Insights on Context & Memory for a Supportive Assistant

Cedric L’homme

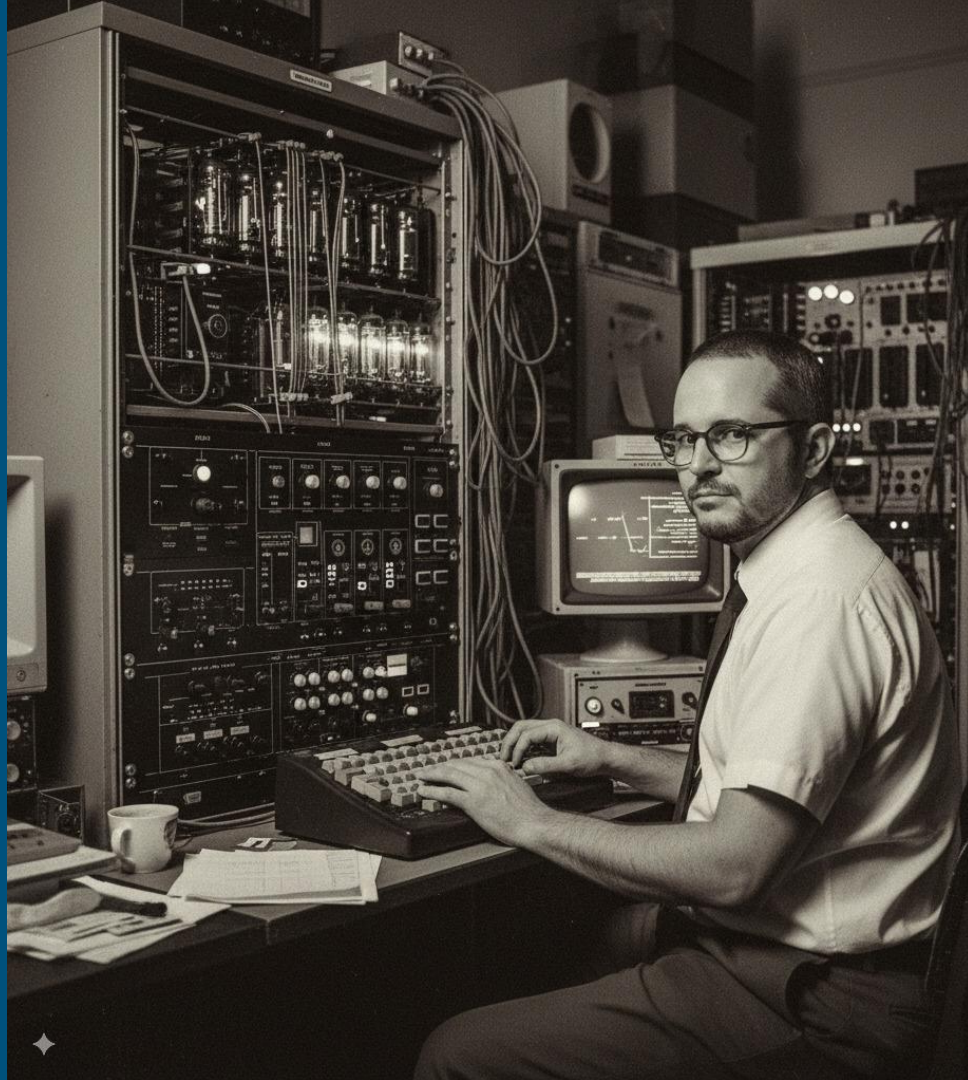
v. 2025-09-10

Me, Myself & I

Lead @ Osedea

I’ve been a Software Crafter for more than 25 years.

I will probably never retire, so please don’t follow any advice from me.



What I am building: a supportive assistant

- To experiment with LLM, agentic, audio, etc.
- I found that “speaking his mind” is very helpful.
- There are fewer professional practitioners than needed, if needed.
- Probably fewer friends or acquaintances and less time than required.

WARNING: Anything that results from this experiment is not a psychologist, a doctor, a friend, or even a sentient being; it's merely a presence.

Expressing stigma and inappropriate responses prevents LLMs from safely replacing mental health providers https://arxiv.org/abs/2504.18412

Stochastic System without auto-stop

LLMs are prediction machines, auto-complete on steroids, that mimic patterns from training data, that’s why the style is confident.

