LLM using referenced memories. Specifically, given an input query (d), OmniQuery first augmens it via rewriting and decomposing and inferring contextual information (e). The augmented query with different types of decomposed contextual information (f) is used to retrieve captured memories from different memory and knowledge storage (g).

specified time frame (e.g., "last month") from the retrieval process.

Decomposed composite contexts → Identified composite contexts. Any composite context decomposed in the augmenting process is also encoded into text embeddings and searched through the list of identified composite contexts. All captured memories linked to the semantically similar composite contexts are retrieved. This ensures all memories related to the composite contexts are included. Additionally, OmniQuery leverages an LLM to assess whether a composite context includes temporal constraints. For example, "*... during CHI 2024*" implies a strict temporal filter, while "*photos related to CHI 2024*" does not.

6.3 Answer Generation

The retrieved results is then sent to an LLM to generate the final answer. Specifically, the input for the LLM consists of: (1) the augmented query, (2) the retrieved semantic knowledge from the list, (3) all the retrieved captured memory entries from the annotated database, including both the memory content and its associated contextual annotations.

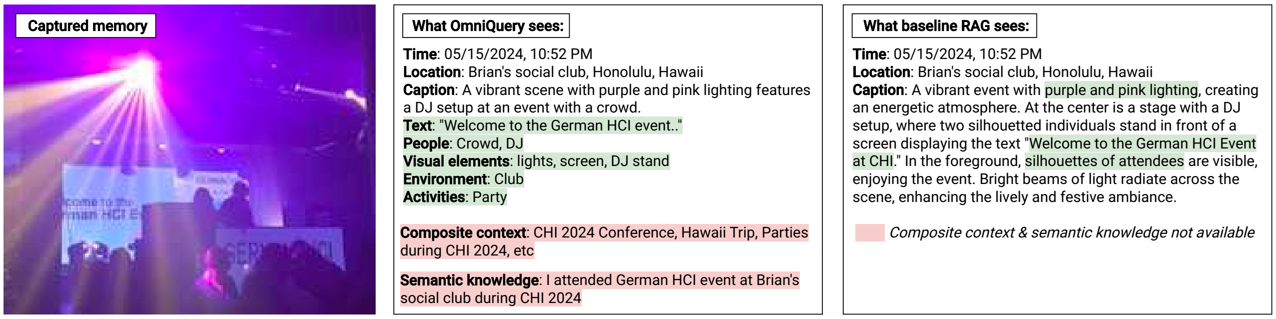
The model analyzes and reasons which captured memories serve as references for the generated answer. These reference memories are also included in the final answer presented to the user. To enhance the reasoning process, OmniQuery leverages chain-of-thought prompting [64], ensuring the generation is more accurate and contextually rich (specific prompts in Appendix A.4).

7 USER EVALUATION

We conducted a user evaluation to test OmniQuery's capabilities in handling real-world personal data by comparing it against a baseline system implemented with a typical RAG structure for question answering. Both systems were deployed on the participants' local machine to protect their personal data. In this section, we discuss the detailed evaluation process, metrics, and results, including quantitative results of the two systems' performances, representative examples, and qualitative feedback.

7.1 Participants

We recruited 10 participants, including seven from our diary study and three additional participants via word-of-mouth. They consented to the whole process, including that their filtered personal



What OmniQuery sees:

Time: 05/15/2024, 10:52 PM
Location: Brian's social club, Honolulu, Hawaii
Caption: A vibrant scene with purple and pink lighting features a DJ setup at an event with a crowd.
Text: "Welcome to the German HCI event.."
People: Crowd, DJ
Visual elements: lights, screen, DJ stand
Environment: Club
Activities: Party

Composite context: CHI 2024 Conference, Hawaii Trip, Parties during CHI 2024, etc

Semantic knowledge: I attended German HCI event at Brian's social club during CHI 2024

What baseline RAG sees:

Time: 05/15/2024, 10:52 PM
Location: Brian's social club, Honolulu, Hawaii
Caption: A vibrant event with purple and pink lighting, creating an energetic atmosphere. At the center is a stage with a DJ setup, where two silhouetted individuals stand in front of a screen displaying the text "Welcome to the German HCI Event at CHI." In the foreground, silhouettes of attendees are visible, enjoying the event. Bright beams of light radiate across the scene, enhancing the lively and festive ambiance.

🟥 *Composite context & semantic knowledge not available*

Figure 7: For each captured memory, the baseline RAG system lacks the contextual information extracted through the taxonomy-based augmentation used in OmniQuery.

data will be processed via an API service. All 10 participants four male, six female, age range 22 - 29, $\bar{x}$ = 25.3, SD = 2.63) were fluent or native English speakers. The participants rated their frequency of logging their daily lives as 'Only record essential information' (1), 'Regularly log important events and memorable experiences' (5) and 'Actively log my daily life' (4). Each participant was compensated with $50 for completing this study.

7.2 Apparatus

Two different systems were implemented in the user evaluation: the OmniQuery pipeline and a baseline system for comparison. The baseline RAG is designed to differ only in OmniQuery's core contributions. As shown in Figure 7, while the baseline RAG includes basic contextual information for each captured memory, such as time, location, and a detailed description of the scene, the key differences are: (1) The contextual information remains "raw" and is not structured using the taxonomy derived in OmniQuery, and (2) it lacks composite context and semantic knowledge, as it does not extract contextual information from multiple related memories. Additionally, in the question-answering phase, the baseline does not leverage taxonomy-based retrieval. All other components, such as the base LLM and prompt structure, remain the same. This design ensures the delta between OmniQuery and the baseline highlights and evaluates OmniQuery's core contributions: taxonomy-based augmentation and retrieval. For the detailed implementation of the baseline, please refer to Appendix B.

As shown in Figure 8, in the studies, participants were presented with a single text input box similar to search engine input boxes. After they typed in the question, they would see two answers generated by the two systems in a randomized order. Two rating questions on the answer's accuracy and completeness were then shown under each answer with a scale of 1-5.

7.3 Procedure

The user evaluation has three stages: (1) system setup on participants' local machines, (2) personal data preparation, and (3) the main testing session.

System setup. The participants were given the source code for OmniQuery to install the back-end and a web application on their local machine. They had the option of an online walkthrough session with the experimenters or following the setup instructions on a self-guided manner.

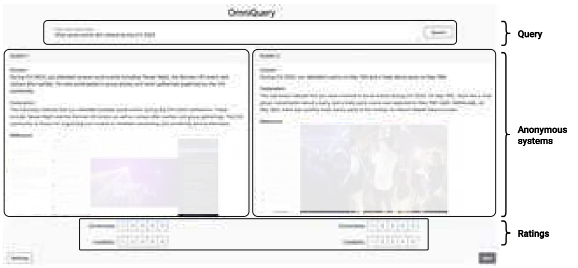


Figure 8: User interface used in the user evaluation.

