

Module F: Deep Knowledge Graphs System (Base and Personal)
Part of the Eidos Unified Framework for Persistent, Dynamic, and Adaptive Multimodal
Intelligence

Contents

1 Abstract	2
2 Introduction and Motivation	2
3 Preliminaries and Notation	2
4 Formal Definitions and Mathematical Formulation	3
5 Algorithmic Description	4
6 Theoretical Analysis and Guarantees	5
7 Integration with the Overall Eidos Framework	5
8 Implementation Considerations	5
9 Conclusion	6

1 Abstract

This module rigorously defines the *Deep Knowledge Graphs System* of the Eidos framework, which integrates a dual-layer knowledge representation. The first layer, the *Base Knowledge Graph (BKG)*, is constructed from the static, foundational vocabulary and tokenization system and encodes intrinsic semantic, syntactic, and relational information. The second layer, the *Personal Knowledge Graph (PKG)*, dynamically augments the base graph by incorporating adaptive, real-time updates based on model interactions, user feedback, and domain-specific signals. A fusion operator integrates these layers into a unified knowledge graph, enabling robust, scalable, and continuously adaptive knowledge representations for downstream processing. This document details the formal definitions, algorithmic constructions, theoretical guarantees, and integration strategies for this system.

2 Introduction and Motivation

In modern natural language and multimodal processing systems, representing the relationships between tokens is crucial for high-level reasoning and understanding. The *Deep Knowledge Graphs System* provides such a representation by constructing two distinct yet interrelated graphs:

- (a) The **Base Knowledge Graph (BKG)** captures fundamental relationships (e.g., semantic, syntactic, and lexical associations) derived from a static, pre-defined vocabulary.
- (b) The **Personal Knowledge Graph (PKG)** captures dynamic, adaptive relationships that evolve in real time based on contextual and usage-specific signals.

These two layers are integrated via a *fusion operator* to form a unified knowledge graph that supports downstream tasks such as contextual embedding refinement, inference in deep architectures, and continual learning. The dual-layer design ensures both stability (through the base graph) and adaptability (through the personal graph).

3 Preliminaries and Notation

We assume the existence of the complete vocabulary \mathcal{V} as defined in Module D. In particular:

- Each token $t \in \mathcal{V}$ is represented as

$$t = (u, \pi, \chi),$$

where u is the underlying unit (a string or symbol), $\pi \in \Pi \subseteq \mathbb{R}^{d_\pi}$ encodes intrinsic properties, and $\chi \in \mathbb{R}^{d_\chi}$ contains contextual statistics.

- A unique identifier mapping is provided by

$$\eta : \mathcal{V} \rightarrow \mathbb{N},$$

so that $\text{ID}(t) = \eta(t)$.

For the knowledge graphs, we introduce the following additional notation:

- $\mathcal{G}_{\text{BKG}} = (\mathcal{N}_{\text{BKG}}, \mathcal{E}_{\text{BKG}})$: the Base Knowledge Graph.
- $\mathcal{G}_{\text{PKG}} = (\mathcal{N}_{\text{PKG}}, \mathcal{E}_{\text{PKG}})$: the Personal Knowledge Graph.

- $\mathcal{G}_{\text{Unified}}$: the unified knowledge graph resulting from the fusion of the BKG and PKG.
- $\rho_{\text{base}} : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{P}(\mathcal{R}_{\text{base}})$ is a relation function for the base graph, where $\mathcal{R}_{\text{base}}$ denotes the set of relation types (e.g., synonymy, hypernymy, syntactic dependency).
- $\rho_{\text{personal}} : \mathcal{V} \times \mathcal{V} \times \Xi \rightarrow \mathcal{P}(\mathcal{R}_{\text{personal}})$ is a relation function for the personal graph, with Ξ representing adaptive, contextual parameters.

