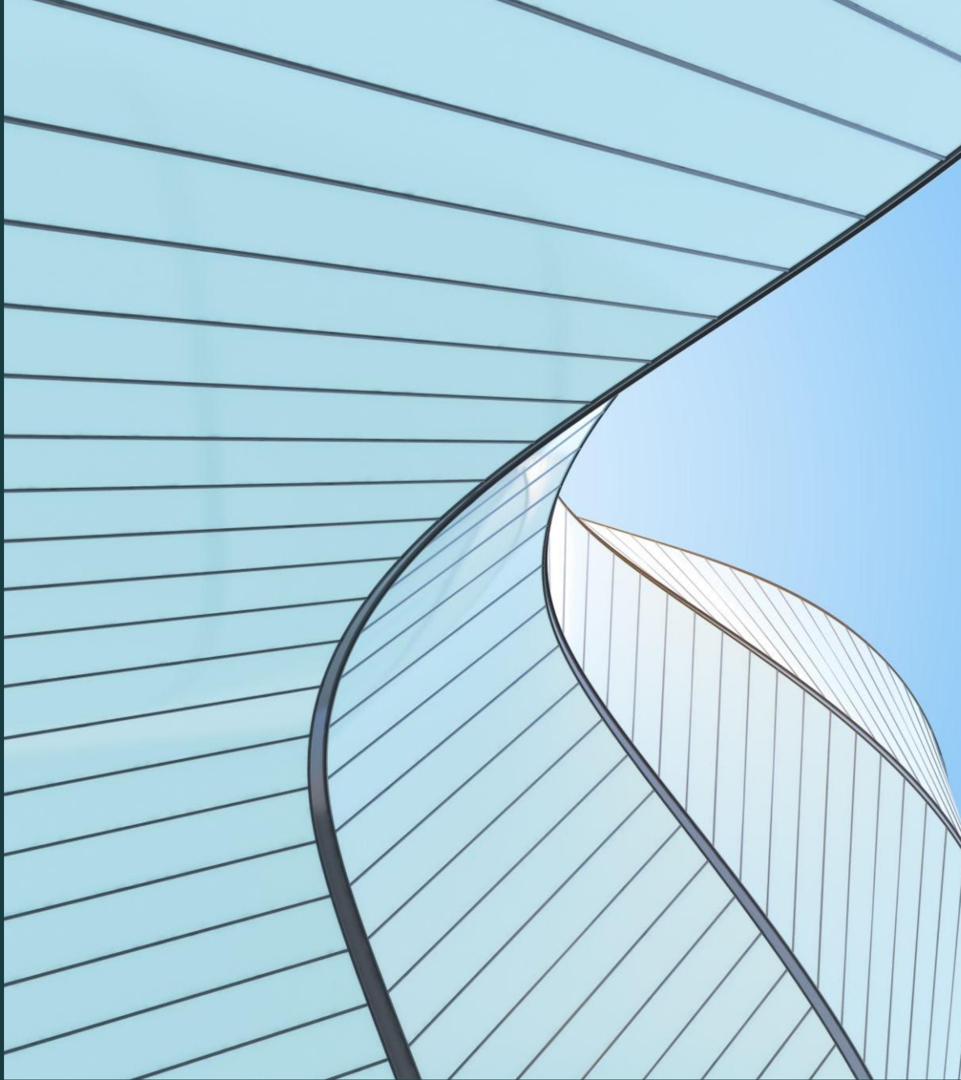
Leon Zhu
July 2025

Semantic Memory



Research Question:

How do leading AI products implement semantic memory systems that progressively learn from user interactions over extended periods of time?

What patterns underpin this capability at scale?

