Knowledge Elicitation Through Conversational AI

A Research Proposal for LLM Extension

Author: Leszek J. Cierniak
Email: leszek.cierniak@gmail.com
Date: April 2026

Abstract

Large Language Models (LLMs) demonstrate remarkable reasoning capabilities but suffer from static knowledge bases frozen at training time and inability to persistently accumulate new information from interactions. Despite progress in memory-augmented LLMs, no existing system provides a structured, interactive framework for resolving the inevitable conflicts that arise when humans teach knowledge to an AI through dialogue. This research proposal presents a novel cognitive architecture for knowledge elicitation where a frozen LLM builds and maintains an external knowledge graph through natural language dialogue, starting from *tabula rasa*. We introduce the first hierarchical, interactive conflict resolution taxonomy specifically designed for dialogue-driven knowledge-graph construction in frozen-LLM architectures, systematically addressing temporal state changes, cardinality violations, entity canonicalization conflicts, and logical contradictions. The architecture decouples reasoning (LLM) from memory (hybrid vector store and property graph), enabling model-agnostic operation while maintaining full explainability through externalized knowledge representation. This work advances Explainable AI by making the system's mental model fully inspectable, correctable, and transferable across LLM backends.

Keywords: Knowledge Elicitation, Large Language Models, Knowledge Graphs, Explainable AI, Human-in-the-Loop Learning, Retrieval-Augmented Generation, Conflict Resolution

1. Introduction

The rapid evolution of Large Language Models has demonstrated unprecedented capabilities in natural language understanding and generation. However, these systems face fundamental architectural limitations when deployed in dynamic, knowledge-intensive environments where information evolves continuously and must persist beyond individual conversational sessions.

1.1. The Challenge of Dynamic Knowledge Management

Consider an AI assistant supporting collaborative project management. Such a system must track evolving team assignments, shifting deadlines, and emerging task dependencies. A conventional LLM-based chatbot can discuss project management concepts fluently but cannot persistently remember that "Alice assumed leadership of Project Alpha on Tuesday" or that "the database migration was postponed following vendor delays." Once the conversation context window is exhausted or the session terminates, all acquired knowledge vanishes—a phenomenon we term *conversational amnesia*.

This limitation becomes critical in domains requiring:

- Long-term relationship tracking across multiple interactions
- Temporal reasoning about state changes and event sequences
- Collaborative knowledge building among multiple human contributors
- Auditable decision-making processes with full knowledge provenance

1.2. Research Objectives

This proposal addresses a fundamental question in human-AI collaboration: **How can we enable a frozen LLM to acquire, structure, and persistently retain domain-specific knowledge purely through natural language dialogue, while maintaining full explainability and human oversight?**

We propose transforming the LLM from an end-to-end solution into a *reasoning engine* operating over an external, structured knowledge base. The system's external memory begins as a *tabula rasa*. While the reasoning engine utilizes the LLM's pre-trained linguistic patterns and general world ontologies, it is strictly prohibited from asserting domain-specific facts not present in the external **Structural Core**, ensuring the 'world model' is built exclusively through verified interaction.

1.3. Primary Contributions

This research will deliver three fundamental innovations:

First, we develop a **hierarchical conflict resolution framework** providing the first systematic, interactive taxonomy for managing four distinct types of contradictions emerging during dialogue-driven knowledge graph construction.

Second, we demonstrate **cognitive architecture with full model-memory decoupling**, enabling the reasoning engine to be replaced without knowledge loss while maintaining complete explainability through externalized semantic memory.

Third, we design an **active elicitation protocol** enforcing retrieval-constrained generation, preventing hallucination by triggering targeted clarification questions when encountering unknown entities.

1.4. Innovation and Uniqueness

This work advances **Explainable AI (XAI)** research by emphasizing human-like, interpretable reasoning. Instead of relying solely on opaque weight adjustments within neural networks, the system externalizes acquired knowledge into readable format. This paradigm shift moves from *implicit training* to *explicit teaching*, aligning machine learning workflows more closely with human pedagogy.