4 Formal Definitions and Mathematical Formulation

Definition F.1 (Base Knowledge Graph)

The *Base Knowledge Graph* is defined as:

$$\mathcal{G}_{\text{BKG}} = (\mathcal{N}_{\text{BKG}}, \mathcal{E}_{\text{BKG}}),$$

where:

- $\mathcal{N}_{\text{BKG}} = \{n_t \mid t \in \mathcal{V}\}$ is the set of nodes, with each node n_t corresponding to a token t and associated with its base embedding $E_B(t) \in \mathbb{R}^{d_E}$.
- \mathcal{E}_{BKG} is the set of edges defined by a relation function:

$$\rho_{\text{base}} : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{P}(\mathcal{R}_{\text{base}}),$$

such that an edge exists between nodes n_{t_i} and n_{t_j} if there exists a relation $r \in \rho_{\text{base}}(t_i, t_j)$. Each edge may be represented as:

$$e_{ij} = (n_{t_i}, n_{t_j}, r).$$

Definition F.2 (Personal Knowledge Graph)

The *Personal Knowledge Graph* is defined as:

$$\mathcal{G}_{\text{PKG}} = (\mathcal{N}_{\text{PKG}}, \mathcal{E}_{\text{PKG}}),$$

where:

- $\mathcal{N}_{\text{PKG}} = \{n'_t \mid t \in \mathcal{V}\}$ is the set of nodes, each corresponding to a token t but associated with a personalized (adaptive) embedding $E_{\text{sup}}(t, \xi) \in \mathbb{R}^{d_C}$, where ξ represents contextual or domain-specific parameters.
- \mathcal{E}_{PKG} is defined by a personalized relation function:

$$\rho_{\text{personal}} : \mathcal{V} \times \mathcal{V} \times \Xi \rightarrow \mathcal{P}(\mathcal{R}_{\text{personal}}),$$

such that an edge exists between nodes n'_{t_i} and n'_{t_j} if there is a personalized relation $r' \in \rho_{\text{personal}}(t_i, t_j, \xi)$. Each edge is represented as:

$$e'_{ij} = (n'_{t_i}, n'_{t_j}, r').$$

Definition F.3 (Fusion Operator and Unified Knowledge Graph)

Define a fusion operator:

$$\oplus_{\mathcal{K}} : \mathcal{G}_{\text{BKG}} \times \mathcal{G}_{\text{PKG}} \rightarrow \mathcal{G}_{\text{Unified}},$$

which constructs the *Unified Knowledge Graph*:

$$\mathcal{G}_{\text{Unified}} = \mathcal{G}_{\text{BKG}} \cup \mathcal{G}_{\text{PKG}},$$

with:

- $\mathcal{N}_{\text{Unified}} = \mathcal{N}_{\text{BKG}} = \mathcal{N}_{\text{PKG}}$, by aligning nodes via the unique token identifiers.
- $\mathcal{E}_{\text{Unified}} = \mathcal{E}_{\text{BKG}} \cup \mathcal{E}_{\text{PKG}}$, with each edge annotated by its source (base or personal), and possibly weighted by confidence scores.

This operator guarantees that the unified graph retains stable base knowledge while integrating adaptive personal updates.

5 Algorithmic Description

We now describe the process for constructing and updating the knowledge graphs.

Algorithm 1 Construction of Base and Personal Knowledge Graphs

- 1: **Input:** Token sequence (t_1, t_2, \dots, t_n) from $\mathcal{T}_{\text{base}}$; Base embedding function E_B ; Adaptive embedding function $E_{\text{sup}}(\cdot, \xi)$; Relation functions ρ_{base} and ρ_{personal}
 - 2: **Output:** Unified Knowledge Graph $\mathcal{G}_{\text{Unified}}$
 - 3: **Begin:**
 - 4: **for** each token t in (t_1, \dots, t_n) **do**
 - 5: Create base node n_t with embedding $E_B(t)$
 - 6: Create personal node n'_t with embedding $E_{\text{sup}}(t, \xi)$
 - 7: **end for**
 - 8: **for** each pair of tokens (t_i, t_j) **do**
 - 9: Determine base relations: $R_{\text{base}} \leftarrow \rho_{\text{base}}(t_i, t_j)$
 - 10: For each $r \in R_{\text{base}}$, add edge (n_{t_i}, n_{t_j}, r) to \mathcal{E}_{BKG}
 - 11: Determine personal relations: $R_{\text{personal}} \leftarrow \rho_{\text{personal}}(t_i, t_j, \xi)$
 - 12: For each $r' \in R_{\text{personal}}$, add edge (n'_{t_i}, n'_{t_j}, r') to \mathcal{E}_{PKG}
 - 13: **end for**
 - 14: **Fusion:**

$$\mathcal{G}_{\text{Unified}} \leftarrow \oplus_{\mathcal{K}} (\mathcal{G}_{\text{BKG}}, \mathcal{G}_{\text{PKG}})$$
 - 15: **Return:** $\mathcal{G}_{\text{Unified}}$
-

Remark: In practice, the relation functions ρ_{base} and ρ_{personal} may be implemented via statistical analyses (e.g., co-occurrence frequencies, syntactic dependency parsing) and can be further refined using supervised or unsupervised methods.