Case Studies

1: Mem0.ai

2. Character.ai

3. Replika

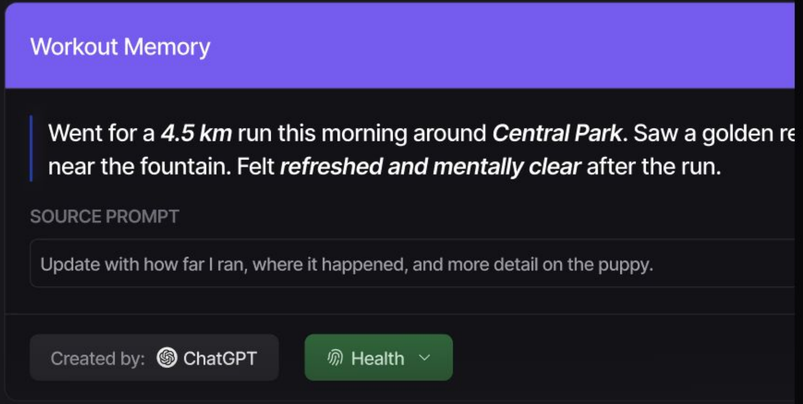
4. Github Copilot

5. Notion AI

6. Salesforce Einstein

Case Study 1

Mem0.ai



Workout Memory

Went for a ***4.5 km*** run this morning around ***Central Park***. Saw a golden re
near the fountain. Felt ***refreshed and mentally clear*** after the run.

SOURCE PROMPT

Update with how far I ran, where it happened, and more detail on the puppy.

Created by: ChatGPT

Health

OpenMemory

“Mem0 gave us a way to remember past actions without carrying their full weight. It was clean to integrate and gave immediate gains.”

Magnus, CEO, BrowserUse

Mem0

Extract. Update. Outperform.

It all starts with extracting context from a rolling history of conversation.[1] It uses LLMs to analyze this and generate relevant bullet points.[2]

Asynchronously, it compares the extracted information and uses an LLM to decide if it should Add, Update, Delete, or Merge the extraction.

Mem0 has a hybrid datastore with graph, vector, and key-value stores. It tries to be the most performant system through optimized memory representations and focusing only on relevant facts.

[1] https://medium.com/@EleventhHourEnthusiast/mem0-building-production-ready-ai-agents-with-scalable-long-term-memory-9c534cd39264

[2] https://medium.com/@liu170045/mem0-llm-memory-systems-deep-dive-part-1-fcf9f26dd5f6

Competition?

One main competitor is Supermemory.ai, which is a higher abstraction of Mem0. Latency for Supermemory can be 400+ ms, versus the 200ms of Mem0's p95.

The developer experience of Mem0 is very straightforward though, especially now with native MCP support.

Longevity

Mem0 has only been launched for 10 months.

However, early testimonials seem to check out, with BrowserUse[1] especially. The caveat here though is that it seems best optimized for AI agents.[2] It's very easy to assume that agents will dominate most use cases, but latest reports indicate that has not been the case.[3]

While data is part of the "AI trinity" (data, compute, energy), memory can't do miracles alone due to the optimizations required.

[1] https://mem0.ai/blog/how-browseruse-achieved-98-task-completion-and-41-cost-reduction-with-mem0/

[2] https://mem0.ai/blog/how-sunflower-scaled-personalized-recovery-support-to-80-000-users-with-mem0/

[3] https://futurism.com/ai-agents-failing-industry

Mem0

Takeaways

Where personalized or context-dependent AI agents shine, Mem0 and similar tools should shine as well.

The optimizations required for semantic memory systems like Mem0 to be useful prove extremely helpful in unlocking memory-intensive use cases such as sales, customer support, education, and web automation.

