Making LLM Memory Useful

What Matters and What Doesn’t

Eddie Landesberg
CEO, Fondu
4/16/25

Who dis?

- Co-founder/CEO of Fondu Technologies, building user-owned contextual AI
- 12+ years building AI systems at the intersection of personalization and consumer data
- First data science hire in marketing at Salesforce. Led advertising spend optimization at Stitch Fix ($150M annual spend)

Good memory

The purpose of long term memory is to improve system outputs.

It’s up to you to determine what quality means for your application.

Core components

1. Save useful information

2. Process it

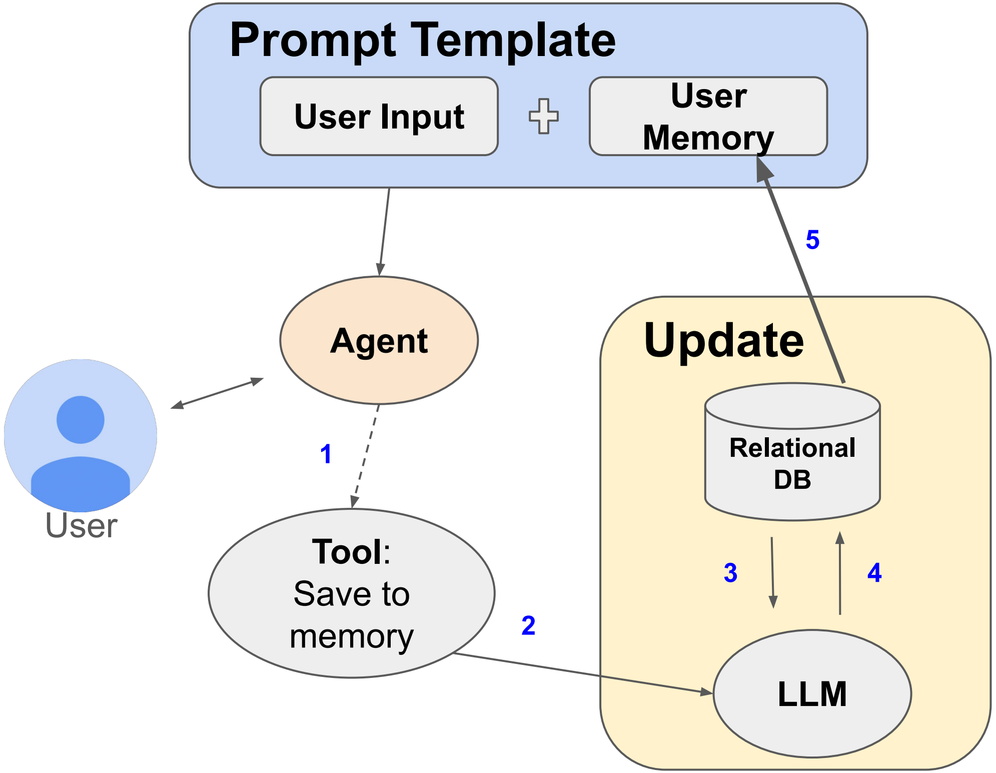
3. RAG it

Simple memory system that works surprisingly well

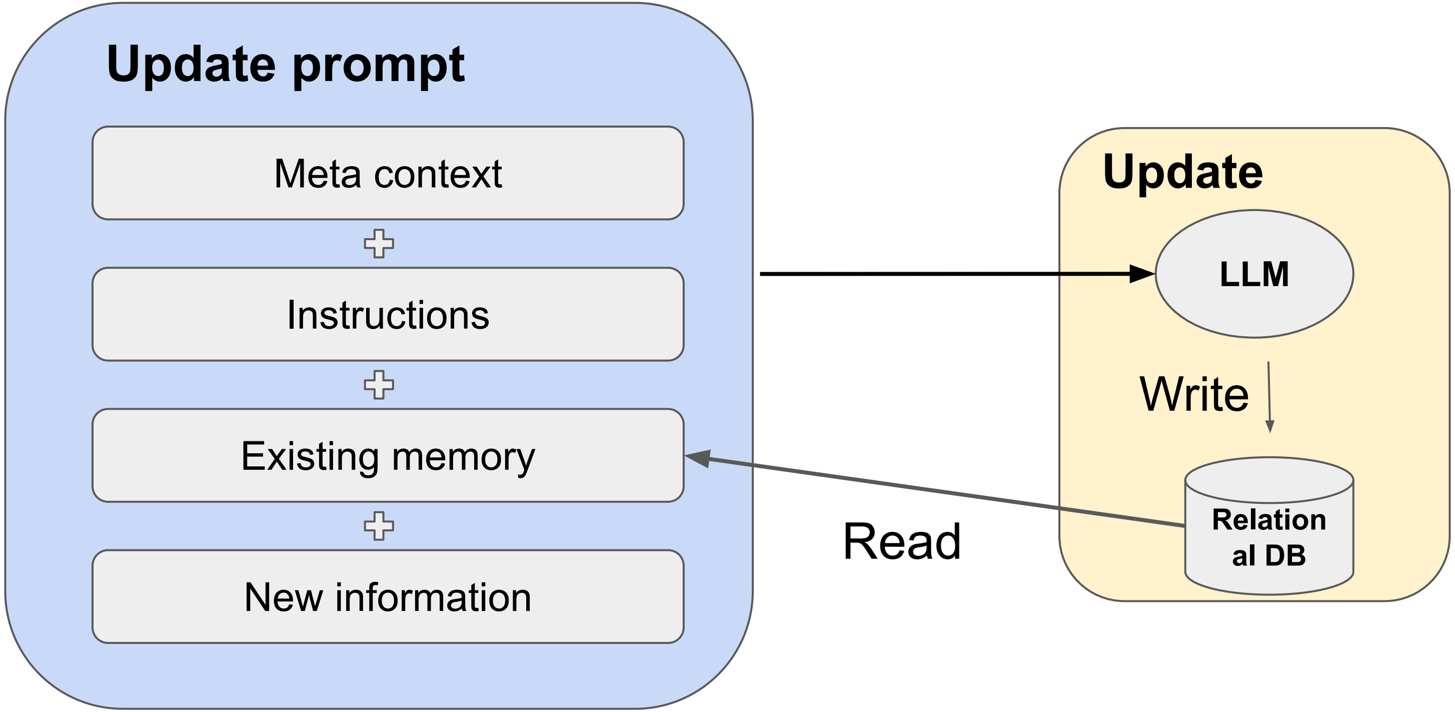
1. Agent decides what information to save, produces string

