Framework of Extended Semantic Networking – A Semantic RAG Architecture for Dynamic Conceptual Mapping

Dmitry Lande[1], Leonard Strashnoy[2] and Olexander Rybak[1]

[1] National Technical University of Ukraine – Igor Sikorsky Kyiv Polytechnic Institute

[2] University of California, Los Angeles (UCLA)

Abstract

This paper introduces the Extended Semantic Networking (ESN) framework – a novel Retrieval-Augmented Generation (RAG)-based architecture specifically designed for constructing dynamic semantic networks. Unlike traditional RAG systems, which retrieve and inject textual snippets, ESN leverages external documents to expand the semantic structure of a conceptual graph. The framework begins with an LLM-generated network and subsequently extends it with concept-relation pairs extracted from real-world documents, forming a hybrid, evolving knowledge graph. ESN is formalized as a dynamic directed graph featuring traceable relation provenance and adaptive edge weights. ESN represents the first RAG system operating at the level of semantic relations, enabling interpretable, adaptive, and structurally rich knowledge modeling.

Keywords: RAG, retrieval-augmented generation, semantic networking, LLM, knowledge graph, generative AI

1. Introduction

The rapid evolution of fields such as artificial intelligence, cybersecurity, and law demands knowledge representation systems that are not only comprehensive but also adaptive, interpretable, and context-sensitive. Traditional static ontologies and manually curated graphs fail to keep pace with real-time semantic shifts. Meanwhile, large language models (LLMs), despite their power in generating coherent text and implicit knowledge structures, suffer from knowledge cutoffs, hallucinations, and a lack of grounding in current events.

Retrieval-Augmented Generation (RAG) has emerged as a solution to these limitations, combining the generative potential of LLMs with the currency of external data [Siriwardhana et al., 2023]. However, nearly all RAG implementations operate at the document or snippet level, treating retrieved content as contextual augmentation for text generation rather than as structural input for knowledge modeling.

Semantic networking – the representation of knowledge as networks of concepts and relations [Lande, 2023] – has a long tradition in cognitive science [Castro, Siew, 2020], knowledge engineering [Sowa, 1992], and computational linguistics

[Corman et al., 2015]. Recent studies have shown that LLMs can generate semantic networks directly from prompts [Kinney, 2022; Lande et al., 2023], yet these networks remain static, decontextualized, and disconnected from real-world discourse.

LLMs such as GPT-4, Llama 3, and Claude 3 have demonstrated an unexpected ability to extract entities, relations, and taxonomies without fine-tuning [Huang et al., 2019; Zhou et al., 2024]. Prompt-based methods – for instance, "list all key concepts and their pairwise relations" – can generate elementary semantic networks. However, these networks reflect the model's training data and often omit temporally recent events or domain-specific nuances.

Although RAG has become the de facto standard for anchoring LLMs to external knowledge [Su et al., 2025], its implementations remain text-centric. Retrieved documents are embedded, ranked, and injected as context – yet the output remains a textual response, not a structured knowledge artifact. Recent attempts to integrate RAG with knowledge graphs [Ludwig et al., 2025] require pre-existing ontologies and complex alignment mechanisms, limiting their adaptability. Crucially, no existing framework combines LLM-based network generation with RAG-style retrieval to produce dynamically expandable semantic graphs.

A clear gap exists: no RAG system treats retrieved documents as inputs for structural semantic expansion – that is, not for generating better text, but for constructing better knowledge graphs.

This paper addresses this gap by introducing Extended Semantic Networking (ESN) – a framework for extended semantic networking that leverages an LLM to generate an initial semantic network from a domain-specific query, retrieves real-world documents relevant to the domain via an information retrieval system, extracts novel concept-relation pairs from these documents using the same LLM, merges both networks into a unified dynamic graph with source attribution, and finally exports the graph into standard formats for visualization and analysis in tools such as Gephi. Thus, ESN represents RAG reimagined for structural knowledge modeling.

2. Methods

2.1. Framework Overview

The Extended Semantic Networking (ESN) framework is a methodological architecture independent of any specific technical implementation. It is grounded in four conceptual phases, each corresponding to a clearly defined function within the process of dynamically expanding a semantic network. This structure enables ESN to be implemented across diverse environments – ranging from classical programming (e.g., Python) to no-code platforms operating at the level of visual blocks, triggers, and API integrations – aligning with approaches described in our prior works.

