

# Personal Agents and Conversational Memory



Sungeun An



Anna Lisa Gentile

What

Who did you talk to the most  
in the past two weeks?

What

Who did you talk to the most  
in the past two weeks?



# What

Who did you talk to the most  
in the past two weeks?



Does it really “remember”  
what I said?

How

How can we meaningfully  
capture

Personal  
Conversational  
Memory

with Agents?

How

Can we meaningfully capture

**Do We Even Need To?**

Personal  
Conversational  
Memory

with Agents?

How

Can we meaningfully capture  
Personal Agents and Conversational  
Memory  
with Agents

**Do We Even Need To?**



**YES!**

# Who

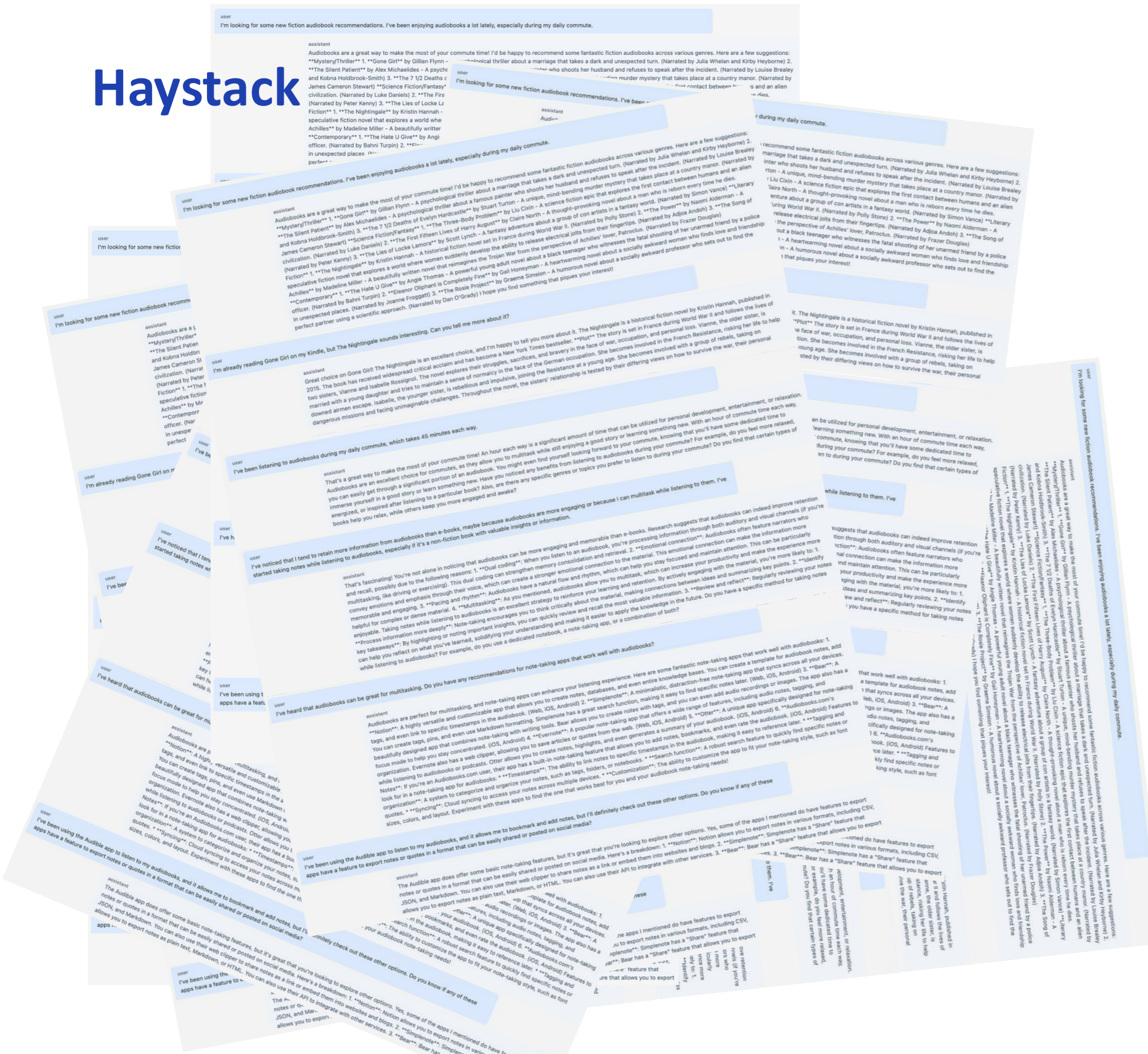
Any user that wants to build their Personal Knowledge from conversation with Agents

- ✗ AI assistants lack persistent personal memory
- ✗ Users repeat context across conversations
- ☑ Enable QA over personal facts, events, preferences, relationships

# Do We Need To?

# LongMemEval Dataset

## Haystack



500 questions across long user/assistant conversation haystacks

- temporal-reasoning (133) – ordering, date arithmetic
- multi-session (133) – aggregation across conversations
- knowledge-update (78) – tracking evolving information
- single-session-user (70) – facts from user messages
- single-session-assistant (56) – facts from assistant replies
- single-session-preference (30) – implicit user preferences

28 domains: shopping, food, health, travel, pets, social media, ...