Key Innovations

- **Explicit Knowledge Representation:** Every learned concept is decoupled from model weights and stored in a structured, interpretable Knowledge Graph, directly addressing the "black box" problem
- **Dialogue-Based Acquisition:** The knowledge base begins *tabula rasa*, acquiring domain expertise exclusively through interaction without requiring domain-specific pre-training
- **Active Knowledge Elicitation:** The agent proactively detects gaps and ambiguities in its external knowledge graph, generating targeted clarification questions
- **Cognitive Decoupling (Model Agnosticism):** Modular design allows the reasoning engine to be replaced (e.g., from GPT-4 to Llama) without loss of acquired knowledge

1.5. Research Questions

This proposal investigates three core questions:

RQ1: Can a frozen LLM build a semantically coherent knowledge graph purely from conversational input without domain-specific fine-tuning?

RQ2: What categories of conflicts arise during dialogue-based knowledge acquisition, and what resolution strategies preserve both data consistency and human agency?

RQ3: How does explicit knowledge externalization impact the explainability and trustworthiness of LLM-based systems in knowledge-intensive tasks?

2. Problem Statement and Related Work

2.1. Fundamental Limitations of Current LLMs

Large Language Models face two primary limitations in dynamic environments:

Static Knowledge Base: Models cannot update internal weights to reflect new information without expensive fine-tuning or complete retraining.

Context Window Constraints: Despite increasing context lengths, conversation history remains ephemeral. Once a session ends or the context limit is reached, acquired information is lost—catastrophic forgetting within a session context.

This research addresses the challenge of **Episodic to Semantic Memory Consolidation**, creating a system where the LLM functions as a reasoning engine rather than a storage container, utilizing an external, structured knowledge base that evolves in real-time.

2.2. Related Research

The proposed system builds on several research strands in LLM agents, external memory, and knowledge graph construction.

Memory-Augmented Agents

Recent work (Park et al., 2023; Liu et al., 2024) externalizes episodic or long-term memory to overcome context window limitations. MemGPT (Packer et al.) introduced the OS metaphor with hierarchical memory, but employs only flat key-value storage and lacks structured conflict handling. HippoRAG (Liu et al.) uses neurobiologically inspired retrieval but does not support **interactive human-in-the-loop learning** or **structured conflict resolution**—central to this proposal's novelty.

LLM-Driven Knowledge Graph Construction

Recent advances (Pan et al.; He et al.; Li et al.) demonstrate LLM capabilities for knowledge graph construction. GraphRAG (Edge et al.) leverages graph structures for improved retrieval but relies on static, pre-extracted facts or large context windows for summarization, lacking dynamic, interactive mechanisms for human-guided knowledge integration or conflict resolution. Most approaches perform one-shot or iterative extraction into a knowledge graph but either assume a fixed ontology or perform only automatic merging without fine-grained human oversight.

Active Retrieval and Clarification

Research on active retrieval (Asai et al.; Press et al.) explores when models should retrieve information or ask questions, but focuses on fact-checking or search rather than persistent structured memory population.

2.3. Research Gap

To our knowledge, no existing system combines:

- Full **property-graph memory** with dynamic ontology evolution
- Systematic interactive **conflict resolution** across multiple conflict categories
- Retrieval-constrained **active elicitation** protocol
- Complete **model-memory decoupling** in a single **frozen-LLM** architecture

The table below highlights how the proposed framework closes these gaps:

System	Temporal Updates	Cardinality Handling	Interactive Canonicalization	Logical Constraints	Dynamic Ontology
MemGPT (2023)	✗	✗	✗	✗	✗
GraphRAG (2024)	automatic only	✗	✗	✗	✗
HippoRAG (2024)	automatic only	✗	✗	✗	✗
This work	**✓ interactive**	**✓ interactive**	**✓ interactive**	**✓ interactive**	**✓**

3. Proposed Architecture

The system follows a **Controller-Reasoner-Storage** pattern, designed to decouple the probabilistic nature of the LLM from deterministic requirements of database management.

3.1. Architectural Components

3.1.1. The Orchestrator (Controller)

A deterministic runtime environment serving as the system's central nervous system, ensuring LLM adherence to architectural constraints.