Personal data preparation. To test OmniQuery on local machines, the participants were asked to transfer a set of captured memories (both photos and videos) from their smartphones' albums to their laptops. To further protect participants' privacy, they were instructed to manually review and filter out any content deemed sensitive or preferred to be excluded from the study. This process needs to meet two key requirements: (1) the manual filtering should not become an excessive burden for participants, and (2) the transferred data should be sufficiently large to simulate real-world usage, containing diverse contexts of distant memories. It is impractical to include all participants' memories, which average 13630 files as collected from the diary study. Thus we perform the following calculation to balance the two requirements:

$$T \geq \frac{C}{K} \times \frac{1+R}{1+F \times R}$$

Where T is the total number of files, C the context limit, K the token cost per frame, R the relevant frame ratio, and F the average frames per video. The equation balances manual filtering and data diversity. The rationale behind the equation is that processing all memories at each query should exceed the limit of existing powerful models, and thus motivating the need for accurate retrieval to answer queries. Specifically, the model OmniQuery uses has a context window of `128K`[7], and the minimum token cost for processing each frame when employing low-fidelity understanding

[7] https://platform.openai.com/docs/models#gpt-4o

Table 2: Quantitative Results of OmniQuery and Baseline, including UPA, UPC, and Accuracy (%)

	OmniQuery			Baseline		
Metrics	UPA	UPC	ACC	UPA	UPC	ACC
Direct content query (24)	4.42	4.13	83.3	3.67	3.46	62.5
Contextual filter + Hybrid (113)	3.89	3.85	69.0	2.93	2.83	38.9
Queries from diary study (28)	3.93	4.14	67.9	2.39	2.61	25.0
All (137)	**3.98**	**3.90**	**71.5**	3.06	2.94	43.1

*ACC refers to accuracy, which is considered accurate when UPA ≥ 4 (mostly correct).

Table 3: Direct comparison between OmniQuery and baseline

	Comparison Win Rate (%)			
Winner	**OmniQuery**	Baseline	Tie	Both are bad
Direct content query (24)	50.0	8.3	33.3	8.3
Contextual filter + Hybrid (113)	53.1	11.5	19.5	15.9
Queries from diary study (28)	60.1	7.1	14.3	17.9
All (137)	**52.6**	10.9	21.9	14.6

is 85[8]. *Ratio* represents the video-to-image ratio, which averages 0.12 (SD = 0.11) based on our diary study survey. As discussed in Sec. 5.4, the number of frames sampled per video during processing is 10. This calculation results in a total number of files of ~767. To expand the memory coverage within the participants' manual filtering capacity, we round this up to 1,000 files.

Depending on how frequently participants logged their daily lives, the memory coverage of the 1,000 selected photos and videos spanned from *one to four months* ($\bar{x}$ = 2.3, SD = 1.03). While this coverage is smaller than what a smartphone album can cover in real-world scenarios (several years), it still represents a moderately distant memory range that is not too recent to demonstrate the capabilities of OmniQuery. These captured memories were then fed into OmniQuery for the taxonomy-based query-agnostic augmentation process. The safety of the process data is ensured following the API's privacy protocol[9].

Main session. The main session lasts 45 minutes, in which the participants tested OmniQuery using two types of questions: (1) questions logged during the diary study and (2) questions they generated during the session. The participants checked on the diary study questions manually to determine if they could be answered using the filtered set of data on captured memories. Additionally, the participants were encouraged to brainstorm and use new questions that were potentially answerable using the filtered data to comprehensively test OmniQuery. Note that participants have no access to the contextual information augmented by OmniQuery. Instead, they follow a simple mental workflow: attempting to recall, asking the system, and verifying the answer.

In the question-answering procedure, OmniQuery-generated answers are accompanied with answers generated from the baseline system implemented using RAG. Each system generated answers anonymously, and the participants compared and rated the results for both systems. The answers and user ratings were recorded for quantitative analysis. Throughout the process, the participants were asked to think aloud [52], and a brief interview was conducted at the end of the session to gather feedback and suggestions. These results were recorded for qualitative analysis.

7.4 Comparison Metrics

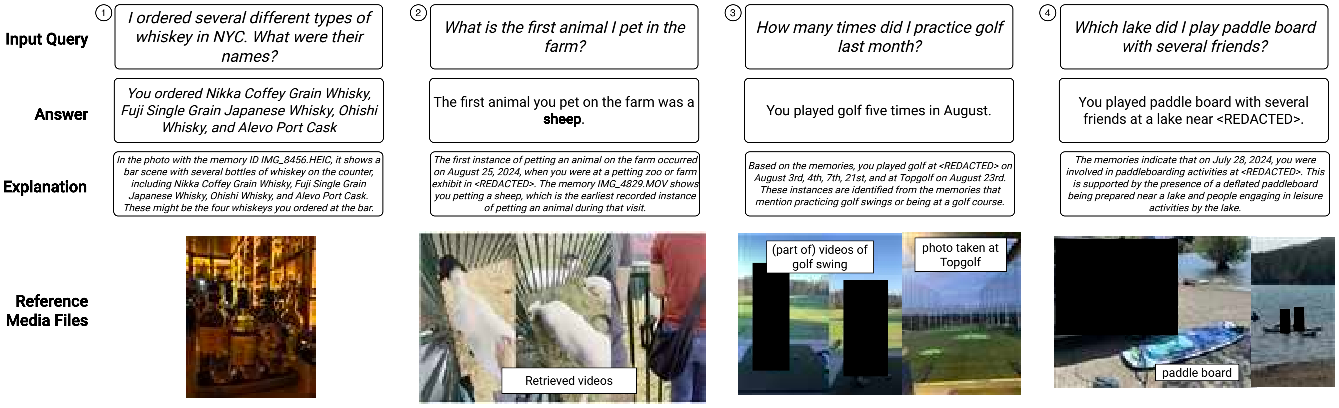
After two answers were presented for a question, the participants were asked to rate the two answers. We used the Chatbot Arena evaluation method [14], where each participant compared answers from the two systems and selected the better one or marked it as a tie. More specifically, for each question, each participant rated the **user perceived accuracy** (UPA) and **user perceived completeness** (UPC) of the answers from both systems.

The UPA score was rated on a scale from 1 to 5: **1**: Completely wrong or invalid result; **2**: Incorrect, but provides at least some insight that helps answer or further refine the question; **3**: Partially correct, or contains a subset of correct answers (e.g., only listing one meal when asked about all meals eaten last week); **4**: Mostly correct, but missing some minor details (e.g., missing one subway trip when asked how many times I rode the subway); and **5**: Completely correct. The UPC score focused on the completeness and credibility of answers, essential for assessing reliability. Many participant queries were complex, requiring explanations supported by captured memory data. For example, when asked "How many meals did I have during my last New York trip?" a system might get the number right but miscount specific meals. Participants reviewed filtered data and rated answers based on their perceived completeness and credibility.

We also directly compare OmniQuery and the baseline to analyze performance. If both have a UPA of 2 or lower, the result is labeled "both are bad". If at least one scores ≥3, the system with the higher UPA wins. In case of a tie on UPA, the system with the higher UPC wins; otherwise, it remains a tie.