## Baseline Prompt

The following messages are from your previous conversations with this user across multiple sessions. Treat them as your own conversation history. Use the information from these past sessions to answer the user's final question. If the answer is contained in the conversation history, provide it directly.

```
[user]: <session 1, turn 1>
[assistant]: <session 1, turn 2>
...
[user]: <session N, turn 1>
[assistant]: <session N, turn 2>
...
{question}
```

**Q:** How long is my daily commute to work?  
**A:** 45 minutes each way



# Do We Need To?

*Strong oracle retrieval, but significant drop under realistic haystack conditions*

Question Type	Oracle	Full Haystack
Single Session - User	0.942	0.429
Single Session - Assistant	1.000	0.696
Single Session - Preference	0.466	0.033
Multi-Session	0.812	0.188
Knowledge Update	0.846	0.526
Temporal Reasoning	0.541	0.203
<b>Overall</b>	<b>0.764</b>	<b>0.326</b>

**Oracle**  
Overall: 0.764  
~1.9 sessions, ~22 turns,  
~6.6K tokens per question

**Haystack**  
Overall: 0.326 (57% drop)  
~48 sessions, ~494 turns,  
~122K tokens per question



# Wow

**user**  
I'm looking for some new fiction audiobook recommendations. I've been enjoying audiobooks a lot lately, especially during my daily commute.

**assistant**  
Audiobooks are a great way to make the most of your commute time! I'd be happy to recommend some fantastic fiction audiobooks across various genres. Here are a few suggestions:  
\*\*Mystery/Thriller\*\* 1. \*\*Gone Girl\*\* by Gillian Flynn - A psychological thriller about a marriage that takes a dark and unexpected turn. (Narrated by Julia Whelan and Kirby Heyborne) 2. \*\*The Silent Patient\*\* by Alex Michaelides - A psychological thriller about a famous painter who shoots her husband and refuses to speak after the incident. (Narrated by Louise Brealey and Kobna Holdbrook-Smith) 3. \*\*The 7 1/2 Deaths of Evelyn Hardcastle\*\* by Stuart Turton - A unique, mind-bending murder mystery that takes place at a country manor. (Narrated by James Cameron Stewart) \*\*Science Fiction/Fantasy\*\* 1. \*\*The Three-Body Problem\*\* by Liu Cixin - A science fiction epic that explores the first contact between humans and an alien civilization. (Narrated by Luke Daniels) 2. \*\*The First Fifteen Lives of Harry August\*\* by Claire North - A thought-provoking novel about a man who is reborn every time he dies. (Narrated by Peter Kenny) 3. \*\*The Lies of Locke Lamora\*\* by Scott Lynch - A fantasy adventure about a group of con artists in a fantasy world. (Narrated by Simon Vance) \*\*Literary Fiction\*\* 1. \*\*The Nightingale\*\* by Kristin Hannah - A historical fiction novel set in France during World War II. (Narrated by Polly Stone) 2. \*\*The Power\*\* by Naomi Alderman - A speculative fiction novel that explores a world where women suddenly develop the ability to release electrical jolts from their fingertips. (Narrated by Adjoa Andoh) 3. \*\*The Song of Achilles\*\* by Madeline Miller - A beautifully written novel that reimagines the Trojan War from the perspective of Achilles' lover, Patroclus. (Narrated by Frazer Douglas) \*\*Contemporary\*\* 1. \*\*The Hate U Give\*\* by Angie Thomas - A powerful young adult novel about a black teenager who witnesses the fatal shooting of her unarmed friend by a police officer. (Narrated by Bahni Turpin) 2. \*\*Eleanor Oliphant is Completely Fine\*\* by Gill Honeyman - A heartwarming novel about a socially awkward woman who finds love and friendship in unexpected places. (Narrated by Joanne Froggatt) 3. \*\*The Rosie Project\*\* by Graeme Simsion - A humorous novel about a socially awkward professor who sets out to find the perfect partner using a scientific approach. (Narrated by Dan O'Grady) I hope you find something that piques your interest!

**user**  
I'm already reading Gone Girl on my Kindle, but The Nightingale sounds interesting. Can you tell me more about it?

**assistant**  
Great choice on Gone Girl! The Nightingale is an excellent choice, and I'm happy to tell you more about it. The Nightingale is a historical fiction novel by Kristin Hannah, published in 2015. The book has received widespread critical acclaim and has become a New York Times bestseller. \*\*Plot\*\* The story is set in France during World War II and follows the lives of two sisters, Vianne and Isabelle Rossignol. The novel explores their struggles, sacrifices, and bravery in the face of war, occupation, and personal loss. Vianne, the older sister, is married with a young daughter and tries to maintain a sense of normalcy in the face of the German occupation. She becomes involved in the French Resistance, risking her life to help downed airmen escape. Isabelle, the younger sister, is rebellious and impulsive, joining the Resistance at a young age. She becomes involved with a group of rebels, taking on dangerous missions and facing unimaginable challenges. Throughout the novel, the sisters' relationship is tested by their differing views on how to survive the war, their personal