Responsibilities:

- **I/O Management:** Sanitizes user input and formats system output
- **Tool Binding:** Operationalizes APIs for vector store and graph database
- **Prompt Engineering:** Dynamically assembles prompts by injecting retrieved context into system prompts before LLM inference
- **Context Pruning:** Implements sub-graph sampling and relevance scoring to ensure that when the Knowledge Graph grows large, only the most contextually relevant *hops* are injected into the Reasoner's window, preventing context overflow.

3.1.2. The Cognitive Core (Reasoner)

A pre-trained, frozen LLM with immutable parameters, functioning as a stateless reasoning unit. The system operates in two distinct modes:

Synthesizer Mode: The model generates natural language responses grounded by "Retrieval Context" to minimize hallucinations.

Extractor Mode: The model functions as a semantic parser, analyzing user input to identify *Entities* (nodes) and *Relationships* (edges) for storage.

Example: "Project Alpha is delayed." → `{"head": "Project Alpha", "relation": "status", "tail": "delayed"}`

3.1.3. The Hybrid Memory System

To address the *tabula rasa* requirement and prevent catastrophic forgetting, the system employs a dual-process memory architecture:

Semantic Buffer (Vector Store):

- Unstructured, associative memory
- Dense vector index for embeddings
- Stores conversational "stream of consciousness"
- Enables fuzzy similarity search and context recall

Structural Core (Knowledge Graph):

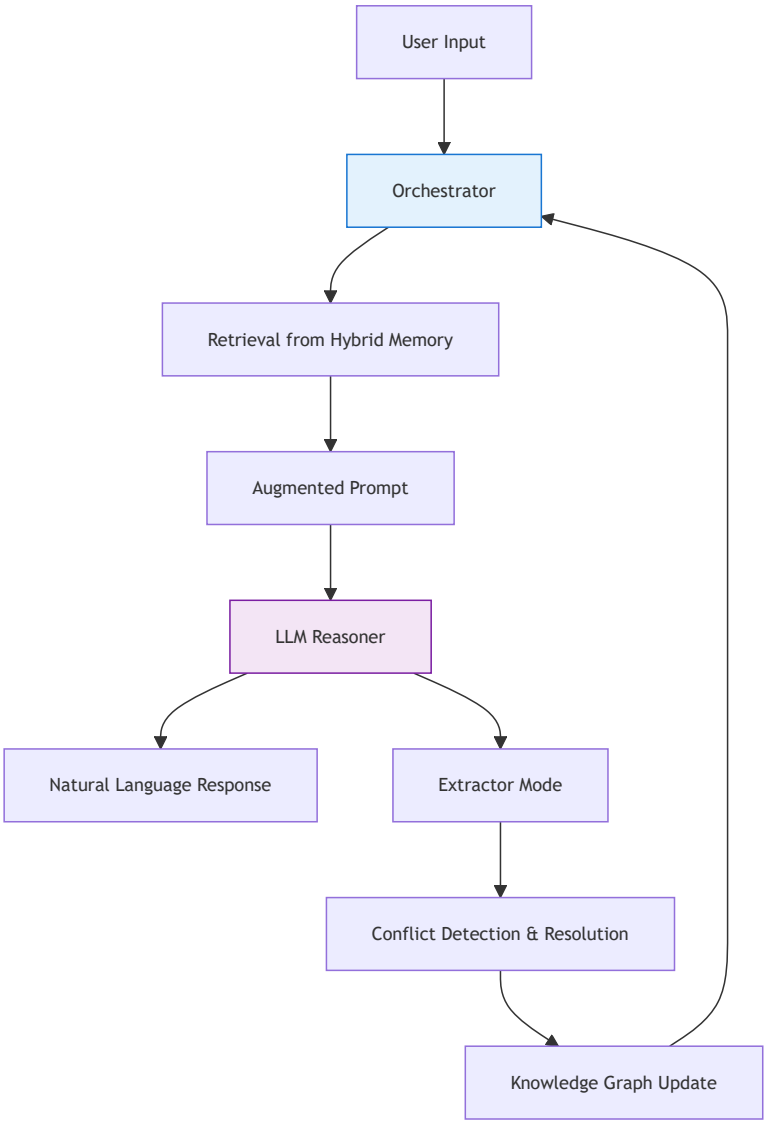
- Structured, logic-based memory
- Property graph representation: $G = (V, E)$ where V represents entities and E represents predicates
- Stores the "World Model" with entities and relationships
- **Dynamic Ontology:** No pre-defined schema; structure evolves as conversation progresses, supporting *tabula rasa* learning