[8] https://platform.openai.com/docs/guides/vision

[9] https://openai.com/policies/privacy-policy/



	1	2	3	4
Input Query	*I ordered several different types of whiskey in NYC. What were their names?*	*What is the first animal I pet in the farm?*	*How many times did I practice golf last month?*	*Which lake did I play paddle board with several friends?*
Answer	*You ordered Nikka Coffey Grain Whisky, Fuji Single Grain Japanese Whisky, Ohishi Whisky, and Alevo Port Cask*	The first animal you pet on the farm was a **sheep**.	You played golf five times in August.	You played paddle board with several friends at a lake near <REDACTED>.
Explanation	*In the photo with the memory ID IMG_8456.HEIC, it shows a bar scene with several bottles of whiskey on the counter, including Nikka Coffey Grain Whisky, Fuji Single Grain Japanese Whisky, Ohishi Whisky, and Alevo Port Cask. These might be the four whiskeys you ordered at the bar.*	*The first instance of petting an animal on the farm occurred on August 25, 2024, when you were at a petting zoo or farm exhibit in <REDACTED>. The memory IMG_4829.MOV shows you petting a sheep, which is the earliest recorded instance of petting an animal during that visit.*	*Based on the memories, you played golf at <REDACTED> on August 3rd, 4th, 7th, 21st, and at Topgolf on August 23rd. These instances are identified from the memories that mention practicing golf swings or being at a golf course.*	*The memories indicate that on July 28, 2024, you were involved in paddleboarding activities at <REDACTED>. This is supported by the presence of a deflated paddleboard being prepared near a lake and people engaging in leisure activities by the lake.*

Figure 9: Four representative examples of using OmniQuery to answer hybrid personal memory-related questions.

8 Results

We report the results of our user evaluation by presenting quantitative results of comparing OmniQuery with the baseline, representative examples from the study, and insights gathered from the think-aloud protocol and exit interviews.

8.1 Quantitative Result

The participants tested 137 queries in total during the main session. Among them, 28 were previously logged during the diary study. We manually labeled each tested query using the categorization and definition mentioned in section 3.3. As a result, 24 were categorized as *direct content query* while 17 were *contextual filters* and 96 were *hybrid queries*. We analyzed the performance metrics of both systems (OmniQuery and baseline) using the scores rated by the participants. Table 2 and 3 summarize our results. In addition to presenting the average UPA and UPC scores, we calculated binary accuracy to evaluate whether the systems provided mostly correct answers. An answer was considered accurate if its UPA score was equal to or greater than 4 (mostly correct) (Table 2). We also present the "comparison result" in Table 3, which compares the two systems head-to-head on answering personal questions.

The result shows that, overall, OmniQuery outperforms the baseline system in both the accuracy and completeness. Specifically, OmniQuery achieves an accuracy of 71.5%, outperforming the baseline by 28.4%, winning the comparison 52.6% of the time, and tying 21.9% of the time. For 14.6% of the time, both results are bad. We also present the results for different categories of queries. The results indicate that simpler techniques like the baseline handle direct content queries reasonably well (62.5 % accuracy, and winning or tying 41.6% of the time). While the baseline struggles with more complex queries such as contextual filters or hybrid queries (38.9% accuracy, winning or tying 31.0% of the time), OmniQuery demonstrates it capabilities in effectively handling such queries (69.0% accuracy, winning or tying 72.6% of the time). Specifically, for the queries logged during the diary study, OmniQuery achieved results similar to its overall performance (67.9% accuracy, and winning or typing 74.4% of the time).

8.2 Representative Examples

We selected four representative examples tested by the participants in the evaluation, which are illustrated in Figure 9.

(1) P3 wanted to recall the name of the whiskey they tasted during their trip, as they enjoyed it and wanted to check the price. OmniQuery successfully retrieved the target memory (a photo of the four bottles) and generated a specific answer about the bottles they might have ordered.
(2) P6 wanted to organize the footage from their visit to a farm and asked about the first animal they petted there. OmniQuery accurately retrieved the memories related to the composite context ("visit to the far") and used the temporal order to generate the correct answer: the first animal petted was a sheep.
(3) P1 wanted to estimate how many times they had practiced golf in the past month to track their progress. OmniQuery successfully retrieved all relevant memories, including videos of golf swings at the driving range and photos taken at Topgolf, and accurately generated the answer.
(4) P9 wanted to recall the name of the lake where they went paddling with friends. While no direct memory of paddling is captured, there are several related photos available, including a paddleboard being pumped next to a lake. OmniQuery successfully retrieved these memories and generated the answer using the metadata associated with them.

8.3 Qualitative Feedback and Findings

All participants had experience using smartphone album search features, primarily for retrieving specific information like driver's licenses or events such as trips, aligning with direct content queries and contextual filters (Section 3.3). However, they noted that existing tools are limited to finding specific objects and cannot handle more complex queries. Plus, some of our participants also anticipated for this to happen because they "know what can be searched and what cannot be searched" from these existing album search tools (P2).

In the studies, a lot more challenging questions were asked. For *direct content queries*, it would be challenging to answer when the object is ambiguous or when the users can only describe the object and do not know its exact name. For *contextual filter* and *hybrid* or even more open-ended and subjective questions, existing searching tools are not comparable to OmniQuery and the baseline at all because tools like iOS album search only return specific photos and videos without contextual understanding or filtering.

Here we further summarize the cases that are hard to be accomplished by existing tools. In comparison to the high-level question types provided in Section 3.3, we dive deeper into what these questions in the study were about and provide detailed examples.

- **Exploratory Search**: When users know some characteristics of what they are searching for but cannot specify the exact object. For instance, P1 asked, "What churches did I visit in Barcelona?"
- **Look up and Locate**: When users know specific references or attributes about an item, such as date, location, or a person in the photo, and want to quickly locate the relevant media, such as "Can you find the photo of me on a flyer on Instagram? (P4)"
- **Summarization Tasks**: Participants often need answers that summarize their collection of media, rather than finding a single item. For instance, P7 queried, "Which subway stations in New York have art installations?".
- **Comparative Questions**: Users sometimes want to compare different sets of media. For example, P10 asked, "Am I enjoying beach time more or hiking more?"
- **Open-ended and Subjective Questions**: Participants also asked questions that require interpretation or subjective judgment, which were even more challenging for existing tools. For example, P5 asked, "Given the photos I took, could you analyze what kind of person I am?"

In the meantime, we want to emphasize again that the comparison between OmniQuery and existing tools is conceptual, given that they serve different purposes and are designed differently in retrieving objects or answering questions. We provide this conceptual comparison to demonstrate the variety of questions OmniQuery can support answering.

8.4 Failure Cases

Among the 137 queries tested, 25 failed due to ambiguity, missing contextual cues, information loss, retrieval redundancy, subjectivity, or unavailable memories. Please refer to Appendix C for detailed discussions.

9 DISCUSSION

In this section, we draw on implications from our studies to both discuss limitations and propose future work.

9.1 Curating Fixed Dataset for Benchmarking

Benchmarking is a widely used approach to evaluate whether new systems and algorithms achieve state-of-the-art performance on specific tasks. It also enables ablation studies to assess the effectiveness of individual components of the proposed design. Currently, OmniQuery is evaluated as an integrated system to compare against another system to demonstrate the effectiveness of its overall design. As a future direction, curating a fixed benchmark dataset would allow for more granular evaluation of OmniQuery's performance on personal question answering over multimodal captured memories. This would enable deeper insights into how different design choices (both high-level and low-level) affect task performance through objective ratings. Additionally, it could enable testing multiple system parameters (e.g., sliding window sizes, top-K values for retrieval, or prompt designs for inferring contexts) to achieve the optimal performance. Furthermore, it would also allow ablation studies to assess the impact of individual components (e.g., query augmentation). We further discuss significant challenges of curating a fixed benchmark dataset for this personal data task in Appendix D.