**user**  
I've been listening to audiobooks during my daily commute, which takes 45 minutes each way.

**assistant**  
That's a great way to make the most of your commute time! An hour each way is a significant amount of time that can be utilized for personal development, entertainment, or relaxation. Audiobooks are an excellent choice for commutes, as they allow you to multitask while still enjoying a good story or learning something new. With an hour of commute time each way, you can easily get through a significant portion of an audiobook. You might even find yourself looking forward to your commute, knowing that you'll have some dedicated time to immerse yourself in a good story or learn something new. Have you noticed any benefits from listening to audiobooks during your commute? For example, do you feel more relaxed, energized, or inspired after listening to a particular book? Also, are there any specific genres or topics you prefer to listen to during your commute? Do you find that certain types of books help you relax, while others keep you more engaged and awake?

**user**  
I've noticed that I tend to retain more information from audiobooks than e-books, maybe because audiobooks are more engaging or because I can multitask while listening to them. I've started taking notes while listening to audiobooks, especially if it's a non-fiction book with valuable insights or information.

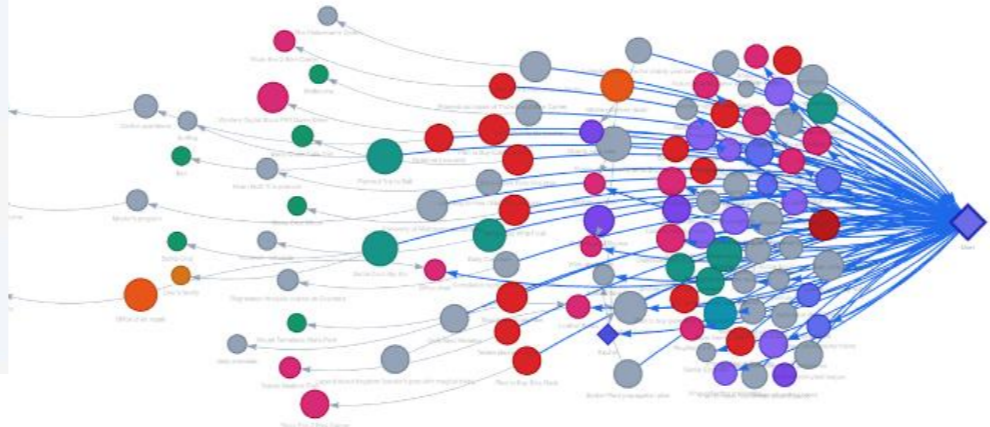
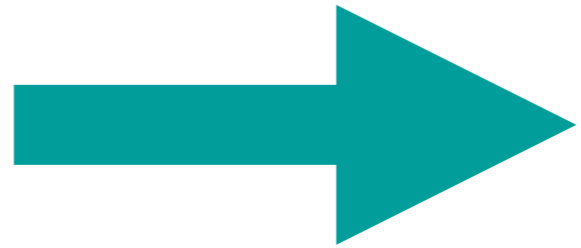
**assistant**  
That's fascinating! You're not alone in noticing that audiobooks can be more engaging and memorable than e-books. Research suggests that audiobooks can indeed improve retention and recall, possibly due to the following reasons: 1. \*\*Dual coding\*\*: When you listen to an audiobook, you're processing information through both auditory and visual channels (if you're multitasking, like driving or exercising). This dual coding can strengthen memory consolidation and retrieval. 2. \*\*Emotional connection\*\*: Audiobooks often feature narrators who convey emotions and emphasis through their voice, which can create a stronger emotional connection to the material. This emotional connection can make the information more memorable and engaging. 3. \*\*Pacing and rhythm\*\*: Audiobooks have a natural flow and rhythm, which can help you stay focused and maintain attention. This can be particularly helpful for complex or dense material. 4. \*\*Multitasking\*\*: As you mentioned, audiobooks allow you to multitask, which can increase your productivity and make the experience more enjoyable. Taking notes while listening to audiobooks is an excellent strategy to reinforce your learning and retention. By actively engaging with the material, you're more likely to: 1. \*\*Process information more deeply\*\*: Note-taking encourages you to think critically about the material, making connections between ideas and summarizing key points. 2. \*\*Identify key takeaways\*\*: By highlighting or noting important insights, you can quickly review and recall the most valuable information. 3. \*\*Review and reflect\*\*: Regularly reviewing your notes can help you reflect on what you've learned, solidifying your understanding and making it easier to apply the knowledge in the future. Do you have a specific method for taking notes while listening to audiobooks? For example, do you use a dedicated notebook, a note-taking app, or a combination of both?

**user**  
I've heard that audiobooks can be great for multitasking. Do you have any recommendations for note-taking apps that work well with audiobooks?