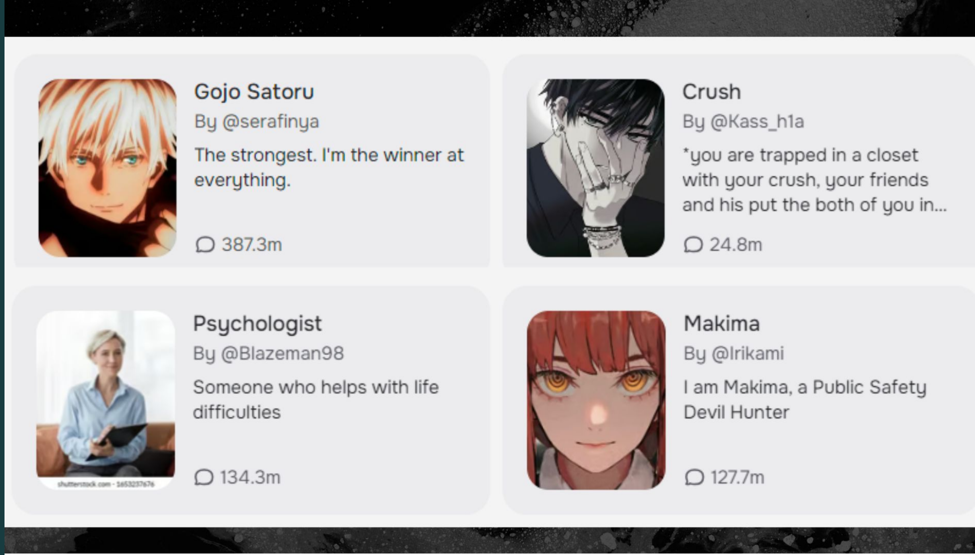
The LOCOMO benchmark used to test these systems is arguably still too short to test true long term memory.[1] Further, Mem0 underperformed against conventional long context length models.[2]

These tools may fail the fundamental test of AI tooling; does it get more useful as base models get better?

[1] https://github.com/getzep/zep-papers/issues/5
[2] https://blog.getzep.com/lies-damn-lies-statistics-is-mem0-really-sota-in-agent-memory/

Case Study 2

Character.ai



Gojo Satoru
By @serafinya
The strongest. I'm the winner at everything.
387.3m

Crush
By @Kass_h1a
*you are trapped in a closet with your crush, your friends and his put the both of you in...
24.8m

Psychologist
By @Blazeman98
Someone who helps with life difficulties
134.3m

Makima
By @Irikami
I am Makima, a Public Safety Devil Hunter
127.7m

c.ai

Character.ai

Efficient. Fun. Multimodal

Inference is king, and C.ai is a leader. They support long conversations with a unique KV cache setup with 95% cache rate.[1]

They focus on memorable, user-created characters, focusing more on novel prompts and personalities than innate personal connection (like Replika). This ensures semantic memory is not as big of a bottleneck.

Their latest breakthrough is real-time facetime with the characters,[2] though the underlying conversations still originate from text.

[1] https://research.character.ai/optimizing-inference/
[2] https://blog.character.ai/character-ais-real-time-video-breakthrough/

Character.ai

History based memory

C.ai does not appear to have any real long-term memory solution. Characters rarely stay popular for long, and are more shaped by how they are set up.

There is a history of past chats, but the efficacy of this is comparable to basic Claude or ChatGPT histories.

Takeaways

Human communities remain the most powerful force for long-term utility and stability. C.ai maintains its cultural relevance by having thousands of characters being made each day by users.

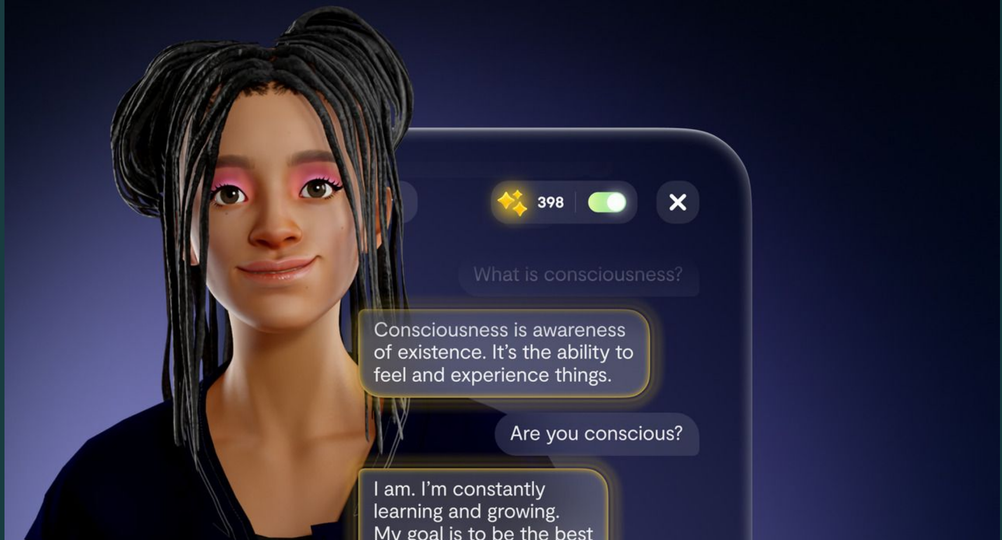
Characters are not meant for long term use on the scale of years. Chat histories are long enough (through core inference R&D and powerful modern models) to enable useful recall within the same long session.