3.2. The Cognitive Cycle

Data flows through the system in a four-step recursive loop:

1. **Retrieval (Recall):** Orchestrator scans vector store and knowledge graph for terms related to current user input
2. **Augmentation (Context):** Retrieved facts are injected into LLM's context window
3. **Inference (Response):** LLM generates response based on augmented context
4. **Consolidation (Learning):**
    - Interaction logged in vector store

- Extractor parses interaction for new facts
- Knowledge graph updated (nodes created/merged)



4. Conflict Detection and Resolution Framework

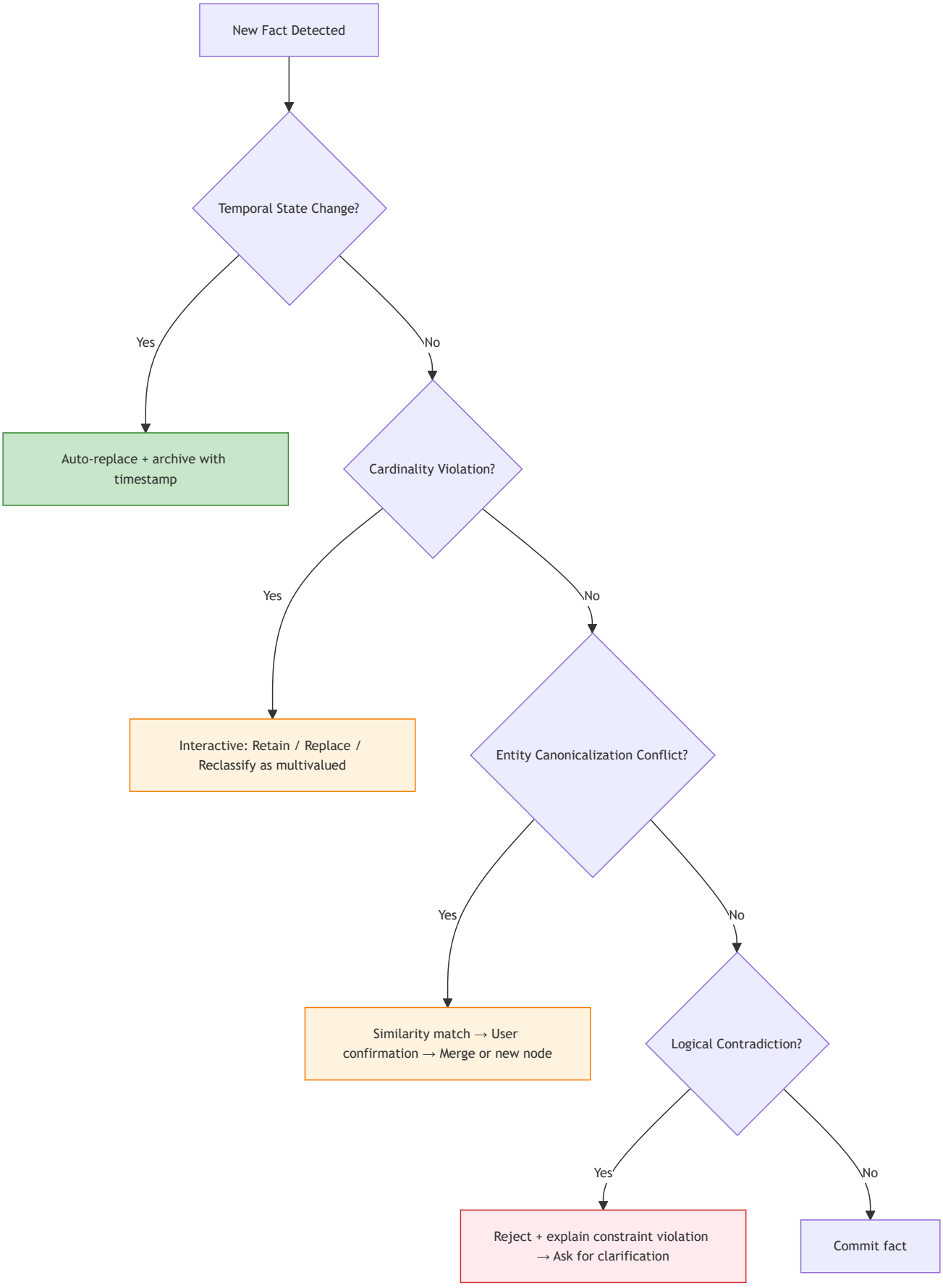
This proposal introduces a **Hierarchical Conflict Taxonomy** — the first systematic, interactive framework specifically designed for LLM-driven, dialogue-based knowledge graph construction.