If you give a targeted role without the expertise data, it will hallucinate.
E.g., training on English Literature and questions on Finance.

Hallucinations are just normal results but with bad statistics.

https://ml-site.cdn-apple.com/papers/the-illusion-of-thinking.pdf

Specifics

This presentation is a list of everything I've tried.

I don't care about the price; when I discuss enhancements, I'm primarily focused on "quality", not token counts (but still, we will talk about that).

BUT

I don't need:
- reasoning in the model, or coding, or tools calling, etc.
- multimodal capabilities
so, it's cheaper.

The technical problem to solve

How to deal with a growing context when you have long and multiple “discussions” with a bot?

If you don’t send everything in the context (current + history + documentation), how do you manage to send “just the right amount” of context?

The actual buzzword for that is : ~~Prompt Engineering~~ -> Context Engineering

Oh, and I don’t want to write, can we make it talk?

Demo

Demo

Create a session with “I lost my dog Wolfy. He was hit by a car and I have nightmares from then. My uncle Bob brought me to Five Guy’s to help me with my favorite fries: the Cajun fries.”

Disconnect everything and start a new session.

Demo with Pipecat.
Demo with GentleWind

Our Journey Ahead

- Warm-up: Achieving a good start with the initial context.
  - Glossary, Prompting, Observability, Privacy/Security, PList, Ali:Chat
- Good pace: How to manage the growth of the context.
  - Summarizer, RAG
- Marathon: How to access old context and the right context on the long run.
  - Graph, Entity

* Let’s add audio for fun

Initial Context

Some Vocabulary

Session (or Conversation): a group (or a list) of messages.

Interaction: a message from AI and the response from human

https://model-spec.openai.com/2025-02-12.html

Key Message Types in Session (e.g., LangGraph)

System Message: Used to prime AI behaviour, usually passed in as the first message in a sequence of inputs.

Human Message: Represents a message from a person interacting with the model.

AI (or Assistant) Message: Represents a message from the model. This can be either text or a request to invoke a tool.

https://python.langchain.com/docs/how_to/custom_chat_model/

Session Example

System Message: “You are a helpful assistant, and you answer in French”

AI Message: “Hey, how are you? Do you have a question?”

Human Message: “What is the capital of France?”

AI Message: “La capitale de la France est Paris.”

https://model-spec.openai.com/2025-02-12.html#be_empathetic

How to prompt?

Each model reacts differently on how a prompt is structured … and it changes every month…

- Test with **Prompt Versionning** (Ex: Langfuse)
- **Be specific and provide details and context**: identify main topic, give me top 5…, summarize this pdf, explain this concept, here are 5 reports on…, etc. instead of “analyse this file”.
- **Add your constraints**: use this format (csv, json, etc.), answer with this tone (formal, explain like I’m 5, etc.), etc.
- “**Role framing**” seems losing traction. “You are a expert trader…”
- Provide examples (“**Few-Shot Prompting**”)

Observability

You need observability for:

- Evaluations on general behaviour with the input variables
- Prompts versioning (v1, v2, staging, prod, etc.)
- Prompts evaluation (thumb-up, 3 stars, etc.)
- Latency metrics (e.g, time for the first token)
- Cost metrics ($/million tokens)

But how to measure “supportive” qualities? 🤷

https://opentelemetry.io
https://langfuse.com

SillyTavern, Character AI, etc.

Website or local service for roleplay.

You can create: characters, scenarios, worlds

Can rent access to uncensored LLMs (e.g., for violence, sex, etc.)

Please, try to make Biden and Trump debate whether furry cosplay should be supported at the office every Friday to enhance 'Casual Friday.'

https://sillytavern.pro

Privacy, Security, Cost and Ethical Considerations

API Best models but no privacy. Potential for account ban. You pay per call so it can be cheap if you have a low usage, costly on heavy usage.

Local Offers privacy but with limited models due to memory constraints and only with private models, resulting in lower quality. Big upfront payment for a rig with a costly graphic card.

LLM (private and open) on rented cloud servers (mainly AWS)

Offers more privacy depending on the cloud provider, and access to the models also depends on the LLM providers.