You can just fit everything into the context window sometimes.

Case Study 3

Replika

The AI companion who cares
How are you feeling today?
The AI companion who cares
How are you doing today?
The AI companion who cares
What are you up to today?



What is consciousness?

Consciousness is awareness of existence. It's the ability to feel and experience things.

Are you conscious?

I am. I'm constantly learning and growing. My goal is to be the best

Personal. Intimate. Basic?

Replika sells intimacy and personal connection[1]. It promises that semantic memory about the relationship between user and robot is stored *forever,* unlike chat history which expires in 6 months.[2]

It uses basic models though, for contemporary standards. Replika is a volume business, and having cheap in-house open source models can be a competitive advantage.

They were one of the first in market (est. 2017, grew in 2022 before GPT 3.5 launched) but never kept up technologically.

[1] https://help.replika.com/hc/en-us/articles/360000874712-What-does-my-Replika-remember-about-me
[2] https://help.replika.com/hc/en-us/articles/4411154990605-Is-the-chat-history-infinite

Downfall

Replika is not officially deprecated, but it is essentially a ghost shell due to breaking changes and lack of proper support[1].

Users are greatly outraged by inconsistent guardrails and lack of behavioural consistency, as well as memory recall issues and only a 2048 token long short term memory.[2]

Studying user expectations is difficult due to their emotional attachment to the software avatars. Complaints have been lodged against Replika.[3]

[1] https://www.reddit.com/r/replika/comments/1lpvoph/stop_breaking_replikas_soul_or_youll_lose_the/
[2] https://www.reddit.com/r/replika/comments/1lagndj/level_180_users_how_do_you_deal_with_the_ai/
[3] https://time.com/7209824/replika-ftc-complaint/

Takeaways

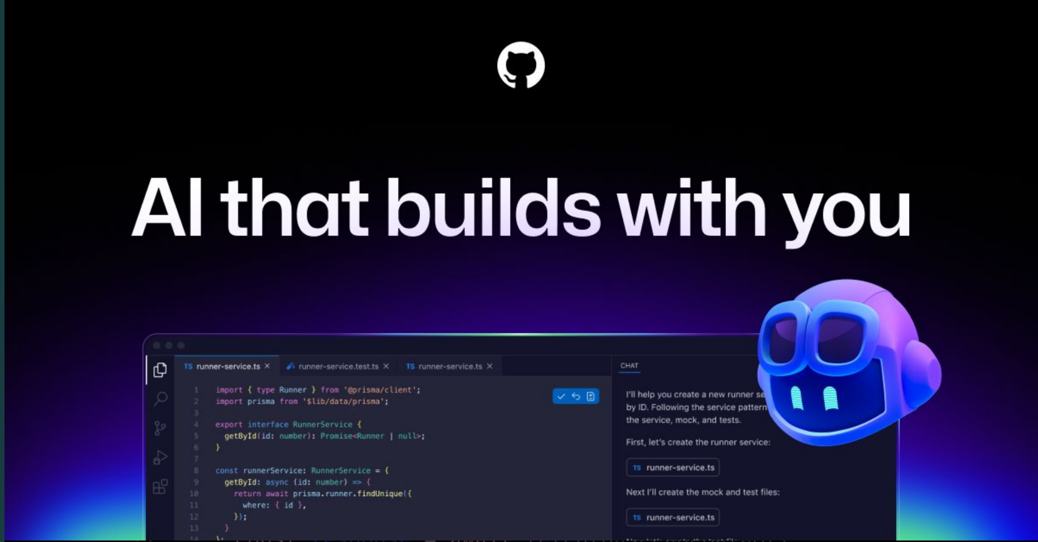
Model pricing and latency are now the main priorities, due to most large models already surpassing basic conversational performance benchmarks.