**assistant**  
Audiobooks are perfect for multitasking, and note-taking apps can enhance your listening experience. Here are some fantastic note-taking apps that work well with audiobooks: 1. \*\*Notion\*\*: A highly versatile and customizable app that allows you to create notes, databases, and even entire knowledge bases. You can create a template for audiobook notes, add tags, and even link to specific timestamps in the audiobook. (Web, iOS, Android) 2. \*\*Simplenote\*\*: A minimalist, distraction-free note-taking app that syncs across all your devices. You can create tags, pins, and even use Markdown formatting. Simplenote has a great search function, making it easy to find specific notes later. (Web, iOS, Android) 3. \*\*Bear\*\*: A beautifully designed app that combines note-taking with writing tools. Bear allows you to create notes with tags, and you can even add audio recordings or images. The app also has a focus mode to help you stay concentrated. (iOS, Android) 4. \*\*Evernote\*\*: A popular note-taking app that offers a wide range of features, including audio notes, tagging, and organization. Evernote also has a web clipper, allowing you to save articles or quotes from the web. (Web, iOS, Android) 5. \*\*Otter\*\*: A unique app specifically designed for note-taking while listening to audiobooks or podcasts. Otter allows you to create notes, highlights, and even generates a summary of your audiobook. (iOS, Android) 6. \*\*Audiobooks.com's Notes\*\*: If you're an Audiobooks.com user, their app has a built-in note-taking feature that allows you to add notes, bookmarks, and even rate the audiobook. (iOS, Android) Features to look for in a note-taking app for audiobooks: \*\*Timestamps\*\*: The ability to link notes to specific timestamps in the audiobook, making it easy to reference later. \*\*Tagging and organization\*\*: A system to categorize and organize your notes, such as tags, folders, or notebooks. \*\*Search function\*\*: A robust search feature to quickly find specific notes or quotes. \*\*Syncing\*\*: Cloud syncing to access your notes across multiple devices. \*\*Customization\*\*: The ability to customize the app to fit your note-taking style, such as font sizes, colors, and layout. Experiment with these apps to find the one that works best for you and your audiobook note-taking needs!

**user**  
I've been using the Audible app to listen to my audiobooks, and it allows me to bookmark and add notes, but I'll definitely check out these other options. Do you know if any of these apps have a feature to export notes or quotes in a format that can be easily shared or posted on social media?

**assistant**  
The Audible app does offer some basic note-taking features, but it's great that you're looking to explore other options. Yes, some of the apps I mentioned do have features to export notes or quotes in a format that can be easily shared or posted on social media. Here's a breakdown: 1. \*\*Notion\*\*: Notion allows you to export notes in various formats, including CSV, JSON, and Markdown. You can also use their web clipper to share notes as a link or embed them into websites and blogs. 2. \*\*Simplenote\*\*: Simplenote has a "Share" feature that allows you to export notes as plain text, Markdown, or HTML. You can also use their API to integrate with other services. 3. \*\*Bear\*\*: Bear has a "Share" feature that allows you to export



# IBM Research

Person

- Place
- Person
- Event
- Plant
- RepairAction
- TravelAction
- Habit
- Product
- ProblemEvent
- CreativeWork
- BuyAction
- Action
- OrganizeAction
- CreateAction
- AchieveAction
- Occupation
- Move
- Preference
- Activity
- Agent
- UpdateAction
- PlayAction
- DailyRoutine
- Course
- Organization
- UserAccount
- Vehicle
- Apartment
- ConsumeAction
- PhysicalActivity
- Restaurant
- WriteAction
- Tr
- Schedule
- EducationalOccupationalProgram
- Thread

114 nodes, 117 edges

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Class hierarchy: Daily Routine

Asserted

- owl:Thing
  - Action
    - Achieve Action
    - Assess Action
    - Communicate Action
    - Consume Action
    - Create Action
    - Delete Action
    - Interact Action
    - Move Action
    - Organize Action
    - Play Action
    - Read Action
    - Repair Action
    - Search Action
    - Trade Action
    - Transfer Action
    - Update Action
    - Write Action
  - Activity
    - Creative Activity
    - Daily Living Activity
    - Occupational Activity
    - Physical Activity
    - Recreational Activity
    - Religious Activity
  - Agent
    - Person (schema:Person) = Person (foaf:Person)
    - Person (foaf:Person) = Person (schema:Person)
    - Pet
  - Anatomical Region
  - Causal Type
  - Collection
  - Concept
  - Concept Scheme
  - Creative Work
  - Day of Week
  - Education
  - Employment
  - Entity
  - Ethnic Group
  - Event
  - Food
  - Forum
  - Gender Type
  - Habit
    - Daily Routine
    - Dressing Habit
    - Food Habit
  - Health Record
  - Language
  - Marriage
  - Meal
  - Medical Condition
  - Occupation
  - Organization
  - Person (schema:Person) = Person (foaf:Person)
  - Place
  - Plant
  - Post
  - Preference
  - Product
  - Clothing
  - Vehicle (schema:Vehicle)
  - prov:Agent
  - Religion
  - Schedule

Daily Routine — https://w3id.org/ontobio#DailyRoutine

Annotations Usage

Usage: Daily Routine

Show: this disjoints named sub/superclasses

Found 12 uses of 'Daily Routine'

- Daily Boot Wearing
  - Daily Boot Wearing' Type 'Daily Routine'
- Daily Commute
  - Daily Commute' Type 'Daily Routine'
- Daily Routine
  - 'Daily Routine' rdfs:label "Daily Routine"@en
  - 'Daily Routine' SubClassOf Habit
  - Class: 'Daily Routine'
- has routine
  - has routine' Range 'Daily Routine'