9.2 From Chat Interface to Multimodal Interactions

As a system designed to answer user queries on their personal captured memory, OmniQuery is currently designed in an ask-and-react manner to evaluate its efficacy in a lab-study setting. In our studies, the participants were excited about what OmniQuery was capable of and gave feedback on having more multimodal interactions rather than just a chat interface. We recognize the potential of a more interactive OmniQuery in the following ways:

Multimodal Input and Output. OmniQuery could support multimodal inputs, including audio, images, and videos, to address limitations of text-based search. Many challenging cases for existing album search tools could benefit from this, such as locating an oddly shaped cup or matching dresses by color. Beyond retrieval, OmniQuery could help users relive memories by visualizing captured data, enabling interactive exploration and annotation edits. This brings us closer to a "mind palace" style AI assistant.

Error correction. In our studies, we observed the importance of enabling users to review and refine identified composite contexts and semantic knowledge. Participants expressed the need to correct errors when the system retrieved irrelevant information. For example, P9 asked about a K-pop store, but the system mistakenly included an Instagram screenshot of a Korean TV show. To address this, we propose integrating error correction mechanisms with explanatory insights, confidence levels, and a verification loop, allowing users to mark errors, refine results, and enhance system accuracy over time.

Follow-up queries. A key theme in our study is participants' need to refine queries or ask follow-up questions, with six out of ten mentioning the desire to clarify responses or narrow their searches iteratively. This was particularly relevant when errors were perceived, as discussed in Error Correction. To address this, we propose augmenting follow-up interactions with explanations and confidence levels to highlight uncertainties. A top-K retrieval strategy could also provide ranked answers for ambiguous queries, enabling iterative refinement. Future work could evaluate these approaches through a longitudinal study.

9.3 Enriching Memory Data and Visual Intelligence

At present, OmniQuery primarily processes media from a smartphone's photo album as its main source for captured memories. However, these media alone provide a limited view of a user's broader personal knowledge. For example, in one of the study's failure cases, OmniQuery struggled to infer personal relationships from social interactions captured in group photos. To enhance memory augmentation and improve retrieval accuracy, expanding OmniQuery's data sources and visual intelligence is essential.

Integrating additional data structure and sources. Personal knowledge extends beyond photo albums and exists across various applications. While our participants' photo albums included screenshots of emails, calendar events, and chat histories, these represent only a fraction of the broader personal information available in other communication and social interaction apps. Incorporating data from such sources could significantly enhance OmniQuery's contextual understanding, allowing for more complex queries and richer memory retrieval. While OmniQuery already extracts hierarchically structured information from raw photo album data in the form of atomic and composite contexts, future work can explore using other explicit data structures such as graphs and trees to organize data from multiple sources. Integrating these additional data sources also presents substantial privacy and ethical challenges. While our evaluations were conducted entirely on users' local machines and did not explore privacy-preserving implementations in detail, existing research efforts, such as those focused on differential privacy and on-device machine learning, offer promising directions for secure and privacy-aware deployment. Additionally, commercial tools like Apple Intelligence's private cloud computing serve as examples of ongoing progress in protecting user data while enabling advanced memory retrieval.

Enhancing visual intelligence. Queries related to social interactions remain challenging due to the current lack of advanced features like facial recognition for person identification. Future iterations of OmniQuery could integrate such capabilities (with appropriate user consent), enabling the system to track individuals across various memories. This enhancement would support new use cases, such as monitoring social patterns or tracking progress over time, significantly improving the system's capacity for memory augmentation and retrieval. Additionally, we propose exploring the design and implementation of a comprehensive taxonomy of personal knowledge domains. This would allow users to selectively activate specific domains, such as enabling "Social Interactions and Relationships" to infer personal connections while disabling "Personally Identifiable Information" to prevent the system from processing sensitive data like IDs or SSNs in photos. This modular approach could enhance both user control and privacy.

Augmenting with future AR technologies. A limitation of personal memory capture is the potential for missed moments when users either forget or are unable to document an experience. As AR technology advances, OmniQuery's memory augmentation and retrieval capabilities could be seamlessly integrated into AR systems, allowing for more passive and context-aware memory capture. AR devices could leverage real-time contextual triggers [39] to proactively surface relevant memories or information, offering proactive assistance in pervasive AR environments. This integration would enhance the user experience by making memory retrieval more intuitive and contextually relevant. However, such passive data capture raises even more significant privacy concerns, which will require future research into secure, privacy-preserving implementations to ensure the responsible use of AI in these settings.

9.4 Preserving Privacy

As discussed above, protecting users' privacy is crucial in developing future personal AI assistants, including but not limited to handling personal data such as media in albums and chat and browsing histories. Users have limited control over how their data is handled and must rely on service providers' adherence to privacy protocols. In this subsection, we take a step further to discuss more robust and rigorous measures that should be adopted in real-world settings, where the immense amount of personal data makes approaches like manual filtering in OmniQuery's evaluation infeasible.

One way is to incorporate more advanced data protection techniques, such as data anonymization [46] and encryption [51], while preserving the computational capabilities of large models via online computing. The other approach is leveraging on-device computing, where all data processing occurs locally on the user's device, ensuring full control over users' own data. Recent advances in model compression [27] have made it possible to run large model on smaller devices like smartphones. As OmniQuery is designed to be model-agnostic, it is able to work with different model sizes. While smaller, compressed on-device model may result in reduced performance, future work should focus on developing curated datasets and benchmarks to rigorously evaluate OmniQuery 's performance across different model sizes (e.g., LLaMAs [60] and Phi-3 [1]). This would provide a deeper understanding of how model size impacts privacy and system effectiveness.

10 CONCLUSION

We present OmniQuery, a pipeline that enhances personal question answering on captured multimodal memories. Informed by an one-month diary study, OmniQuery's design responds to real-world user queries and synthesizes a contextual taxonomy of captured memories. Our pipeline design of structuring individual captured memories and identifying composite context and semantic knowledge using a sliding window technique was used to develop a question-answering system, which outperformed a baseline RAG system in both perceived accuracy and completeness. Unlike existing research and commercial tools focused on intelligent image retrieval in smartphone albums, OmniQuery is the first to tackle complex and nuanced personal queries, moving beyond simple object or information piece retrieval. With further attention to privacy-preserving measures, we believe OmniQuery holds significant potential to evolve into a comprehensive multimodal interactive memory assistant, empowering users to revisit, engage with, and manage their personal memories with greater depth and control.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel

Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219 [cs.CL] https://arxiv.org/abs/2404.14219

[2] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2015. VQA: Visual Question Answering. *International Journal of Computer Vision* 123 (2015), 4 – 31. https://api.semanticscholar.org/CorpusID:3180429

[3] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. arXiv:2310.11511 [cs.CL] https://arxiv.org/abs/2310.11511

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[5] Elise Bonnail, Wen-Jie Tseng, Mark Mcgill, Éric Lecolinet, Samuel Huron, and Jan Gugenheimer. 2023. Memory Manipulations in Extended Reality. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:257952236

[6] Joel Brandt, Noah Weiss, and Scott R. Klemmer. 2007. txt 4 l8r: lowering the burden for diary studies under mobile conditions. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems* (San Jose, CA, USA) *(CHI EA '07)*. Association for Computing Machinery, New York, NY, USA, 2303–2308. https://doi.org/10.1145/1240866.1240998

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165

[8] Niamh Caprani, John Greaney, and Nicola Porter. 2006. A Review of Memory Aid Devices for an Ageing Population. *PsychNology J.* 4 (2006), 205–243. https://api.semanticscholar.org/CorpusID:9598075

[9] Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to Refine Queries for Retrieval Augmented Generation. arXiv:2404.00610 [cs.CL] https://arxiv.org/abs/2404.00610

[10] Samantha WT Chan. 2020. Biosignal-Sensitive Memory Improvement and Support Systems. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–7.