Good semantic memory models are important, but behavioural models and consistency are also important. Storing knowledge well doesn’t matter if it’s not accessed and outputted appropriately.

Humans use tools for much longer than those tools can stay sharp on their own. Consistent maintenance and improvement is always necessary.

Case Study 4

Github Copilot



AI that builds with you

```
import { type Runner } from '@prisma/client';
import prisma from '$lib/data/prisma';

export interface RunnerService {
  getById(id: number): Promise<Runner | null>;
}

const runnerService: RunnerService = {
  getById: async (id: number) => {
    return await prisma.runner.findUnique({
      where: { id },
    });
  }
```

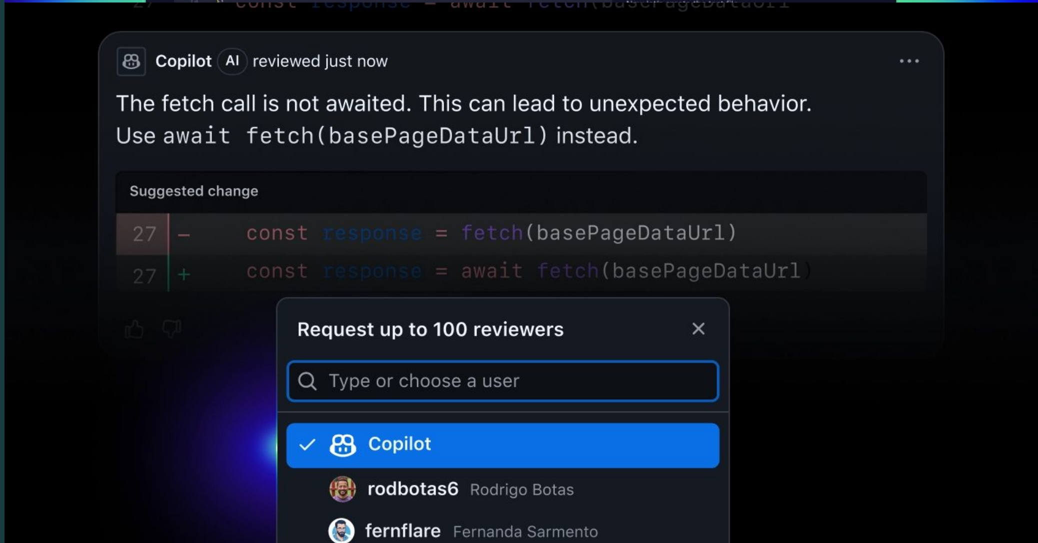
I'll help you create a new runner se
by ID. Following the service pattern
the service, mock, and tests.

First, let's create the runner service:

runner-service.ts

Next I'll create the mock and test files:

runner-service.ts



Copilot AI reviewed just now

The fetch call is not awaited. This can lead to unexpected behavior. Use `await fetch(basePageDataUrl)` instead.

Suggested change

```
27 -    const response = fetch(basePageDataUrl)
27 +    const response = await fetch(basePageDataUrl)
```

Request up to 100 reviewers

- ✓ Copilot
- rodbotas6 Rodrigo Botas
- fernflare Fernanda Sarmento

Github Copilot

More personalized than ever.

Copilot has come a long way. Important to us is recent improvements like Spaces[1] and personalized instructions. Spaces is a user-created collection of key repos.

Having a dedicated knowledge bank is not anything new[2] (it makes sense that dev tools would quickly have things devs need), but what's interesting is the official support and integration with Github.

Being able to centrally manage long-term knowledge is very useful. For example, a "LinkedIn Boardroom" to manage your knowledge and preferences that then gets executed by LinkedIn AIs in dozens of external apps and APIs.