2. When knowledge is added, retrieve and update a string

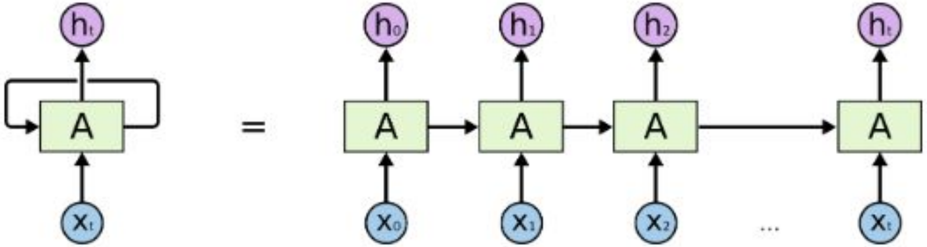
3. Inject into every prompt



Prompting for memory updates



Fun analogy



An unrolled recurrent neural network.

Trade Offs

Strengths

- Great latency: retrieval is just a relational database read
- Dead simple: easy to understand / debug, no fancy tools needed
- Always available
- If your prompting is good, it works way better than you’d expect

Weaknesses

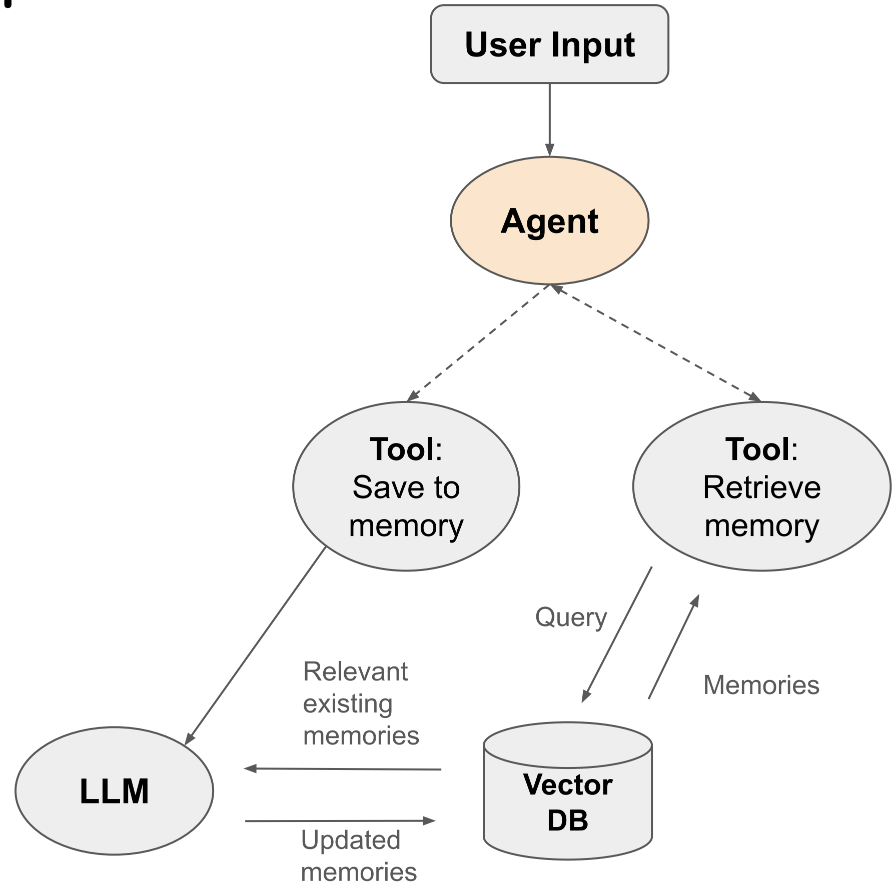
- Token inefficient when string gets big
- Not very scalable

A more scalable approach

1. Agent decides what information to save, produces string

2. When knowledge is added, retrieve and update top k most relevant memories

3. Agent decides when to retrieve top k memories. Submits query to determine relevance



Trade Offs

Strengths

- Scalable
- Token efficient

Weaknesses

- Higher latency
- Update logic is significantly more complicated
- Availability depends on retrieval quality and agent decision making

A pragmatic road-map

1. Start with a simple implementation that’s easy to introspect
2. Use it, look at the data, refine your prompts, develop evals
3. Extend as needed

Thank you

Blog post about our Fondu's semantic memory system:
https://www.youfondu.com/blog/semantic-understanding

On x: @edwardlandesber
On linkedin: https://www.linkedin.com/in/eddie-landesberg/
On github: https://github.com/elandesberg