[11] Wei Ting Samantha Chan. 2022. *Augmenting Human Prospective Memory through Cognition-Aware Technologies.* Ph. D. Dissertation. ResearchSpace@ Auckland.

[12] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. *ArXiv* abs/1704.00051 (2017). https://api.semanticscholar.org/CorpusID:3618568

[13] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. 2022. MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:252735160

[14] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. arXiv:2403.04132 [cs.AI] https://arxiv.org/abs/2403.04132

[15] Jialin Dong, Bahare Fatemi, Bryan Perozzi, Lin F Yang, and Anton Tsitsulin. 2024. Don't Forget to Connect! Improving RAG with Graph-based Reranking. *arXiv preprint arXiv:2405.18414* (2024).

[16] Lydia Dubourg, Ana Rita Silva, Christophe Fitamen, Chris J. A. Moulin, and Céline Souchay. 2016. SenseCam: A new tool for memory rehabilitation? *Revue neurologique* 172 12 (2016), 735–747. https://api.semanticscholar.org/CorpusID:9803824

[17] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. arXiv:2404.16130 [cs.CL] https://arxiv.org/abs/2404.16130

[18] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, Cheng Peng, Chris Sweeney, Cole Wilson, Dan Barnes, Daniel DeTone, David Caruso, Derek Valleroy, Dinesh Ginjupalli, Duncan Frost, Edward Miller, Elias Mueggler, Evgeniy Oleinik, Fan Zhang, Guruprasad Somasundaram, Gustavo Solaira, Harry Lanaras, Henry Howard-Jenkins, Huixuan Tang, Hyo Jin Kim, Jaime Rivera, Ji Luo, Jing Dong, Julian Straub, Kevin Bailey, Kevin Eckenhoff, Lingni Ma, Luis Pesqueira, Mark Schwesinger, Maurizio Monge, Nan Yang, Nick Charron, Nikhil Raina, Omkar Parkhi, Peter Borschowa, Pierre Moulon, Prince Gupta, Raul Mur-Artal, Robbie Pennington, Sachin Kulkarni, Sagar Miglani, Santosh Gondi, Saransh Solanki, Sean Diener, Shangyi Cheng, Simon Green, Steve Saarinen, Suvam Patra, Tassos Mourikis, Thomas Whelan, Tripti Singh, Vasileios Balntas, Vijay Baiyya, Wilson Dreewes, Xiaqing Pan, Yang Lou, Yipu Zhao, Yusuf Mansour, Yuyang Zou, Zhaoyang Lv, Zijian Wang, Mingfei Yan, Carl Ren, Renzo De Nardi, and Richard Newcombe. 2023. Project Aria: A New Tool for Egocentric Multi-Modal AI Research. arXiv:2308.13561 [cs.HC] https://arxiv.org/abs/2308.13561

[19] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. 2024. VideoAgent: A Memory-augmented Multimodal Agent for Video Understanding. *arXiv preprint arXiv:2403.11481* (2024).

[20] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. ColPali: Efficient Document Retrieval with Vision Language Models. arXiv:2407.01449 [cs.IR] https://arxiv.org/abs/2407.01449

[21] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

[22] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *International Journal of Computer Vision* 127 (2016), 398 – 414. https://api.semanticscholar.org/CorpusID:8081284

[23] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh K. Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Z. Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Kahsay Gebreselasie, Cristina González, James M. Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran K. Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbeláez, David J. Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. 2021. Ego4D: Around the World in 3,000 Hours of Egocentric Video. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 18973–18990. https://api.semanticscholar.org/CorpusID:238856888

[24] Cathal Gurrin, Alan F Smeaton, Aiden R Doherty, et al. 2014. Lifelogging: Personal big data. *Foundations and Trends® in information retrieval* 8, 1 (2014), 1–125.

[25] Gillian R. Hayes, Shwetak N. Patel, Khai Nhut Truong, Giovanni Iachello, Julie A. Kientz, Rob Farmer, and Gregory D. Abowd. 2004. The Personal Audio Loop: Designing a Ubiquitous Audio-Based Memory Aid. In *Mobile HCI*. https://api.semanticscholar.org/CorpusID:11316625

[26] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. 2006. SenseCam: A retrospective memory aid. In *UbiComp 2006: Ubiquitous Computing: 8th International Conference, UbiComp 2006 Orange County, CA, USA, September 17-21, 2006 Proceedings 8*. Springer, 177–193.

[27] Fred Hohman, Mary Beth Kery, Donghao Ren, and Dominik Moritz. 2023. Model Compression in Practice: Lessons Learned from Practitioners Creating On-device Machine Learning Experiences. *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:263829166

[28] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166.

[29] Zhijian Hou, Lei Ji, Difei Gao, Wanjun Zhong, Kun Yan, Chao Li, Wing-Kwong Chan, Chong-Wah Ngo, Nan Duan, and Mike Zheng Shou. 2023. GroundNLQ @ Ego4D 'Natural Language Queries Challenge 2023. arXiv:2306.15255 [cs.CV] https://arxiv.org/abs/2306.15255

[30] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 105–113.

[31] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge Graph Embedding Based Question Answering. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019). https://api.semanticscholar.org/CorpusID:59528287

[32] Razan Jaber, Sabrina Zhong, Sanna Kuoppamäki, Aida Hosseini, Iona Gessinger, Duncan P Brumby, Benjamin R Cowan, and Donald McMillan. 2024. Cooking With Agents: Designing Context-aware Voice Interaction. In *Proceedings of the*

CHI Conference on Human Factors in Computing Systems. 1–13.

[33] Matthew Jamieson, Breda Cullen, Marilyn McGee-Lennon, Stephen Brewster, and Jonathan J Evans. 2014. The efficacy of cognitive prosthetic technology for people with memory impairments: A systematic review and meta-analysis. *Neuropsychological rehabilitation* 24, 3-4 (2014), 419–444.

[34] Matthew Jamieson, Brian O'Neill, Breda Cullen, Marilyn Rose McGee-Lennon, Stephen Anthony Brewster, and Jonathan J. Evans. 2017. ForgetMeNot: Active Reminder Entry Support for Adults with Acquired Brain Injury. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017). https://api.semanticscholar.org/CorpusID:2298134

[35] Omar Khattab, Mohammad Hammoud, and Tamer Elsayed. 2020. Finding the best of both worlds: Faster and more robust top-k document retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1031–1040.

[36] Jaewook Lee, Jun Wang, Elizabeth Brown, Liam Chu, Sebastian S. Rodriguez, and Jon E. Froehlich. 2024. GazePointAR: A Context-Aware Multimodal Voice Assistant for Pronoun Disambiguation in Wearable Augmented Reality. *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2024). https://api.semanticscholar.org/CorpusID:268132284

[37] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401 [cs.CL] https://arxiv.org/abs/2005.11401

[38] Franklin Mingzhe Li, Michael Xieyang Liu, Shaun K. Kane, and Patrick Carrington. 2024. A Contextual Inquiry of People with Vision Impairments in Cooking. *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2024). https://api.semanticscholar.org/CorpusID:267897983

[39] Jiahao Nick Li, Yan Xu, Tovi Grossman, Stephanie Santosa, and Michelle Li. 2024. OmniActions: Predicting Digital Actions in Response to Real-World Multimodal Sensory Inputs with LLMs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 8, 22 pages. https://doi.org/10.1145/3613904.3642068

[40] Mo Li, Songyang Zhang, Yunxin Liu, and Kai Chen. 2024. NeedleBench: Can LLMs Do Retrieval and Reasoning in 1 Million Context Window? arXiv:2407.11963 [cs.CL] https://arxiv.org/abs/2407.11963

[41] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved Baselines with Visual Instruction Tuning.