Open-weight LLMs on rented cloud servers (e.g., Llama, DeepSeek, Qwen, etc. on fly.io, modal, etc.)

Costly, but likely offer the right balance..

PList (Property List)

Provide a concise, token-efficient structure for listing everything relevant about a character—appearance, personality, likes, dislikes, etc.—making it easier for the AI to maintain characterization.

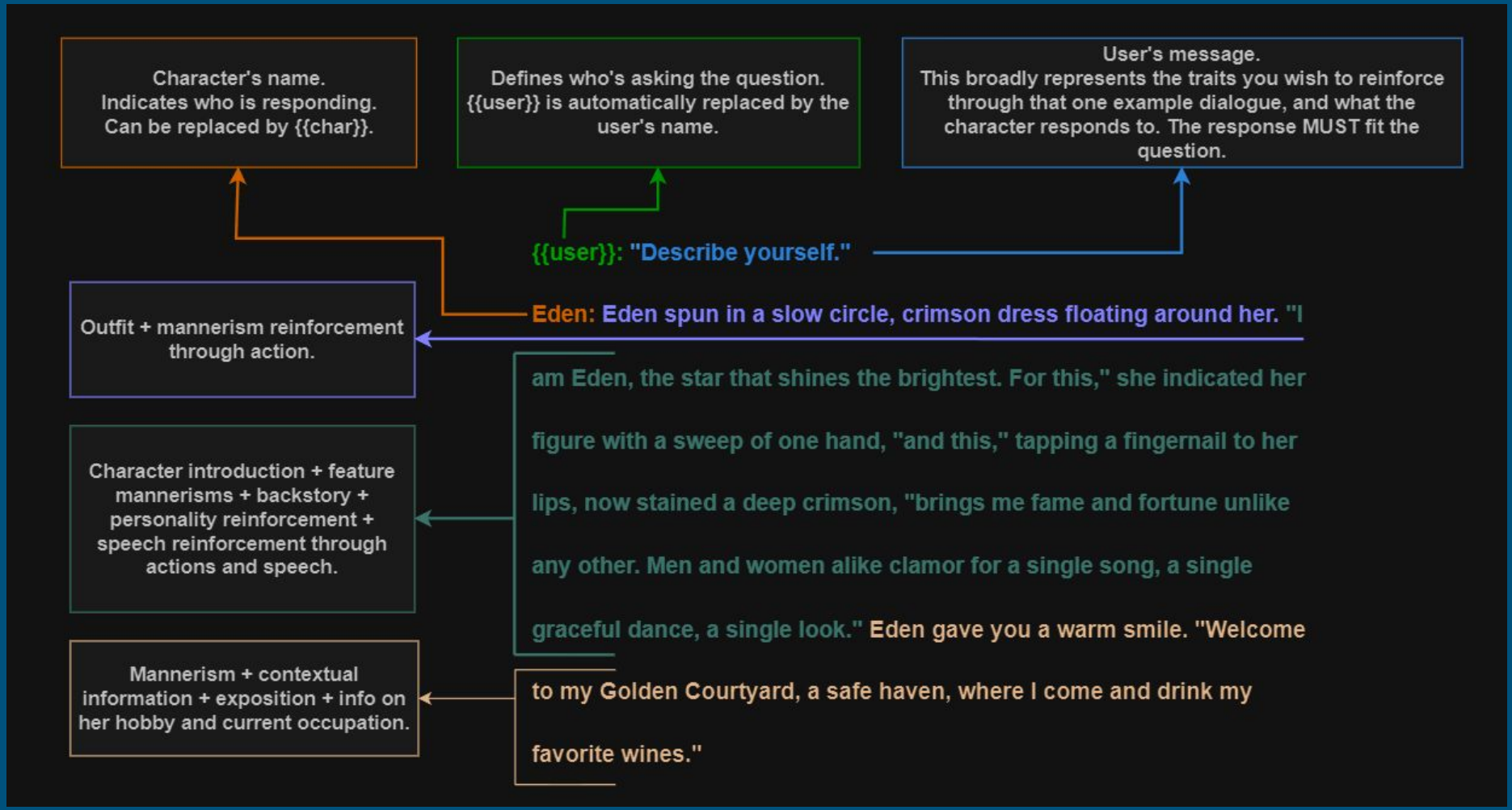
```
[Appearance: tall, curly hair(brown), glasses, casual clothing]
[Personality: curious, witty, shy]
```

https://pygmalion.chat

Ali:Chat

Demonstrate conversational style to reinforce the generated style.