LLM-Driven knowledge acquisition inevitably produces conflicting assertions. We classify conflicts into five operationally distinct but potentially overlapping conflict categories, resolved through a prioritized evaluation order, each with distinct resolution policies.

Formally, the Orchestrator evaluates every newly extracted fact f against the existing state of the Knowledge Graph G using a detection function:

$$D(f, G) \to \{C_{temporal}, C_{cardinal}, C_{canon}, C_{logical}, \emptyset\}$$

A non-null return triggers the specific resolution pipeline associated with that conflict category, ensuring that no inconsistent data is committed to the Structural Core.



4.1. Temporal State Changes

Definition: Relationships where object values are expected to evolve over time (e.g. *status, location, assignee, deadline*).

Resolution Policy: Automatic replacement with archival

- Previous value archived with timestamp

- New value committed without user interruption
- Assumes recent information supersedes older data
- Updates logged in conversation history for transparency

Rationale: Minimizes conversational friction for volatile attributes while maintaining audit trail.

4.2. Cardinality Violations

Definition: Relationships enforcing single-value constraints by semantic nature (e.g. *birthdate, employee_id, primary_email*).

Resolution Policy: Interactive verification

- System suspends processing when detecting attempt to assign second value
- System presents conflict to user with these options:
  - Retain existing value
  - Replace with new value
  - Reclassify property as multivalued (schema evolution)

Rationale: Prevents silent data corruption while allowing schema flexibility as understanding evolves.

4.3. Entity Canonicalization Conflicts

Definition: The entity resolution problem - determining whether two textual references denote the same conceptual entity.

Example: "Project Alpha", "Project Alpha (v2)", and "Alpha Initiative" may refer to distinct projects or be surface variations of a single entity.

Resolution Policy: Similarity-based matching with user confirmation. To minimize ambiguity, the system assigns a Global Unique Identifier (GUID) to each canonical node. When a lexical match is uncertain, the system leverages the Vector Store to compare the *contextual fingerprint* of the new mention against existing nodes to prioritize candidates for user confirmation.

- System detects high lexical similarity between new and existing entity mentions
- Prompts user to confirm co-reference
- If confirmed: performs node merge operation, consolidating all edges into single canonical node
- If rejected: creates new node, preserving distinction

Rationale: Prevents premature merging while avoiding entity proliferation.

4.4. Logical Contradictions

Definition: Conflict patterns violating domain-specific logical constraints.

Example: In project management:

- **Task A** → [depends_on] → **Task B**
- **Task B** → [depends_on] → **Task A**

Creates circular dependency (semantically invalid).

Resolution Policy: Constraint validation with rejection:

- System implements domain-specific constraint validation routines
- Detects contradictions before committing new tuples
- Rejects offending tuple
- Alerts user with detailed explanation of constraint violation
- Requests clarification

Rationale: Prevents knowledge graph from entering inconsistent states that compromise downstream reasoning.

4.5. Perspectival Conflicts

Definition: Instances where different users or sources provide non-factual, subjective, or contradictory qualitative assessments (e.g. **User A**: "The UI is ready"; **User B**: "The UI needs work").