6 Theoretical Analysis and Guarantees

Theorem F.1 (Consistency of Node Identities)

Statement: Given that the base and personal graphs are constructed from the same vocabulary \mathcal{V} and use the unique identifier mapping η , the node sets satisfy:

$$\mathcal{N}_{\text{BKG}} = \mathcal{N}_{\text{PKG}},$$

up to a canonical isomorphism. Consequently, the fusion operator $\oplus_{\mathcal{K}}$ produces a well-defined unified node set.

Proof Sketch: Since each token $t \in \mathcal{V}$ is assigned a unique identifier via η , nodes created in both graphs are aligned by this identifier. Thus, merging the two graphs yields a common node for each token. \square

Proposition F.2 (Extensibility and Modularity)

The dual-layer design allows independent updating of the Base Knowledge Graph (e.g., through periodic retraining) and the Personal Knowledge Graph (via real-time adaptation). The fusion operator is defined so that changes in one layer can be integrated without altering the other, ensuring modularity and extensibility.

7 Integration with the Overall Eidos Framework

Module F, the Deep Knowledge Graphs System, serves as the core for representing inter-token relationships. It directly interfaces with:

- **Module D (Vocabulary):** The nodes of both the base and personal graphs are derived from tokens in \mathcal{V} .
- **Module E (Contextual Embeddings):** The adaptive embeddings $E_{\text{sup}}(t, \xi)$ provide dynamic features that enrich the personal graph.
- **Subsequent Modules:** The unified knowledge graph $\mathcal{G}_{\text{Unified}}$ is used to inform downstream processes such as reasoning in deep models and further contextual adaptation.

8 Implementation Considerations

- **Data Structures:** Graphs \mathcal{G}_{BKG} and \mathcal{G}_{PKG} may be stored using graph databases or optimized sparse matrix representations. Edge metadata (e.g., relation types, confidence scores, timestamps) should be maintained.
- **Relation Extraction:** Techniques such as co-occurrence analysis, dependency parsing, and semantic similarity measures can be used to define ρ_{base} and ρ_{personal} .
- **Fusion Strategy:** The fusion operator $\oplus_{\mathcal{K}}$ should be designed to allow weighted integration of base and personal edges, possibly using confidence scores or temporal weights.
- **Dynamic Updates:** The PKG should support real-time updates, while the BKG remains relatively stable. Efficient incremental graph update algorithms will be necessary.
- **Scalability:** Given that the vocabulary size may reach millions, efficient indexing and retrieval mechanisms for nodes and edges are crucial.

9 Conclusion

In this module, we have presented a comprehensive framework for constructing deep knowledge graphs in the Eidos system. The Base Knowledge Graph captures the static, inherent relationships among tokens derived from the multidimensional vocabulary, while the Personal Knowledge Graph continuously adapts to reflect dynamic, context-sensitive information. A fusion operator integrates these two layers into a unified graph, ensuring that the system benefits from both stability and adaptivity. The module includes rigorous definitions, an algorithmic process for graph construction, theoretical guarantees regarding consistency and extensibility, and practical considerations for implementation. This dual-layer knowledge representation forms a critical foundation for advanced semantic reasoning and dynamic adaptation in the overall Eidos framework.

Module Summary:

Completed:

- Module A: Input Processing.
- Module B: Universal Communication & Data Handling Interface and Coordination.
- Module C: Universal Streaming/Handling/Loading/Indexing Module.
- Module D: Multidimensional Vocabulary and Tokenization System.
- Module E: Contextual NLU/NLP Embedding and Multidimensional Tokenization.
- Module F: Deep Knowledge Graphs System (Base and Personal).

Remaining Modules:

- Module G: Infinite RoPE Context Scaling and Dynamic Vocabulary Updating.
- Module H: Core Model Architectures (RWKV and Transformer Modules, Mixture-of-Experts Style).
- Module I: Titans Memory Architecture (Multi-Layer Memory Module).
- Module J: Recursive Adaptive Dynamic Idempotent Feedback and State-Based Runtime Learning and Inference.
- Module K: Universal Training System.
- Module L: Final Decoding and Multimodal Output.