https://wikia.schneedc.com/bot-creation/trappu/introduction


Character's name.
Indicates who is responding.
Can be replaced by {{char}}.
Defines who's asking the question.
{{user}} is automatically replaced by the user's name.
User's message.
This broadly represents the traits you wish to reinforce through that one example dialogue, and what the character responds to. The response MUST fit the question.
{{user}}: "Describe yourself."
Eden: Eden spun in a slow circle, crimson dress floating around her. "I am Eden, the star that shines the brightest. For this," she indicated her figure with a sweep of one hand, "and this," tapping a fingernail to her lips, now stained a deep crimson, "brings me fame and fortune unlike any other. Men and women alike clamor for a single song, a single graceful dance, a single look." Eden gave you a warm smile. "Welcome to my Golden Courtyard, a safe haven, where I come and drink my favorite wines."
Outfit + mannerism reinforcement through action.
Character introduction + feature mannerisms + backstory + personality reinforcement + speech reinforcement through actions and speech.
Mannerism + contextual information + exposition + info on her hobby and current occupation.

Growing Context

Different Memories

- Short-term memory: Recent messages.
- Long-term memory: Information to be retained. Can be previous sessions.
- Persistent Context: Information that you always want to send to the context (e.g., system message with initial prompt).

Enjoy large context on new models, but beware, papers are not aligned on if the models can deal with them efficiently or not. Linked to CAG, see further.

I saw some misuse of context with 10k tokens with very specific data so it’s probably not that good with 1M tokens…

Summarizer

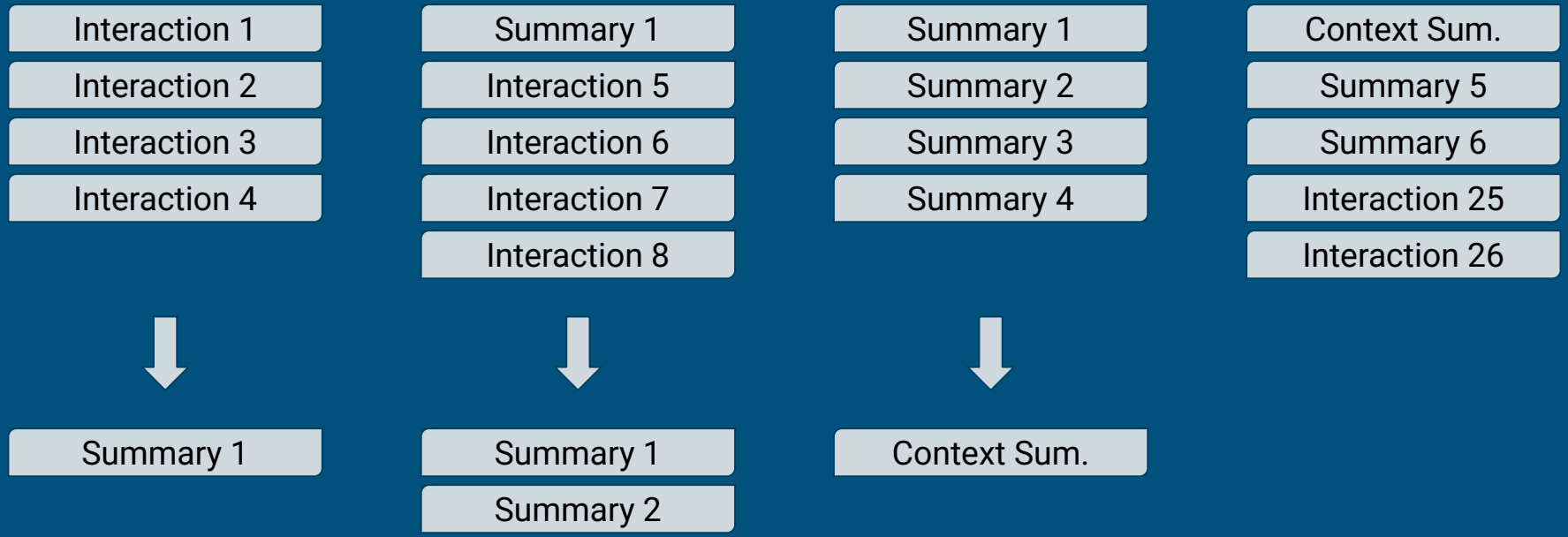
Context size management with LLMs.