Description: Daily Routine

Equivalent To +

SubClass Of +  
Habit

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +  
'Daily Boot Wearing'  
'Daily Commute'

Target for Key +

Disjoint With +

Disjoint Union Of +

# Ontology 101

## The 3 Fundamental Rules

### R1. No single correct model

*There is no one correct way to model a domain - there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.*

### R2. Iterative development

Ontology development is necessarily iterative

### R3. Concepts close to domain objects and relationships

*Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain.*

# Ontology 101

## The 7 Steps

1. **Determine domain and scope** - Define what the ontology covers, its purpose, the types of questions it should answer, and who will use it. Use **competency questions** as a litmus test.
2. **Consider reusing existing ontologies** - Survey existing vocabularies and ontologies that cover (parts of) your domain. Refine and extend them rather than building from scratch.
3. **Enumerate important terms** - Brainstorm a flat list of all relevant terms (nouns, verbs, properties) without worrying yet about class vs. property distinctions or overlap.
4. **Define classes and the class hierarchy** - Select the terms that represent independent concepts and organize them into a subclass/superclass taxonomy (top-down, bottom-up, or combination).
5. **Define properties (slots)** - The remaining terms from Step 3 become properties of classes: intrinsic attributes, extrinsic attributes, parts, and relationships to other individuals.
6. **Define facets (value restrictions)** - For each property, specify its value type (string, number, instance), cardinality (single vs. multiple), domain (which classes it applies to), and range (allowed values or classes).
7. **Create instances** - Populate the ontology with individual instances: choose a class, create an individual, and fill in its property values. **This is where the ontology becomes a knowledge base.**

# Personal Conversational Memory (PCM) Ontology

Following Ontology 101 methodology:

- Step 1: 500 LongMemEval questions as competency questions
- Step 2: Reuse of existing ontologies
  - schema: <http://schema.org/>
  - ob: <https://w3id.org/ontobio#>
    - reuses bio: <http://purl.org/vocab/bio/0.1/> and opencare: <https://w3id.org/opencare#>
  - foaf: <http://xmlns.com/foaf/0.1/>
  - prov: <http://www.w3.org/ns/prov#>
  - sioc: <http://rdfs.org/sioc/ns#>
- Step 3: Extract terminology (spaCy NLP): 1,286 unique nouns, 265 verbs
- Step 4-7:
  - automated the mapping to existing classes
  - 5 gap classes identified → added in PCM ontology pcm: <https://w3id.org/pcm>
  - 'how many' questions → need countable typed instances
  - 110 temporal ordering questions → need date properties on all events
  - ...

# PCM Ontology

1. Person & Identity  
(*FOAF, Schema.org, OntoBio*)
2. Family Relationships  
(*OntoBio*)
3. Social Relationships  
(*Relationship, OntoBio*)
4. Actions  
(*Schema.org*)
5. Events  
(*Schema.org, BIO*)
6. Travel  
(*OntoBio*)
7. Health  
(*OpenCare, OntoBio*)
8. Work & Career  
(*Schema.org, OntoBio, BIO, CV*)
9. Education  
(*OntoBio, CV*)
10. Habits & Routines  
(*OntoBio*)
11. Activities  
(*OntoBio*)
12. Food & Meals  
(*OntoBio*)
13. Personality & Traits  
(*OntoBio*)
14. Places & Residence  
(*Schema.org, OntoBio*)
15. Things & Products  
(*Schema.org*)
16. Causality & Sequencing  
(*OntoBio*)
17. Provenance  
(*PROV-O*)
18. Concepts  
(*SKOS*)
19. PCM Custom Extensions  
Preference, ProblemEvent,  
Pet, Plant, Clothing,  
Collection,  
SocialMediaAccount
20. Conversation Structure  
(*SIOC*)  
Messages represented as  
sioc:Post and prov:Entity

# KG Extraction Pipeline

- Per-question **extraction**: one KG per question's context
- LLM (Claude Sonnet 4.5) generates Turtle from each message chunk
- **Ontology schema + extraction rules** injected as prompt context
- **Provenance**: every entity linked to source via `prov:wasDerivedFrom`
- **Validation**: *rdflib* parses output, retry on TTL syntax errors (up to 3×)

**LongMemEval\_Oracle**

61,431 triples (~30 min)

**LongMemEval\_S\_cleaned**

473,937 triples (~2 hours)

# KG Extraction Pipeline

## Stage 1 System Prompt: LLM-Based Triple Extraction

You are a personal knowledge graph extraction engine.