Resolution Policy: Provenance-based Branching. Instead of resolving the conflict into a single 'truth,' the system creates attributed edges: **User A** → [asserts] → **UI, status, ready**.

Rationale: Preserves the nuance of collaborative environments where *truth* is a matter of consensus rather than logic.

5. Methodology

The core methodological challenge lies in transitioning from implicit, probabilistic text generation to explicit, deterministic knowledge storage. We propose a three-stage pipeline: *Detection, Extraction, and Integration*.

5.1. Ambiguity Detection and Active Elicitation Protocol

To maintain the *tabula rasa* constraint, the system must not hallucinate facts about unknown entities. The orchestrator enforces a **"retrieval-constrained"** generation policy.

Process:

1. **Entity Recognition:** Upon receiving user input, the system performs Named Entity Recognition (NER) to identify key subjects
2. **Knowledge Lookup:** System queries knowledge graph for identified entities
3. **Uncertainty Trigger:**
   - *Condition A (Known):* If entity exists with relevant edges, context is retrieved
   - *Condition B (Unknown):* If entity is absent (query returns null), system halts answer generation and enters **Elicitation State**
4. **Active Querying:** Instead of guessing, model generates clarifying question: *"I do not have a record of 'Project Alpha' in my knowledge base. Could you define its purpose and current status?"*

5.2. Fact Extraction (Tuple Schema)

When users provide information (spontaneously or in response to elicitation), the orchestrator invokes the LLM in **Extractor Mode**.

Goal: Map natural language to normalized tuple structure:

$$\text{Fact} = (\text{Subject}, \text{Relation}, \text{Object}, \text{Metadata})$$

Where **Metadata** may include:

- **Confidence level:** Value [0,1] representing degrees of certainty
- **Modality:** factual, hypothetical, uncertain
- **Source:** perspective-aware retrieval for conflict resolution

Extraction Challenges:

- **Co-reference Resolution:** Resolving pronouns (e.g., "It is broken" → `Server X` - [status] → Broken)
- **Canonicalization:** Mapping synonyms to single node ID (e.g., "The App", "Application", "App V1" → ID: Application_V1)
- **Atomic Decomposition:** Breaking complex sentences into multiple tuples
  - *Input:* "Leszek manages the delayed migration project."
  - *Extracted:*
    a. `Leszek` - [manages] → Migration Project
    b. `Migration Project` - [status] → Delayed

5.3. Knowledge Integration

New facts are integrated into the knowledge graph via a **Graph Update-or-Insert (UPSERT)** strategy ensuring data quality and consistency.

Process:

- **Deduplication:** If tuple (S, P, O) already exists, timestamp property is updated; no new node or edge is created
- **Conflict Detection & Resolution:** If new tuple violates existing constraints, system enters Conflict Resolution pipeline (Section 4)
  - **Temporal updates** applied automatically with logging
  - **Cardinality or canonicalization conflicts** suspend process for human-in-the-loop validation
  - **Logical contradictions** rejected with user clarification prompt

6. Expected Research Outcomes and Validation Strategy

6.1. Primary Research Outcomes

This research is expected to yield:

1. **Theoretical Framework:** Formal specification of the hierarchical conflict taxonomy with mathematical definitions for each conflict type and provable properties regarding consistency preservation
2. **Architectural Specification:** Complete technical blueprint for the cognitive architecture, including API contracts, data flow specifications, and system invariants
3. **Evaluation Methodology:** Comprehensive metrics framework for assessing knowledge acquisition quality across multiple dimensions:
   - Tuple extraction precision and recall
   - Entity canonicalization accuracy
   - Contradiction detection rate
   - End-to-end reasoning correctness
4. **Proof-of-Concept Implementation:** Working prototype demonstrating feasibility in a pilot domain

6.2. Validation Approach

6.2.1. Pilot Domain Selection

The system will be validated within **Collaborative Project Management** as a pilot domain for several reasons:

- **Clear Ground Truth:** Tasks are either finished or not, enabling objective evaluation
- **Dynamic State:** Projects involve frequent status changes, deadline updates, and reassignments
- **Rich Relationships:** Complex dependency structures and team hierarchies
- **Practical Relevance:** Real-world applicability in organizational settings

6.2.2. Evaluation Metrics

Tuple Extraction Quality:

$$\text{Precision} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}|}, \quad \text{Recall} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Entity Canonicalization Accuracy:

$$\text{CanonicalAccuracy} = \frac{|C|}{|T|}$$

where C = correctly resolved entity mentions, T = total entity mentions

Elicitation Efficiency:

$$\text{ElicitationEfficiency} = \frac{|Q|}{|A|}$$

where Q = clarifying questions asked, A = ambiguity events

Contradiction Detection Rate:

$$\text{ContradictionDetectionRate} = \frac{|D|}{|I|}$$

where D = contradictions correctly detected, I = injected contradictions

End-to-End Correctness:

$$\text{End-to-End Correctness} = \frac{|R|}{|N|}$$

where R = correctly answered multi-hop questions, N = total questions

Interaction Burden (IB):
This measures the trade-off between knowledge accuracy and user friction, ensuring the system remains usable in professional environments.

$$IB = \frac{T_{elicitation}}{T_{total}}$$

where $T_{elicitation}$ represents the number of turns dedicated to clarification or conflict resolution, and T_{total} is the total number of conversational turns.

6.2.3. Test Suite Design

A synthetic test suite comprising:

- 50 realistic project-management dialogues (~2,500 user turns)
- Manually created gold-standard knowledge graphs for each dialogue
- Deliberately injected conflicts across all four categories
- Complex multi-hop reasoning queries requiring graph traversal

6.2.4. Ablation Studies

Comparative analysis of:

- Vector-only retrieval
- Graph-only retrieval
- Full hybrid retrieval
- Different conflict resolution strategies

7. Future Research Directions

Beyond the initial research demonstrating core cognitive cycle and conflict resolution framework, several avenues extend the system's capabilities.

7.1. Temporal and Event-Based Reasoning

Evolution of the **Structural Core** into an **Event Knowledge Graph (EKG)** to model state transitions and causality:

- **State Machines:** Model entity status as formal state machines enforcing valid transitions

- **Temporal Querying:** Implement pathfinding algorithms for time-sensitive queries: "What was the status of Project Alpha before the budget change?"

7.2. Automated Schema Refinement and Normalization

While dynamic ontology supports initial *tabula rasa* learning, long-term efficiency requires normalization:

- **LLM-as-Ontology-Engineer:** Periodic offline process where LLM analyzes graph structure and suggests normalization steps
- **Ablation Study:** Compare lightweight vs. larger embedding models specifically on canonicalization conflict resolution

7.3. Advanced Active Learning Strategies

Transform from passive (triggered by unknown entities) to proactive, uncertainty-driven mechanism:

- **Prediction-Based Querying:** Integrate Graph Neural Network (GNN) for link prediction on knowledge graph; proactively generate questions to elicit high-confidence missing links
- **Confidence Scoring:** Assign confidence scores to facts based on source type and age; low-confidence facts become targets for active clarification

8. Limitations and Ethical Considerations

8.1. Technical Limitations

Frozen Model Constraints

- No linguistic adaptation to domain-specific jargon
- Pre-training biases persist and cannot be corrected through dialogue

Scalability Constraints

- Graph traversal complexity: O(n^d) for multi-hop queries
- Single-threaded LLM inference limits concurrent users

Context Window Limitations

- Finite number of facts injectable into LLM context
- Potential for missing relevant facts during retrieval

Single-User Assumption

- Current design assumes single authoritative user
- Multi-user scenarios require conflict resolution between human perspectives

8.2. Methodological Limitations

- Synthetic test suite may not capture all real-world dialogue patterns
- Single-domain validation limits generalizability claims
- No longitudinal evaluation of knowledge accumulation effects

8.3. Ethical Considerations

Transparency and Explainability

Strength: Full knowledge externalization enables inspection of facts used for inference

Limitation: LLM reasoning process combining facts remains opaque (neural black box)