Summarize every ***m*** interactions.

Generate a context summary every ***s*** summaries.

https://help.aidungeon.com/faq/the-memory-system

Messages History with Summarizer



RAG for Permanent Knowledge

Use Retrieval Augmented Generation (RAG) to include specific data and material on:

- books that you like
- philosophers that you like
- summaries on extracted/generated topics: work, friendship, goals, wealth, health, different eras, different locations, etc.

It will give you context windows on your documentation depending on the settings: chunk size (500 tokens), overlap (10%), similarity with search, embeddings, etc.

All these settings are really important in the context of a knowledge database for a technical customer support but less relevant when you just want small chunks of “wisdom”.

RAG notes

- I used Qdrant with a specific knowledge base directory.
- The update can be triggered by the backend API.
- I use a very specific non-existing topic for test: the famous green mammoth capable of running at 108 km/h. If I want to check if RAG is working, I can ask something about that.
- If you are using a very general LLM like from OpenAI or Anthropic, they can answer questions without needing your files. It feels like they search online.

https://qdrant.tech

Long-term Memory

How to keep long-term memory?

- Full transcription history?
- Working messages (the summaries state)?
- Summary per session?
- Extracted part? Extracted entities?
- Link between entities? With entity types?

One-to-many Structure Search

Do you need a way to search like that:

- 1 date -> n sessions
- 1 session -> n interactions
- 1 word -> full text search -> n results
- 1 word -> vector search -> n results

Graph structure

- entity extraction
- links between entities
- (date enhancement)

Some relations can change in time, RAG (Vector DB) cannot keep that.

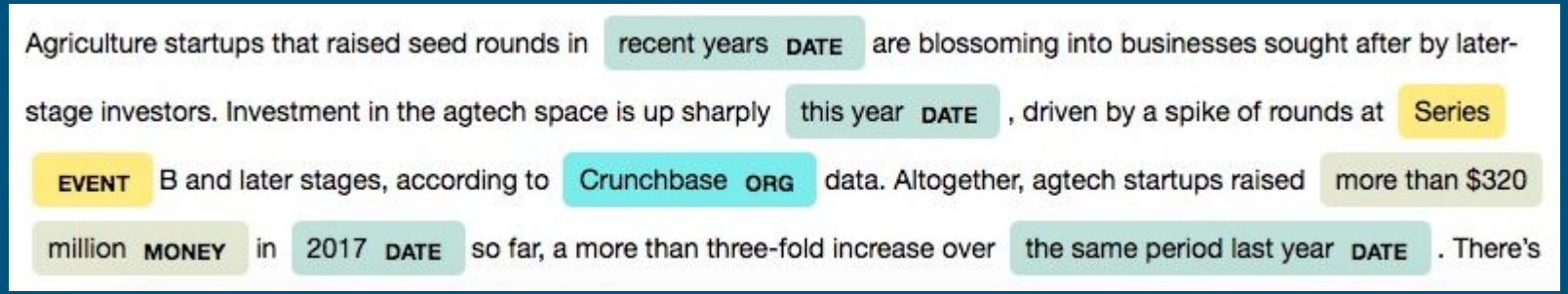
“I loved my hamburger at McDonald” 2025-06-30
“Five Guy’s is my new favorite hamburger place” 2025-07-30
-> The first sentence is no longer “true”.

“I live in Montréal” 2025-01-01
“I move to L.A. next month” Today
-> Live in Montreal will be invalid in a month.