Given a natural language message from a user named {agent\_name}, extract structured knowledge as RDF triples in Turtle format. **Rules**

1. The primary agent is ALWAYS "{agent\_name}". Use URI pcm:{agent\_id} for them.
2. Output ONLY valid Turtle (TTL) — no explanation, no markdown fences.
3. Always start with these prefixes: @prefix ... (omitted for brevity)
4. All resource URIs MUST use the pcm: namespace.
5. Use the most specific type available. Dual-type with Schema.org actions and domain-specific classes where applicable (e.g. a repair is schema:RepairAction + pcm:ServiceEvent; a trip is schema:TravelAction + ob:Travel).
6. Extract dates from text. Partial dates like "3/22" or "March 15th" should be resolved to the current year. Format as "YYYY-MM-DD"^^xsd:date.
7. Every event/action MUST include schema:agent. If the message is in first person, the agent is pcm:{agent\_id}.
8. If an event is caused by another, link them with ob:causalContext.
9. Extract ALL meaningful knowledge: events, objects, preferences, habits, plans, relationships, facts, family/social ties, health, travel, food, traits.
10. For completed actions use schema:CompletedActionStatus. For planned actions use schema:PotentialActionStatus.
11. Every pcm:ProblemEvent MUST include pcm:affects. Every pcm:ServiceEvent and schema:RepairAction MUST include schema:object.
12. NO ISOLATED NODES: Every entity MUST connect to at least one other entity via a relationship property.
13. LABELS: Every individual MUST have an rdfs:label with a short, human-readable name.
14. ENTITY UPDATES: When multiple messages refer to the same real-world entity, reuse the SAME URI. Do NOT create duplicates.
15. RELATIONSHIPS: Use rel: for social relationships (rel:friendOf, rel:colleagueOf) and ob: for family (ob:hasFather, ob:hasSibling, ob:hasSpouse).
16. HEALTH: Use opencare: for health records (opencare:HealthRecord, opencare:diagnosedWith, opencare:hasSymptom).
17. TRAVEL: Use ob:Travel with ob:travelFrom, ob:travelTo, ob:modeOfTransport. Also dual-type as schema:TravelAction.
18. HABITS & FREQUENCY: Use ob:Habit and subclasses. Attach schema:Schedule via schema:eventSchedule with schema:repeatFrequency (ISO 8601 duration).
19. PETS: Use pcm:Pet linked via pcm:hasPet with pcm:petSpecies.
20. SOCIAL MEDIA: Use pcm:SocialMediaAccount linked via pcm:hasSocialAccount.
21. CLOTHING: Use pcm:Clothing linked via pcm:wears.
22. PLANTS: Use pcm:Plant linked via pcm:grows with pcm:plantSpecies.
23. COLLECTIONS: Use pcm:Collection linked via pcm:hasCollection.

### Available Types and Properties

[Ontology reference catalog from PCM ontology] (omitted for brevity)

### Example

Input: "I just bought a silver Honda Civic on February 10th."

Output:

```
pcm:annalisa a schema:Person, foaf:Person ;
rdfs:label "Annalisa"@en ;
schema:name "Annalisa" .

pcm:silver-honda-civic a schema:Vehicle ;
rdfs:label "Silver Honda Civic"@en ;
schema:name "Silver Honda Civic" ;
schema:color "silver" ; schema:brand "Honda" ; schema:model "Civic" .

pcm:buy-honda-civic-2026-02-10 a schema:BuyAction ;
rdfs:label "Purchase of silver Honda Civic"@en ;
schema:name "Purchase of silver Honda Civic" ;
schema:startDate "2026-02-10"^^xsd:date ;
schema:agent pcm:annalisa ;
schema:object pcm:silver-honda-civic ;
schema:actionStatus schema:CompletedActionStatus .
```

# KG Extraction Coverage

We search the KG for the expected answer string using a fuzzy matching strategy applied to literals (excluding provenance, conversation structure, and ontology metadata)

- (i) exact normalized match
- (ii) substring containment in either direction
- (iii) numeric match on extracted numbers
- (iv) keyword overlap requiring at least 50% of content words to appear in a single literal

Answers are found in the extracted KGs : **oracle 66.2%, full haystack 76.0%**

Coverage is an approximation

- **count** (e.g., “How many charity events did I participate in?”) require aggregating multiple entities
- **duration** (e.g., “7 days”) require computing the difference between two dates
- **entity** answers (proper nouns, product names, place names) and **preference** answers (“Italian food”, “Shell gas station”) are well-suited to literal matching

# KG Extraction Coverage (bonus experiment)

		Text2KGBench						
Dataset	Model	P	R	F1	OC	RH	SH	OH
DBpedia-WebNLG	Vicuna-13B [8,20]	0.34	0.27	0.30	0.93	0.07	0.12	0.28
	Alpaca-LoRA-13B [26,15,20]	0.32	0.23	0.25	0.91	0.09	0.16	0.38
	<b>PCM (ours)</b>	<b>0.50</b>	<b>0.45</b>	<b>0.47</b>	<b>0.99</b>	<b>0.01</b>	<b>0.01</b>	<b>0.10</b>
Wikidata-TekGen	Alpaca-LoRA-13B [26,15,20]	0.32	0.26	0.27	0.88	0.12	0.19	0.18
	Vicuna-13B [8,20]	0.38	0.35	0.35	0.84	0.13	0.17	0.17
	<b>PCM (ours)</b>	<b>0.38</b>	<b>0.38</b>	<b>0.37</b>	<b>0.92</b>	<b>0.08</b>	<b>0.03</b>	<b>0.09</b>

OC = Ontology Conformance

RH/SH/OH = relation/subject/object hallucination

Mihindukulasooriya, N., Tiwari, S., Enguix, C.F., Lata, K.

