Tree-of-Thought Driven Perception-Aware Vector-Semantic Knowledge Graphs

Rahul Krishna Gaddam

1 Introduction

In banking scenarios, unstructured data is pervasive. We propose a framework to generate, store, and retrieve Knowledge Graphs (KGs) using Large Language Models (LLMs), guided by Tree-of-Thought (ToT) reasoning. Unlike classical KG pipelines, our system (i) supports *dynamic schema induction*, (ii) maintains *multi-instance representations* of the same entity, and (iii) treats graphs as *vectors in a semantic subspace*, enabling robust retrieval even under temporal, semantic, or structural drift.

2 Data Ingestion and Chunking

Raw documents are pre-processed (OCR, NER, cleaning) and split into chunks:

$$D = \{D_1, D_2, \ldots, D_N\},$$

where each chunk D_i has metadata $\{\text{chunk\_id}, \text{entity\_id}, \text{source}, \text{page}, \ldots\}$.

Optionally, each chunk is embedded via a lightweight model:

$$c_i = \text{avgEmbed}(\text{tokens}(D_i)) \in \mathbb{R}^s,$$

used for seeding ToT exploration.

3 Tree-of-Thought Orchestration

The LLM $\mathcal{L}$ generates KG fragments stepwise under ToT reasoning. A branch τ is defined as a sequence of reasoning steps:

$$\tau = (s_1, s_2, \ldots, s_{L_\tau}), \qquad G^{(\tau)} = \mathcal{L}(D \mid s_1, \ldots, s_{L_\tau}).$$

Each step adds partial nodes/edges with associated confidence. The likelihood of branch τ is

$$\log \ell(\tau) = \sum_{j=1}^{L_\tau} \log p_{\mathcal{L}}(s_j \mid s_{<j}, D).$$

4 Schema Discovery and Canonicalization

The LLM proposes schema labels s. Each label is embedded:

$$\psi(s) = \text{Embed}(\text{label} \,||\, \text{desc} \,||\, \text{examples}) \in \mathbb{R}^s.$$

Canonicalization merges s_a, s_b if

$$\cos(\psi(s_a), \psi(s_b)) > \tau_{\text{schema}}.$$

5 Graph Construction

For each branch $G^{(\tau)}$:

$$x_G = \sum_{s \in \Sigma_G} w_{G,s} \psi(s) \in \mathbb{R}^s, \tag{1}$$

$$A_G \in \mathbb{R}^{s \times s}, \quad [A_G]_{ij} = \text{relation weight}. \tag{2}$$

The raw representation is

$$\Phi_{\text{raw}}(G) = \big[x_G; \text{vec}(A_G)\big] \in \mathbb{R}^D.$$

6 Graph Embedding

Embeddings are produced via an encoder:

$$e_G = f_{\text{enc}}(\Phi_{\text{raw}}(G)) \in \mathbb{R}^d.$$

Examples include:

- Linear: $e_G = W\Phi_{\text{raw}}(G) + b$,
- Graph2Vec/GNN: $e_G = \text{POOL}(\{h_v\})$ where $h_v = \text{GNN}(\text{features}, A_G)$.

Normalize: $\bar{e}_G = e_G / \|e_G\|$.

7 Semantic Projection Subspace

To ensure identity invariance, we learn a semantic subspace $S \subset \mathbb{R}^d$ with projector P_S:

$$P_S^2 = P_S, \quad P_S^\top = P_S.$$

The projected and residual components are

$$e_G^S = P_S e_G, \qquad e_G^\perp = (I - P_S) e_G.$$

7.1 Learning P_S

- **PCA:** compute top-k eigenvectors U from covariance Σ; $P_S = UU^\top$.
- **Contrastive:** minimize

$$\mathcal{L}_{\text{NCE}} = -\sum_q \log \frac{\exp(\text{sim}(P_S e_q, P_S e_{q^+})/\tau)}{\sum_c \exp(\text{sim}(P_S e_q, P_S e_c)/\tau)}.$$

8 Multi-Instance Representation

For entity u, multiple fragments yield embeddings $\{e_{\tau_i}\}$. Canonical identity embedding:

$$e_u^{\text{canon}} = \sum_i w_i e_{\tau_i}^S, \quad w_i \propto \exp(\gamma \log \ell(\tau_i)).$$

Residuals $e_{\tau_i}^\perp$ are stored separately to preserve perception/chunk-specific drift.

9 Retrieval and Q/A

A query q is embedded: $e_q = f_{\text{enc}}(q)$. Projected and residual:

$$e_q^S = P_S e_q, \quad e_q^\perp = (I - P_S)e_q.$$

Candidate fragment f is scored:

$$\text{score}(q, f) = \alpha \cdot \frac{(e_q^S)^\top e_f^S}{\|e_q^S\|\|e_f^S\|} - \beta \cdot \|e_q^\perp - e_f^\perp\|.$$

Top-k fragments are retrieved from a vector index and assembled via Neo4j queries. The LLM then synthesizes a natural language answer.

10 Drift Analysis

- **Temporal drift:** $\Delta e_t = e_t - e_{t-1}$.
- **Projected drift:** $\|P_S e_t - P_S e_{t-1}\|$.
- **Subspace stability:** Davis–Kahan angle $\theta = \|\sin\Theta\|_2$ for old vs. new U.
- **Node distribution flow (EMD):**

$$\text{EMD}(p_{G_1}, p_{G_2}) = \min_{T \geq 0} \sum_{ij} T_{ij} c_{ij}, \quad \sum_j T_{ij} = p_{G_1}[i], \sum_i T_{ij} = p_{G_2}[j].$$

11 Evaluation Metrics

$$\text{PIR} = \frac{\#\{(i,j) : \text{same entity}, \|P_S e_i - P_S e_j\| < \epsilon\}}{\#\{\text{same-entity pairs}\}}, \tag{3}$$

$$\text{Top-K Accuracy} = \frac{\#\{\text{correct retrievals in top-K}\}}{\#\{\text{queries}\}}. \tag{4}$$

12 Conclusion

This unified framework operationalizes LLM-driven ToT KG generation, schema induction, vectorized embedding, semantic projection, drift-aware multi-instance storage, and retrieval. It bridges symbolic graph structures with continuous vector semantics, enabling scalable and semantically consistent reasoning for unstructured banking data.