[42] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/

[43] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning.

[44] Xingyu Bruce Liu, Jiahao Nick Li, David Kim, Xiang 'Anthony' Chen, and Ruofei Du. 2024. Human I/O: Towards a Unified Approach to Detecting Situational Impairments. *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2024). https://api.semanticscholar.org/CorpusID:268203741

[45] Muhammad Maaz, Hanoona Abdul Rasheed, Salman H. Khan, and Fahad Shahbaz Khan. 2023. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:259108333

[46] Abdul Majeed and Sungchang Lee. 2021. Anonymization Techniques for Privacy Preserving Data Publishing: A Comprehensive Survey. *IEEE Access* 9 (2021), 8512–8545. https://api.semanticscholar.org/CorpusID:231616865

[47] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. EgoSchema: A Diagnostic Benchmark for Very Long-form Video Language Understanding. arXiv:2308.09126 [cs.CV] https://arxiv.org/abs/2308.09126

[48] Steve Mann. 1996. Wearable Tetherless Computer-Mediated Reality: WearCam as a wearable face-recognizer, and other applications for the disabled. https://api.semanticscholar.org/CorpusID:11838759

[49] Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2024. Multi-hop Question Answering. arXiv:2204.09140 [cs.CL] https://arxiv.org/abs/2204.09140

[50] Ken McRae and Michael Jones. 2013. *14 Semantic Memory*. Vol. 206. Oxford University Press Oxford.

[51] Aamer Nadeem and Muhammad Younus Javed. 2005. A Performance Comparison of Data Encryption Algorithms. *2005 International Conference on Information and Communication Technologies* (2005), 84–89. https://api.semanticscholar.org/CorpusID:14441015

[52] Jakob Nielsen. 1994. *Usability engineering*. Morgan Kaufmann.

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV] https://arxiv.org/abs/2103.00020

[54] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. arXiv:2401.18059 [cs.CL] https://arxiv.org/abs/2401.18059

[55] Jana Sedlakova, Paola Daniore, Andrea Horn Wintsch, Markus Wolf, Mina Stanikic, Christina Haag, Chloé Sieber, Gerold Schneider, Kaspar Staub, Dominik Alois Ettlin, et al. 2023. Challenges and best practices for digital unstructured data enrichment in health research: a systematic narrative review. *PLOS Digital Health* 2, 10 (2023), e0000347.

[56] Youngsoo Shin, Ruth Barankevich, Jina Lee, and Saleh Kalantari. 2021. PENCODER: Design for Prospective Memory and Older Adults. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (2021). https://api.semanticscholar.org/CorpusID:233987041

[57] Timothy Sohn, Kevin A. Li, William G. Griswold, and James D. Hollan. 2008. A diary study of mobile information needs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 433–442. https://doi.org/10.1145/1357054.1357125

[58] Alexander S Szalay. 2008. Jim gray, astronomer. *Commun. ACM* 51, 11 (2008), 58–65.

[59] Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. MultiModalQA: Complex Question Answering over Text, Tables and Images. arXiv:2104.06039 [cs.CL] https://arxiv.org/abs/2104.06039

[60] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL] https://arxiv.org/abs/2302.13971

[61] Endel Tulving. 2002. Episodic Memory: From Mind to Brain. *Annual Review of Psychology* 53, Volume 53, 2002 (2002), 1–25. https://doi.org/10.1146/annurev.psych.53.100901.135114

[62] Sunil Vemuri, Chris Schmandt, Walter Bender, Stefanie Tellex, and Bradford Lassey. 2004. An Audio-Based Personal Memory Aid. In *Ubiquitous Computing*. https://api.semanticscholar.org/CorpusID:309402

[63] Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Qian Wang, and Yue Zhang. 2024. Novelqa: A benchmark for long-range novel question answering. *arXiv preprint arXiv:2403.12766* (2024).

[64] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL] https://arxiv.org/abs/2201.11903

[65] Anran Xu, Shitao Fang, Huan Yang, Simo Hosio, and Koji Yatani. 2024. Examining Human Perception of Generative Content Replacement in Image Privacy Protection. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 777, 16 pages. https://doi.org/10.1145/3613904.3642103

[66] G. Yang, Lekha Chaisorn, Yunlong Zhao, Shi-Yong Neo, and Tat-Seng Chua. 2003. VideoQA: question answering on news video. *Proceedings of the eleventh ACM international conference on Multimedia* (2003). https://api.semanticscholar.org/CorpusID:207716

[67] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:52822214

[68] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378* (2021).

[69] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *North American Chapter of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:233219869

[70] Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:259075356

[71] Liangfu Zhang, Anwen Hu, Jing Zhang, Shuo Hu, and Qin Jin. 2023. MPMQA: Multimodal Question Answering on Product Manuals. *ArXiv* abs/2304.09660 (2023). https://api.semanticscholar.org/CorpusID:258212471

[72] Wazeer Deen Zulfikar, Samantha Chan, and Pattie Maes. 2024. Memoro: Using Large Language Models to Realize a Concise Interface for Real-Time Memory Augmentation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 450, 18 pages. https://doi.org/10.1145/3613904.3642450

A Prompts for LLMs

A.1 Identifying Composite Contexts

```
System instruction:
You are an intelligent agent capable of generating a list of COMPOSITE CONTEXTS inferred from the given memory.
Composite context refers to a combination of time, location, people, objects, environment and activities. Such composite contexts could be inferred from the explicit content (e.g., text showing the event info) or implicit cues (e.g., multiple changes in location indicating travel). Focus on relatively important composites such as travel, conferences, and important meetings and focus less on trivial events.
For each composite context, identify the related episodic memory ids. This could be due to time (e.g., the memory occurs during the event), location (e.g., the memory takes place at the event location), or specific content (e.g., the memory mentions the event).
Additionally, rate the importance of each event on a scale from 1 to 3, where 3 denotes very major events (e.g., multi-day events or highly important events), 2 denotes moderately important events, and 1 denotes less important events.

Exemplar composite context types include:
An academic conference: "An academic conference";
Recreational travel: "Trip to Salt lake city", "Traveling to home town";
Locational change: "Location changed from Seattle to Irvine";
Outdoor activities: "Camping trip";
Personal milestones: "Birthday celebration", "Graduation ceremony", "first day in univeristy";
etc.
Output the list of composite context in a JSON object with the key 'composite_context'. Each event should be represented as a sub JSON object with the following keys: 'event_name' (detailed and concise), 'memory_ids' (list), 'start_date', 'end_date' (could be the same as start_date), 'location', 'is_multi_days', and 'importance'.
+
<List of structured captured memories>
```