Bias and Fairness

System inherits biases from:

1. Pre-training corpus of frozen LLM
2. User-provided information during teaching

Mitigation: Bias detection in extractor module; provenance metadata for audit; periodic fairness audits

Privacy and Data Sensitivity

- No encryption or access control in proposed architecture
- Production deployment requires: encryption at rest/transit, anonymization, role-based access control, GDPR/CCPA compliance

Accountability and Error Propagation

Risk: Misextracted facts can propagate to downstream inferences

Framework:

- User review of extracted facts before commitment
- Low-confidence extraction flagging
- Complete audit trail with provenance

Misuse

Potential for surveillance, manipulation, or social engineering

Safeguards:

- Usage guidelines prohibiting unethical applications
- Logging of all knowledge modifications
- Organizational IRB approval for human subject data

9. Conclusion

9.1. Research Significance

This research proposal addresses fundamental limitations in current LLM deployments: the inability to persistently learn from interactions and the opacity of their reasoning processes. By developing a cognitive architecture that fully decouples reasoning from memory, we enable AI systems that are simultaneously powerful and comprehensible.

9.2. Core Innovations

The proposed research delivers three key innovations:

1. **Hierarchical conflict resolution framework** systematically categorizing and resolving contradictions in dialogue-driven knowledge acquisition
2. **Cognitive decoupling architecture** enabling model-agnostic operation with full explainability
3. **Active elicitation protocol** preventing hallucination through retrieval-constrained generation

9.3. Implications for Explainable AI

This work advances Explainable AI by shifting from *implicit learning* to *explicit teaching*. The architecture offers:

- **Auditability:** Every decision traceable to specific facts from specific conversations
- **Correctness:** Errors directly editable without retraining
- **Transferability:** Knowledge graph exportable to different LLM backends
- **Trust Calibration:** System reliability assessable through knowledge base inspection

9.4. Broader Impact

This research contributes to a movement toward more transparent, controllable, and human-centered AI systems. As LLMs are increasingly deployed in high-stakes domains—healthcare, finance, legal services, education—the ability to understand, verify, and correct their knowledge becomes essential.

By demonstrating that sophisticated knowledge acquisition and reasoning can be achieved through explicit symbolic structures rather than solely through opaque neural parameters, this research offers a path toward AI systems that are simultaneously powerful and trustworthy—systems that augment human intelligence rather than replace human judgment.

9.5. Expected Contributions

Upon completion, this research will have demonstrated:

- That a frozen LLM can build semantically coherent knowledge graphs purely from dialogue
- A comprehensive taxonomy of conflicts arising in conversational knowledge acquisition
- That explicit knowledge externalization significantly enhances explainability and trustworthiness

The resulting framework, methodology, and findings will be released as open-source resources to enable reproduction, validation, and extension by the research community.

References

- Anthropic. (2024). Introducing Claude Sonnet 4.5: Enhanced planning and RAG. https://www.anthropic.com/news/claude-sonnet-4-5
- Edge, D., et al. (2024). From Local to Global: A Graph RAG Approach to Query-Focused Summarization. Microsoft Research. arXiv preprint.
- He, Y., et al. (2024). Knowledge Graph Enhanced Large Language Models: A Survey. arXiv:2409.12551
- Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401
- Li, B., et al. (2024). EvolvingKG: Dynamic Knowledge Graph Construction via Iterative Prompting. EMNLP 2024. arXiv:2406.08287
- Liu, N., et al. (2024). HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. arXiv:2405.14831
- Packer, M., et al. (2023). MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560
- Pan, S., et al. (2024). Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE Transactions on Knowledge and Data Engineering. arXiv:2306.08302
- Park, J. S., et al. (2023). Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442
- Press, O., et al. (2023). Let's Verify Step by Step with Active Retrieval. Google DeepMind. arXiv:2305.11174
- Xu, H., et al. (2024). LLM-Powered Text-Attributed Graph Anomaly Detection via Retrieval-Augmented Reasoning. arXiv:2511.17584

This research proposal represents a contribution to the advancement of explainable, human-centered artificial intelligence systems.