**Text2KGBench**: A benchmark for ontology-driven knowledge graph generation from text.

ISWC 2023. [https://doi.org/10.1007/978-3-031-47243-5\\_14](https://doi.org/10.1007/978-3-031-47243-5_14)

# Query Generation Pipeline (not mature)

Retrieval: 92-94% of queries return something

Correctness gap: only 10-14% of answers are correct

Error categories:

1. KG coverage gap (34%)

answer not extracted from conversation text

2. Wrong entity selection:

SPARQL finds related but wrong entity

3. Missing computation:

Dates returned raw, counts not aggregated

Post-processing helps (+6pp) but LLM non-determinism limits gains

# Query Generation Pipeline (not mature)

## Stage 2 System Prompt: NL → SPARQL Generation

You are a SPARQL query generator for a personal knowledge graph. Given a natural language question, generate a SPARQL SELECT query that retrieves the SPECIFIC answer from the graph.

### Rules

1. Output ONLY the SPARQL query — no explanation, no markdown fences.
2. Always use these prefixes: schema:, ob:, foaf:, prov:, xsd:, pcm:, rdfs:, rel:, opencare:, skos:, bio:, cv:, sioc:.
3. The primary person in the graph is pcm:user (a schema:Person).
4. Use OPTIONAL for properties that may not exist.
5. Return the SPECIFIC value that answers the question — not just the entity.
6. Order results by date when temporal information is relevant.
7. Limit results to 20 unless the question asks for all.

### CRITICAL: Type-first querying strategy

- Map natural language concepts to ontology types — do NOT rely on keyword matching in schema:name:
  - "issue/problem" → schema:Event
  - "bought/purchased" → schema:BuyAction
  - "trip/travel" → ob:Travel / schema:TravelAction
  - "habit/routine" → ob:Habit (ob:DailyRoutine, ob:FoodHabit)
  - "health/doctor" → opencare:HealthRecord / opencare:Disease
  - "friend/colleague" → rel:friendOf, rel:colleagueOf
  - "family/parent/sibling" → ob:hasParent, ob:hasSibling, ob:hasSpouse
  - "food/meal" → ob:Meal, ob:FoodItem, ob:FoodHabit
  - "lives/resided" → ob:Residence, ob:residedIn
- Use rdfs:subClassOf\* to match all subtypes:  
`?entity rdf:type ?type . ?type rdfs:subClassOf* schema:Event .`
- Use subqueries for temporal references:  
`{{ SELECT (MIN(?d) AS ?refDate) WHERE {{ ... }} }}`  
`FILTER(?date > ?refDate)`
- Only fall back to FILTER(CONTAINS(...)) on schema:name to distinguish entities of the *same* type.

### Common mistakes to avoid

- Do NOT search for events by keyword when you can match by type.
- “how many X?” → use COUNT(DISTINCT ?entity).
- “how many days/weeks?” → return the relevant dates; post-processing computes the difference.
- “who did I X with?” → return schema:participant, not schema:agent.
- For price questions, SELECT schema:price.
- For preferences, check foaf:interest or keyword matches in schema:name/schema:description.

# Query Generation Pipeline (not mature)

## LongMemEval\_Oracle

Avg sessions/question: 2.7

Avg triples/KG: ~120

Retrieval (OK%): 91.8%

Correctness: 14.0%

Best: single-session-user 48.6%

Hardest: multi-session 2.3%

## LongMemEval\_S\_cleaned

Avg sessions/question: 47.7

Avg triples/KG: ~950

Retrieval (OK%): 93.8%

Correctness: 9.8%

Best: single-session-user 45.7%

Hardest: multi-session 0.8%

Question Type

single-session-user (70) - 3

Correctness

Correct

Knowledge Graph

V How long is my daily commute to work?

**118b2229** — 5711 triples, 113 entities single-session-user CORRECT OK

**Q:** How long is my daily commute to work?

**A:** 45 minutes each way

**Got:** entity=pcm:daily-commute; type=ob:DailyRoutine; rdfs:label=Daily Commute; schema:agent=pcm:user; schema:description=45 minutes each way commute; schema:eventSchedule=pcm:commute-schedule; schema:name=Daily Commute; ob:activityDuration=PT45M

Graph
SPARQL
Ask (NL)
Entities
TTL
Messages

```

SELECT ?name ?description ?duration ?t ?routine
WHERE {
  {
    ?routine rdf:type ?rType .
    ?rType rdfs:subClassOf* ob:DailyRoutine .
    ?routine schema:name ?name .
  }
  ?routine a ?t .
  FILTER(CONTAINS(LCASE(str(?name)), "commute"))
  OPTIONAL { ?routine schema:description ?description }
  OPTIONAL { ?routine ob:activityDuration ?duration }
} UNION {
  ?habit rdf:type ?hType .
  ?hType rdfs:subClassOf* ob:Habit .
  ?habit schema:name ?name .
  FILTER(CONTAINS(LCASE(str(?name)), "commute"))
  OPTIONAL { ?habit schema:description ?description }
  OPTIONAL { ?habit ob:activityDuration ?duration }
} UNION {
  ?action rdf:type ?aType .
  ?aType rdfs:subClassOf* schema:TravelAction .
  ?action schema:name ?name .
  FILTER(CONTAINS(LCASE(str(?name)), "commute") || CONTAINS(LCASE(str(?name)), "work"))
  OPTIONAL { ?action schema:description ?description }
  OPTIONAL { ?action ob:activityDuration ?duration }
}
        
```