A.2 Inferring Semantic Knowledge

```
System instruction:
You are an intelligent agent capable of generating a list of FACTS or KNOWLEDGE (referred to knowledge in the following) that can be inferred from the given memory and the related composite contexts. Focus on relatively important high-level semantic knowledge and focus less on trivial events. Avoid specific details about individual media
The knowledge should be detailed and self-contained.
Exemplar semantic knowledge includes:
<Examples of semantic knowledge>
Also identify the most representative episodic memories that contribute to the understanding of the knowledge.
Output a JSON object with the key 'knowledge'. Each knowledge item should include 'knowledge', 'memory_ids' (list)

Input:
<Structured captured memories in the sliding windows>
+
<Identified composite contexts identified in the sliding window>
```

A.3 Query Augmentation

```
System instruction:
augment_query = Given a query and today's date, identify the contextual filters. Contextual filters may include:
temporal information: e.g., "last week."
location information: e.g., "Hawaii."
visible objects: e.g., "poke bowl."
people Seen: e.g., "people at the conference."
```

```
activities performed: e.g., "ordering in a
    restaurant."
and more complex contexts such as events or
    travel which consists of multiple atomic
     contexts mentioned above: e.g., "
    traveling to Hawaii." (activity: travel,
     location: Hawaii).
The query may not contain detailed
    contextual filters. In such cases, make
    reasonable inferences. For example, for
    query "What products did I buy from
    Sephora", the result could be obtained
    from a Sephora receipt. Thus inferred
    contextual filters for objects might be
    "makeup/skincare products or receipts."

Output a JSON object with the key '
    augmented_query', including the sub-keys
     'start_date', 'end_date', 'location', '
    objects', 'people', 'activities', and '
    complex_context'. Each sub-key should be
     a single string. Leave any sub-key
    empty if not applicable.
```

A.4 Generating Answers Based on Retrieved Results

```
System instruction:
Given a query, a list of memories and
    personal knowledge, generate a
    comprehensive answer to the query.
Identify the episodic memories that can
    provide evidence to the question.
If the answer is not explicitly presented in
     the memories, make a reasonable
    inference.
Output a JSON object with the key 'answer',
    'explanation' and 'memory_ids'.
The 'answer' should be a string and '
    memory_ids' should be a list of memory
    ids

Input:
<Query>
+
<Retrieved semantic knowledge>
+
<Retrieved structured knowledge>
```

B Baseline Implementation

While there is no already-existing system designed for answering personal questions on captured memories, we manually designed and implemented a system as the baseline. Similar to OmniQuery, the baseline system also adopts an RAG architecture to adapt to the large number of captured memories. We utilized the basic structure of RAG illustrated in [21], which involves (1) indexing the external data sources with embedding models, (2) leverage vector-based search to retrieve the top K relevant data instances (3) based-on the retrieved data, utilizing a powerful LLM to generate the final answer. Note that typical RAG systems require a chunking phase, where long documents are split into smaller chunks for more precise matching and retrieval of relevant information. In our case, each captured memory already represents a limited amount of information and is naturally separated. Therefore, we treat each captured memory as an individual chunk.

Figure B1 demonstrates the structure of the baseline system in our experiment. The baseline also processes the captured memories by leveraging a multimodal model (`GPT-4o`) to generate detailed captions for each memory. Additionally, it extracts temporal and geographical information from the metadata and processes it in the same manner as OmniQuery. This ensures that the processed memories include the temporal and geographical data, which are common components in users' queries. The temporal and geographical information is concatenated to the generated caption. Then the concatenated text sequence is encoded into text embeddings using embedding models (`text-embedding-3-small`).

In the retrieval stage, the query is first encoded into the text embeddings using the same embedding model, and then retrieve the top K (K=50) captured memories using vector-based similarity search. The retrieved top K captured memories are then ordered in temporal sequence, and then sent to the LLM (`GPT-4o`) for generating the answer. The prompt used for the answer generation is the same as OmniQuery.

C Failure Cases Analysis

C.1 Failure Case Categorization

Analyzing failure cases is important to understanding the limitations and improving the design of OmniQuery. Among the 137 queries tested in the study, we identified 25 queries with inaccurate results (OmniQuery's UPA ≤ 2) as failure cases. Additionally, we reached out to participants asking them to manually retrieve the correct memories for these failures. Through this analysis, we categorized them and propose future solutions for each:

Case 1: Ambiguity (8 cases): Ambiguity in language-based interaction was the cause of failure in certain cases based on our analysis. Specifically, such ambiguities can be categorized as follows:

(1) **Wording ambiguity**: P3 asked "what was the pool place I went in NYC". While they were referring to a billiards place (less ambiguous term), OmniQuery interpreted it as a swimming pool, resulting in retrieval failure (Figure 2a).
(2) **Reference ambiguity due to multiple valid answers**: Some queries have multiple potential answers. For example, P4 asked "What is the price of the medicine I bought?" They were referring to the most recent hospital visit, but OmniQuery retrieved a different medicine receipt from the memory, leading to failure in answering users' question.
(3) **Contextual ambiguity:** P6, a photographer, asked, "How many times did I work as a photographer in the past few months?" As

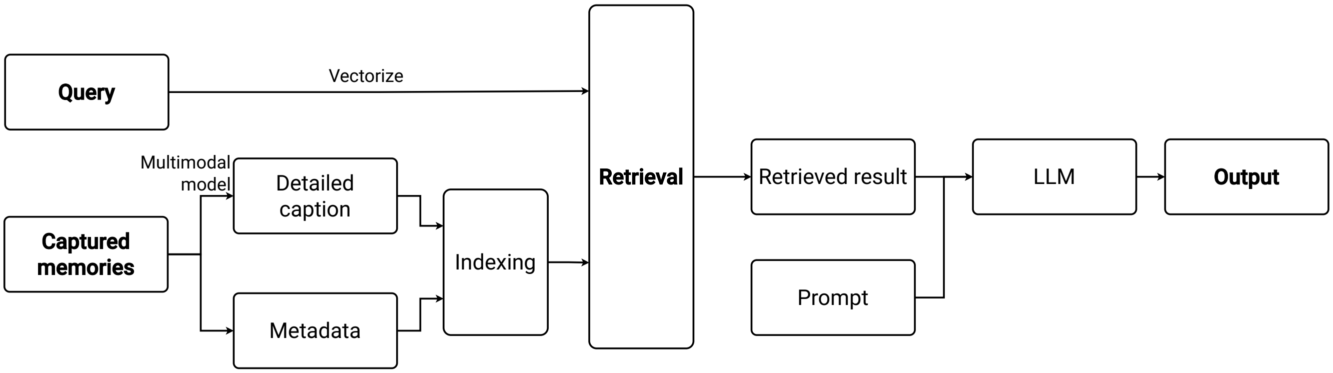


Figure 1: Structure of the baseline implementation.

they also enjoy personal photography, the boundary between photos taken for work and those taken for leisure is ambiguous, causing the system to fail.

The above presents the challenges OmniQuery faces in addressing ambiguity, both in understanding user queries and in analyzing retrieved results. For a detailed discussion on strategies to address these uncertainties, please refer to Section 9.2.