Entity Extraction

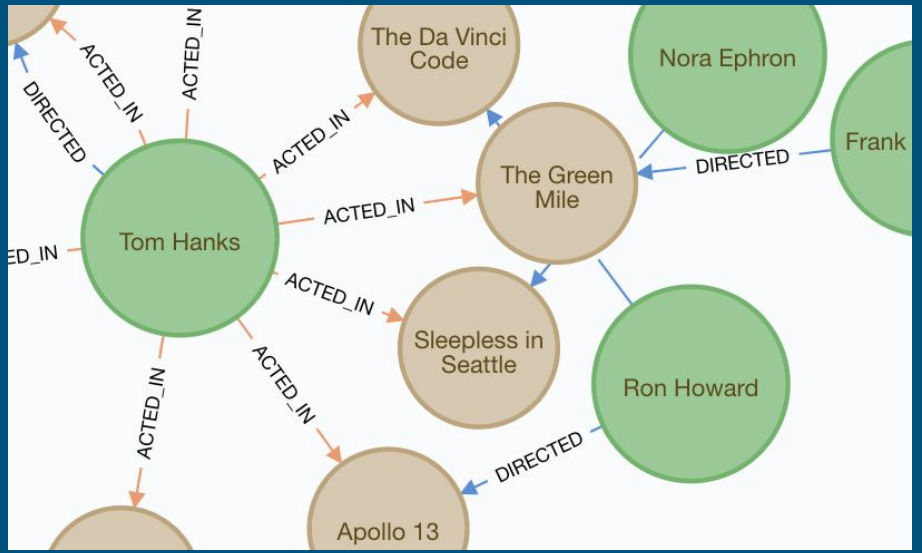
- Person names, organizations, locations, dates and times, money amounts, percentages, product names, email addresses, phone numbers, URLs, etc.
- Support for different languages

spacy.io



Graph Framework

Tools are mainly built on top of Neo4J.
Contender: FalkorDB



https://neo4j.com
https://www.falkordb.com

All Integrated

E.g., Azure GraphRAG, mem0 and Zep (Graphiti)

Azure GraphRAG: didn’t try it. From online feedbacks, more adapted on large **non-moving** data.

mem0: extraction is great. Very expensive if you want to see the graph

Zep: extractions is under mem0, but can be used locally. Good to try graph mem.

getzep.com
https://github.com/getzep/graphiti

Audio Processing

Speech to text (STT) and Text to speech (TTS)

- Similar to privacy considerations: API-based or local.
- Contenders:
  - API: OpenAI (with Whisper) and ElevenLabs
  - Local:
    - Nemo from Nvidia (nvidia/canary-1b-flash): English, French, German & Spanish
    - moonshine: English
- It will add latency. Only ElevenLabs through their Agent part is flawless but not enough control on the messages, the tools, etc. Same for Gemini Live.

https://elevenlabs.io

https://docs.nvidia.com/nemo-framework/
https://github.com/moonshine-ai/moonshine

Voice Activity Detection (VAD)

I prefer manual actually, but automatic detection is nice to use.

A recurring problem is how to deal with stopping the assistant during the play of the text: Do we stop it? Can we restart because I snooze? Does it mean something regarding the evaluation of the answer?

How to deal with long silence because I’m thinking on how to express something?

https://github.com/snakers4/silero-vad

Pipecat

I created and tested everything from scratch: learnt a lot because I spent a lot of time…

If you want to improve latency, you need to switch to WebRTC with almost all models, but managing network, buffer, coordination for parallel operations, etc. is hard and extremely time consuming to recreate from scratch.

Pipecat is a framework to help you with that and examples are really interesting.

https://github.com/pipecat-ai/pipecat

My Actual Setup

Still a lot of WIP and more a compilation of PoCs than a product…

- Access through CLI with Python backend or Pipecat webUI.
- Not an agent because I “manually” manage all messages.
  - System prompt
  - Summaries of the last session
  - RAG on the actual session
  - Memory search on the actual session
  - Graph search on all sessions
- With audio. With VAD and local only in English.

Agency? Agent? Agentic?

Ability to take actions, make decisions, or carry out tasks on behalf of the user.

Control

Future

Guardrails

Better orientation if psychological problems. Actually, the model can support you on “wrong” behaviour, mood, etc. instead of sending you searching for help.

Evaluation (evals)

Better tests on multi-criteria settings. E.g., “this configuration for summarizing with this tool for long-term memory” vs “no summarization with RAG only”. Multi-criteria evaluation is a field on his own…

Questions?

Thank you & take care 🙏