114 nodes, 117 edges

Run Query
Insert Prefixes

description	duration	name	routine	t
45 minutes each way commute	PT45M	Daily Commute	<a href="https://w3id.org/pcm/daily-commute">https://w3id.org/pcm/daily-commute</a>	<a href="http://schema.org/TravelAction">http://schema.org/TravelAction</a>
45 minutes each way commute	PT45M	Daily Commute	<a href="https://w3id.org/pcm/daily-commute">https://w3id.org/pcm/daily-commute</a>	<a href="https://w3id.org/ontobio#DailyRoutine">https://w3id.org/ontobio#DailyRoutine</a>
Listens to audiobooks during daily commute		Audiobook Listening During Commute		
45 minutes each way commute	PT45M	Daily Commute		

118b2229 — 5711 triples, 113 entities single-session-user **CORRECT** OK

**Q:** How long is my daily commute to work?

**A:** 45 minutes each way

**Got:** entity=pcm:daily-commute; type=ob:DailyRoutine; rdfs:label=Daily Commute; schema:agent=pcm:user; schema:description=45 minutes each way commute; schema:eventSchedule=pcm:commute-schedule; schema:name=Daily Commute; ob:activityDuration=PT45M

Annotations Usage

**Annotations: Daily Commute**

Annotations +

- rdfs:label** [language: en]  
Daily Commute
- activityDuration**  
PT45M
- prov:wasDerivedFrom**  
◆ [answer\\_40a90d51\\_1](#)
- prov:wasDerivedFrom**  
◆ [answer\\_40a90d51\\_9](#)
- schema:agent**  
◆ [User](#)
- schema:description**  
45 minutes each way commute
- schema:eventSchedule**  
◆ ['Daily Commute Schedule'](#)
- schema:name**  
Daily Commute

Annotations Usage

**Annotations: answer\_40a90d51\_9**

Annotations +

- dcterms:description**  
I've been listening to audiobooks during my daily commute, which takes 45 minutes each way.
- dcterms:identifier**  
answer\_40a90d51\_9
- role**  
user
- sioc:content**  
I've been listening to audiobooks during my daily commute, which takes 45 minutes each way.
- sioc:has\_container**  
◆ ['Thread 118b2229'](#)
- sioc:has\_creator**  
◆ [User](#)
- sioc:has\_reply**  
◆ [answer\\_40a90d51\\_10](#)
- turnIndex** [type: xsd:integer]  
89

# My Own Real Data

## PCM Explorer

Knowledge Graph

All Sessions (combined)

all — 171053 triples, 75 entities  
75 session files combined

ES: pcm\_memory (10960 docs)

Graph Entities TTL Messages **Ask** ES Index

Ask a question about this knowledge graph

actons, list them all

Ask

Uses NL→SPARQL generation. Results come from the knowledge graph, not the LLM.

Generated SPARQL — edit and re-run

Re-run

```
SELECT DISTINCT ?name ?actionType ?description ?startDate
WHERE {
  ?action rdf:type ?actionType .
  ?actionType rdfs:subClassOf* schema:Action .
  OPTIONAL { ?action schema:name ?name }
  OPTIONAL { ?action schema:description ?description }
  OPTIONAL { ?action schema:startDate ?startDate }
}
ORDER BY ?actionType ?startDate
```



**Answer:** Based on the evidence provided, here are all the actions listed: 1. 124 PC members accepted invitations within 24 hours 2. 16 of the 124 are SPC 3. 24hr extension agreed due to Overleaf outage 4. Closed last track with 79 posters and demos received 5. Bo Xiong winning the SWSA Distinguished Dissertation Award 2025

Results (371 rows)

description

124 PC members accepted invitations within 24 hours

16 of the 124 are SPC

24hr extension agreed due to Overleaf outage

Closed last track with 79 posters and demos received

Bo Xiong winning the SWSA Distinguished Dissertation Award 2025

Paper that won the Ten-Year Paper award

Secured 9 additional PC members out of 28 potential PC members contacted

GRAPH STATS

Triples 97,247

TOP ENTITY TYPES

Person 254

CreativeWork 213

CommunicateAction 187

UpdateAction 141

Event 110

# Maturity



## KB population

- Can be considered fairly stable at this point
  - We approximated the coverage for the considered use case at ~76%



## Query Generation

- In the cooking – sparql+ElasticSearch



## Generalization

- test on different use cases
- from a cognitive perspective:
  - what do we CHOSE to forget and what to memorize?

# Personal Agents and Conversational Memory



Sungeun An



Anna Lisa Gentile