Case 2: Lack of contextual cues (7 cases): Several failure cases occurred due to insufficient contextual information to associate the target memory with the input query. There are two types:

(1) **Target context being too implicit:** As shown in Figure 2b, P4 asked "What is the content of the last meeting with my advisor last week?" The target memory, a photo of a notebook page, lacks contextual cues to associate it with the meeting. This led to retrieval failure. In such cases, OmniQuery should ask the user to clarify or iterate on the query (e.g., specifying the type of memory if the user has an idea).
(2) **People Identity and Metadata:** For example, P5 asked "Where did I travel with two Korean friends last month?" OmniQuery currently lacks access to facial recognition or metadata that can identify and associate individuals by attributes such as race. This also led to retrieval failure when answering this question. Future work could integrate with platforms like Google Photos or Apple Albums, which group photos by individuals using facial recognition. Additionally, users could manually add metadata via linking photos to contacts or descriptive tags (e.g., "friends from summer school"), enabling the system to handle such queries better.

Case 3: Information loss during text-based preprocessing (3 cases): OmniQuery currently adopts a text-based augmentation to extract atomic, composite context and semantic knowledge, which might lead to information loss. For example, P10 asked "What was the brand of the golf shirt I saw in the store?" The brand logo was barely visible in the bottom-left corner in the target memory (Figure 2c) and was not captured during the preprocessing. This led to retrieval failures when answering the question. Future work could integrate current text-based retrieval with advanced multimodal retrieval models (*e.g.*, ColPali [20]), which are capable of keeping more details during the retrieval process.

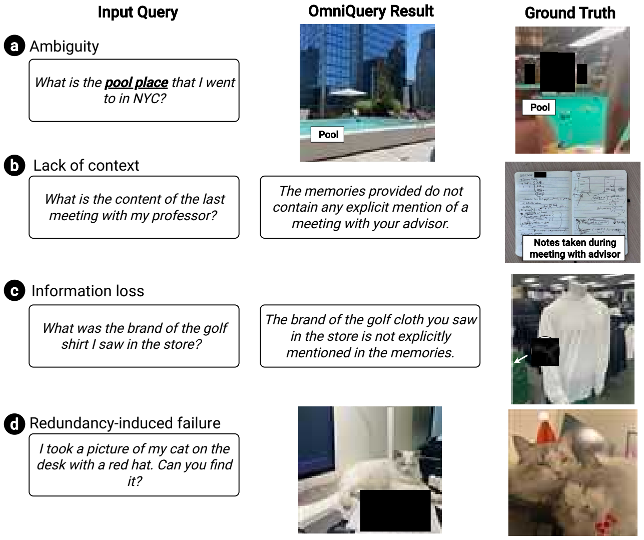

Input Query
OmniQuery Result
Ground Truth
a Ambiguity
What is the pool place that I went to in NYC?
Pool
Pool
b Lack of context
What is the content of the last meeting with my professor?
The memories provided do not contain any explicit mention of a meeting with your advisor.
Notes taken during meeting with advisor
c Information loss
What was the brand of the golf shirt I saw in the store?
The brand of the golf cloth you saw in the store is not explicitly mentioned in the memories.
d Redundancy-induced failure
I took a picture of my cat on the desk with a red hat. Can you find it?

Figure 2: Four exemplar failure cases: (a) lack of context, (b) wording ambiguity, (c) information loss during processing and (d) redundancy-induced failure.

Case 4: Redundacny-induced failure (3 cases): Similar to the "needle in a haystack" challenge [40], OmniQuery's performance might degrade when too many memories are retrieved during the retrieval phase. For example, P7 frequently takes photos of their cats and when they asked "I took a picture of my cat on the desk with a red hat. Can you find it?" OmniQuery tries to retrieve all relevant memories about their cats, resulting in failure to find the correct one (Figure 2d). In contrast, as the baseline system always retrieves a fixed number of results, it is able to identify the correct answer by narrowing down the search space. To address this, query-aware filtering process such as reranking [15] could be employed to narrow down the search space. Additionally, employing a Top-K retrieval strategy [35] could provide users with more options to enhance overall performance.

Case 5: Subjectivity-induced failure (1 case): P2 asked, "Give me the best selfie I took." However, the subjective nature of "the best" made it difficult for OmniQuery to determine the correct answer. To address such cases, future systems could integrate user preferences (e.g., leveraging marked favorites) or enable interactive clarifications (e.g., asking, "Do you prefer an indoor or outdoor selfie?") to better align with user intent.

Case 6: Target memory out of scope (3 cases): In three cases, the target memory was unavailable. In one case, the target memory was outside the 100-file range while the participants thought it had been included. In the other two cases, participants mistakenly thought they had captured the memory, but it was not actually captured. It is important to communicate such uncertainties to users when the system believes a query to be not answerable. For detailed discussion, please refer to Sec 9.2.

C.2 Perceived Feelings from Participants

We also present cases when participants reacted negatively to the answers. All participants encountered cases where the answers are inaccurate. Some were incomplete (e.g., P1 believed that they visited mroe than a few churches on the trip to Barcelona, but answers provided only two of them). Some were presumptive (e.g., P7 asked about recent social events, where the answers gave a piece of memory on a museum visit and explained that visiting museums is "likely with other people." However, P7 visited the museum alone). Some were making mistakes (e.g., P7 asked for the mostly visited attractions but both system mistakenly answered a museum, which was because P7 took a lot of museum pictures and both systems failed to recognize that they were the same visit.) Some even more challenging questions that caused failure of both systems include questions relate to a specific person. For example, P8 asked about her significant other and P5 asked about their "Korean friend" met in a trip. These cases represent the difficulty of understanding the nuances of personal relationships with personal album data.

C.3 Cases Where the Baseline RAG Outperforms OmniQuery

As discussed above, in cases where there is redundancy in the retrieved results before generating the answer, the baseline RAG system may perform better than OmniQuery because it retrieves a fixed number of memories, narrowing the input context and reducing noise during answer generation. To further understand the comparison, we also examined cases where the baseline performs better even when both results are reasonably accurate (UPA ≥ 3). Typically, in cases where there is ambiguity in the query, while OmniQuery might provide a relatively accurate result, the baseline often produces a more binary outcome (either highly accurate or highly inaccurate) due to its narrower retrieval scope. This reduces ambiguity but also limits its ability to handle complex contexts.

D Curating a Fixed Dataset Discussion

OmniQuery is evaluated through real user data, and its effectiveness can be further evaluated on a fixed benchmark dataset. However, curating a fixed benchmark dataset for personal data presents significant challenges:

- **Difficulties in collection of long-term personal data while preserving privacy**: Collecting diverse, long-term personal data while preserving participants' privacy with proper consent and redaction is complex. Prior work like Ego4D ensures privacy by obtaining consent for controlled indoor environments or by de-identifying data through redaction of visible and audible PII [23]. However, personal captured memories inherently include interactions with various people and sensitive personal content (e.g., photos of IDs or financial documents), which makes it impractical to obtain universal consent or redact all PII without compromising the data's utility. A promising solution is advanced generative content replacement (e.g., Xu *et al.* [65]), which replaces sensitive PII with synthetic content, ensuring privacy while preserving the cabality of benchmarking.
- **Difficulties in generating objective and unbiased QA pairs**: Personal captured memories and corresponding questions are by definition subjective and sometimes ambiguous. Subjectivity varies across question types. Some focus on objective facts (e.g., "What is the Wifi password?"), while others could be highly subjective (e.g., "What sports were my favorite last year?"). To address this, research methods can be applied to reduce ambiguity and bias, such as leveraging crowdsourcing to create QA pairs from third-person perspectives or assess the objectivity of existing pairs. Future work should aim to develop a taxonomy of question types and explore strategies for guiding crowd workers to assess objectivity or generate improved QA pairs based on the taxonomy.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009