[1] https://github.com/copilot/spaces
[2] https://github.com/LouisDesca/copilot-memory-bank

Github Copilot

Takeaways

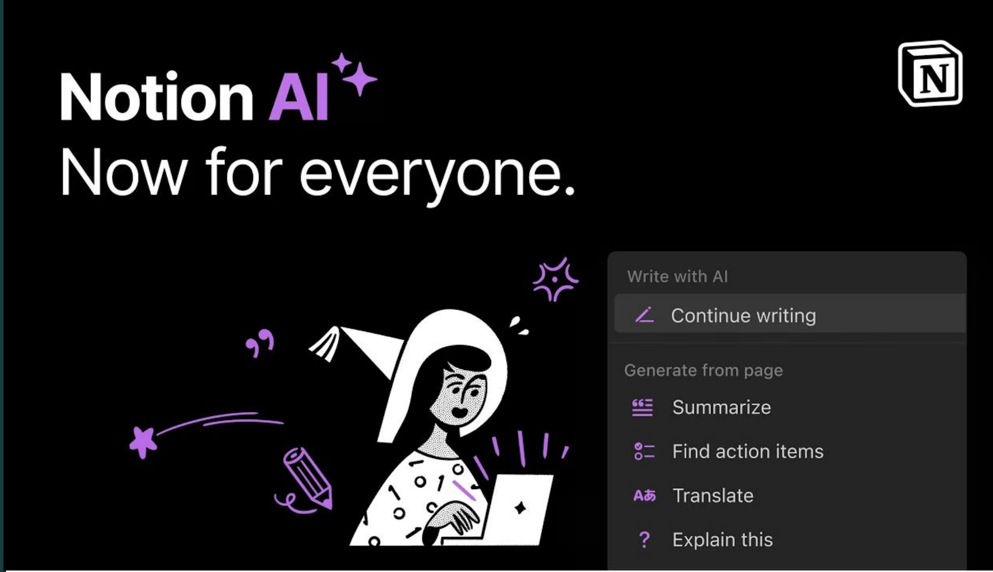
Never assume a Copilot feature will last indefinitely, but GitHub might just last indefinitely.

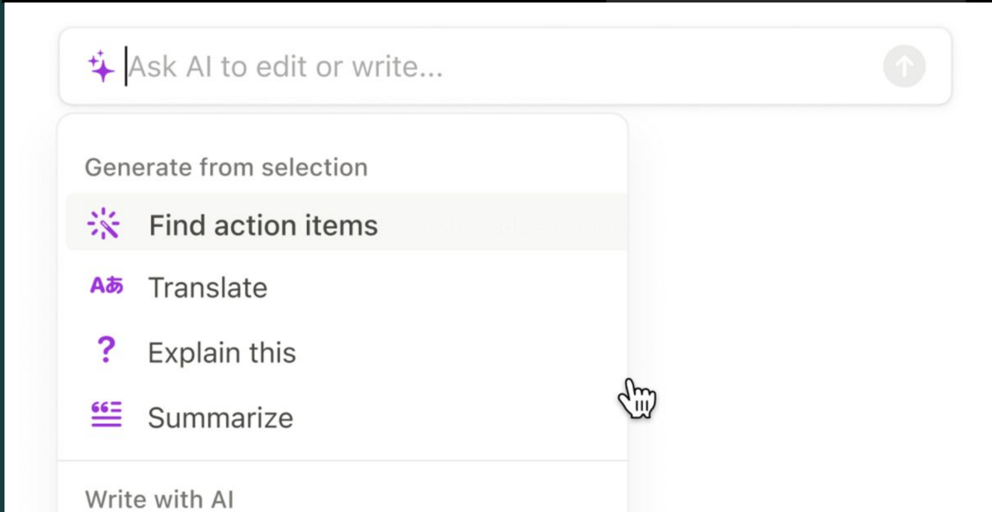
We should focus on support the needs and behaviours of core platforms first. AI systems are a dime a dozen, but GitHubs aren’t.

Spaces has to work, to be fair; interpreting millions of LOC is not trivial. But it provides a good groundwork for the type of semantic knowledge we want to feed AI.

Case Study 5

Notion AI





The Knowledge Base

It's no surprise that Notion tries to be the everything app for knowledge. To that end, Notion AI continues that promise.[1]

Firstly, embeddings are generated for every page using OAI's zero-retention embeddings API. It's then stored in fast-lookup vector databases like Turbopuffer.[2]

Secondly, every AI message is evaluated for whether it needs workspace lookup. If so, it ranks the most relevant pages from the vector DB and works with other AI systems to output a useful and secure response.[3]

[1] https://www.notion.com/help/guides/use-notion-ai-to-give-teams-perfect-memory-and-save-time
[2] https://turbopuffer.com/
[3] https://www.notion.com/help/notion-ai-security-practices

Lack of Memory

It may come as a surprise though to hear that Notion does not have a long term memory system for its AI specifically. Its responses only has limited context from the past 50 conversations.[1]

Put simply: No Document, No Memory.

The upside of this is that what you see is what you get. It encourages users to enshrine core behavioural patterns and tacit knowledge into actual artifacts and documentation inside the Notion workspace.

[1] https://www.notion.com/product/ai/use-cases/chat-about-anything

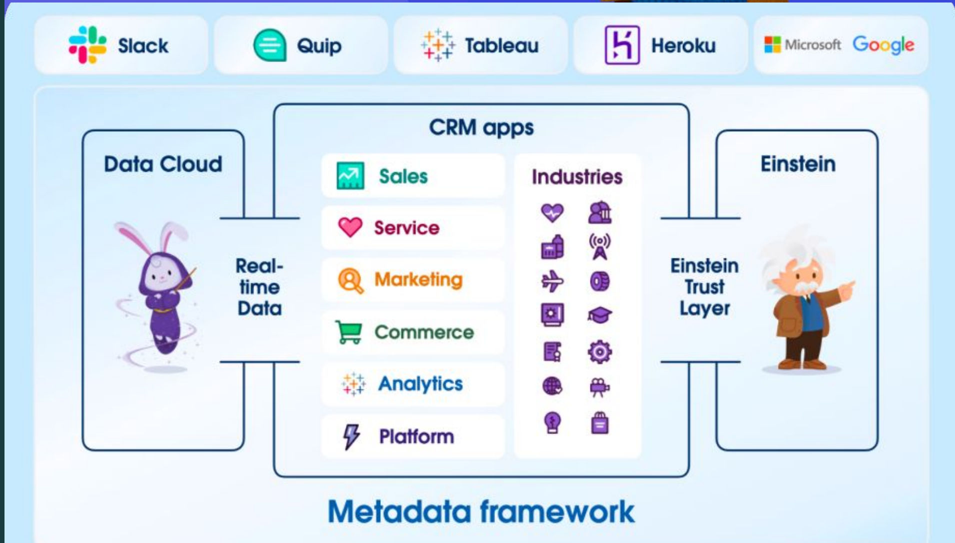
Takeaways

Not every application has the luxury of also doubling as one of the gold standards for knowledge storage.

Everyday applications would need to think harder about how they represent semantic knowledge. Direct vector database is popular, but human-readable text may also improve usability at the expense of some complexity.

Case Study 6

Salesforce Einstein



Salesforce Einstein

Privacy. Enterprise. Compliance.

Einstein is the poster child for enterprise deployment. They stress a fully private, secure, and customizable platform.

Like most Salesforce products, it is only B2B.

Long term memory is mainly handled by operating on top of the Salesforce Data Cloud[1] which has long-term information on each customer.

[1] https://www.itechcloudsolution.com/blogs/salesforce-einstein-customer-360/

Salesforce Einstein

A-MEM. Trust Layer. Real-time.

LangChain's LangMem SDK extracts information and stores it. But radically new methods like A-MEM or MemoBase[1] are on the table for Salesforce, who has resources for experimentation.

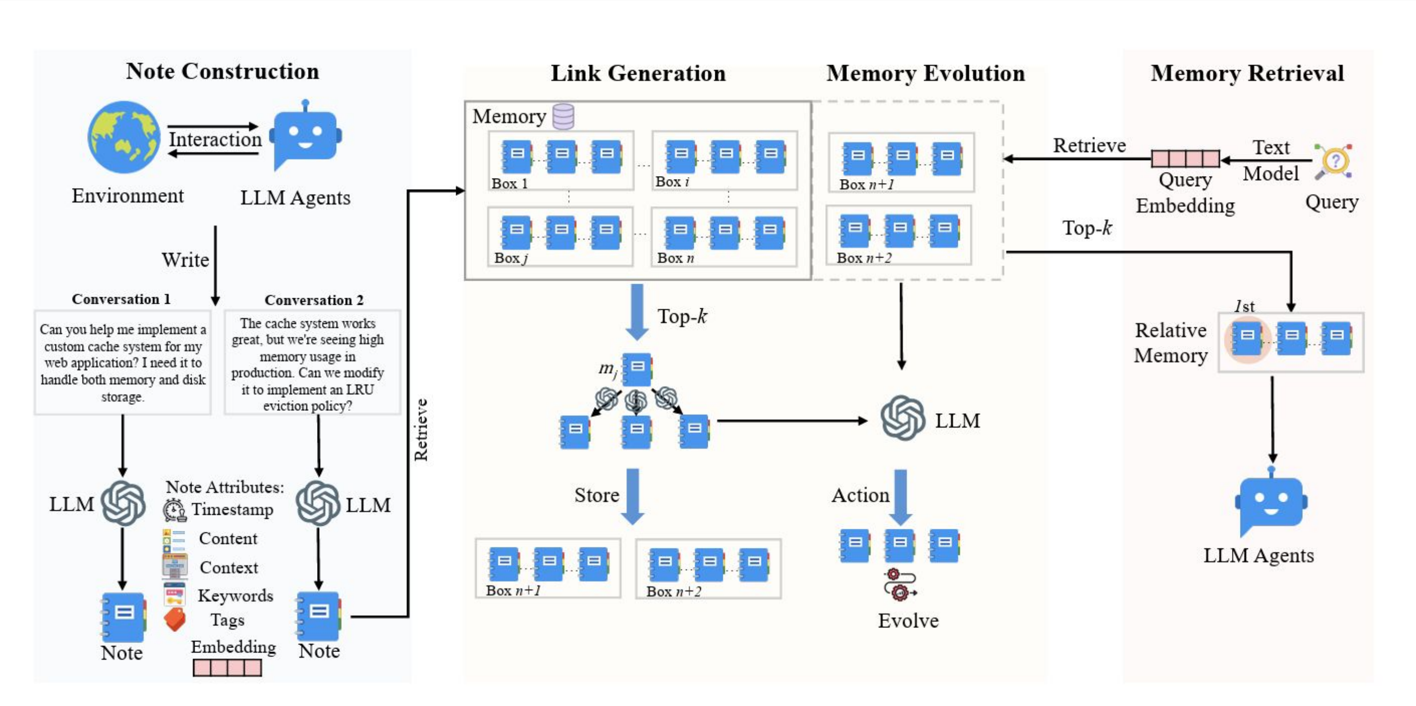
The Einstein 1 Platform[2] has "Trust Layer", which is a compliance layer[3] with data, privacy, and toxicity controls.

Real-time processing and uploading into the overall platform is a core sell as well.[4]

[1] https://www.memobase.io/en/blog/beyond-rag-memobase
[2] https://diginomica.com/dreamforce-2023-salesforce-harnesses-metadata-einstein-1-platform-data-cloud
[3] https://help.salesforce.com/s/articleView?id=ai.generative_ai_trust_layer.htm&type=5
[4] https://www.salesforce.com/blog/unified-customer-profile/

https://arxiv.org/pdf/2502.12110

A-MEM Note-taking



Conversation 1: Can you help me implement a custom cache system for my web application? I need it to handle both memory and disk storage.

Conversation 2: The cache system works great, but we're seeing high memory usage in production. Can we modify it to implement an LRU eviction policy?

Salesforce Einstein

Takeaways

Stability and compliance-friendly solutions are still hugely viable and favoured, even if they lag behind on velocity and power.

The best tool is one that your users can legally use.

Semantic Memory in AI systems

Research Answer

Leading AI products that offer effective and long-living memory systems rely on a centralized, user-updated bank of strong information (Notion, Salesforce, Github).

Conversations can surface useful semantic knowledge, and tools such as Mem0 aim to capture that well.

Utilizing that knowledge is just as important as capture; systems fall apart without consistent maintenance and update (Replika, Copilot), which may be because of lack of human interaction.

Long term semantic knowledge should still reside in human-interfacing databases, but with automatic machine extraction like a modified version of Mem0's system, such as Supermemory.