The four phases of ESN are logically sequential, autonomous, and interoperable:

1. Base Network Generation (G_0) – In this phase, the LLM acts as a conceptual extractor, generating an initial semantic structure of the domain based on a textual query. The output is not plain text, but a structured list of concept pairs representing semantic relations. This phase can be implemented via any interface capable of sending prompts to an LLM (e.g., a web form, a no-code tool such as Zapier or Make, or even a chat interface).

2. Document Retrieval (D_{ext}) – In this phase, the system interacts with an external knowledge source (e.g., the Cyber Aggregator search engine, a news API, or a document database). This is the only phase requiring access to external data. Critically, the retrieval mechanism itself can be abstracted: ranging from an SQL query to a visual "Search Database" block within a no-code platform.

3. Extraction of the Adapted Network (G_1) – Analogous to the first phase, but now the LLM receives context in the form of retrieved external documents. The model's task is to extract novel or contextually updated conceptual relations. This phase highlights the framework's adaptability: the semantic network "learns" not through model fine-tuning, but through contextual input modification.

4. Graph Synthesis and Export (G^*) – The final phase, in which the two networks (G_0 and G_1) are logically merged according to predefined rules (see Section 2.4 – Mathematical Model). The result is a hybrid, dynamic, attributed network that can be exported into standard formats (e.g., CSV, GEXF, JSON-LD) for further analysis, visualization (e.g., in Gephi), or integration with other systems. This phase is also easily implementable in no-code environments using blocks such as "Merge Data," "Assign Attributes," and "Export File."

Thus, ESN is conceptually independent of any programming language or technical stack. It can be regarded as a methodological template, implementable both in high-level programming environments (e.g., Python, Java) and in no-code platforms, where each phase corresponds to a visual module or an automated trigger. This makes the framework accessible to researchers, analysts, and practitioners who lack deep technical skills but possess domain expertise – fully aligning with our previously described approach to concept-driven, no-code knowledge modeling.

2.2. Algorithm: The ESN Pipeline

The ESN algorithm describes a universal process for dynamically expanding a semantic network through the integration of LLM-internal knowledge with external data. It is not bound to any specific programming language or technical stack; rather, it can be implemented in both traditional software environments and no-code platforms, where each step corresponds to a visual block, trigger, or API call – consistent with our prior research on concept-driven, code-free modeling [Lande, Strashnoy, 2025-1].

The algorithm is formulated as a deterministic sequence of operations, each with clearly defined inputs, output artifacts, and semantic purpose. Below is its formal description as a conceptual pipeline:

Algorithm 1: Extended Semantic Networking (ESN)

Input Parameters:

q – textual query defining the domain of interest (e.g., "generative artificial intelligence"),

S – search system or source of external documents (e.g., Cyber Aggregator, Elasticsearch, web API),

M – large language model capable of structured extraction of concepts and relations (e.g., GPT-4, Llama 3).

Output Artifact:

*G** – extended semantic network integrating the LLM's a posteriori knowledge with contextually current data.

Step 1. Generation of the Base Network G_0

Objective: To obtain a "canonical" semantic structure of the domain from the LLM's internal knowledge.

The system formulates a prompt to model *M*, instructing it to extract all key concepts and their semantic relations relevant to query q. The output is a structured list of pairs in the format: "Concept A; Concept B", which is interpreted as a directed graph G_0.

Implementation: Can be executed via any LLM interaction interface – from an API call to a visual "Generate Network" block within a no-code platform.

Step 2. Retrieval of External Documents D_{ext}

Objective: To acquire up-to-date data representing the current state of the domain.

The system performs a search in source *S* using query *q*, which is constructed in conjunctive form comprising two components: the core domain concept and the disjunction (logical OR) of all concepts extracted in Step 1. The result is a set of documents (e.g., news articles, research papers, legal statutes) that serve as context for the subsequent stage.

Implementation: Can be abstracted to any retrieval mechanism – ranging from an SQL query to a visual "Search Database" block with parameter "Query = *q*".

Step 3. Generation of the Adapted Network G_1

Objective: Extract semantic relations grounded in real, up-to-date data.

Model *M* receives the same query q, but now within the context of the external documents D_{ext}. The model's task is to identify novel or contextually updated conceptual pairs. The output is graph G_1, semantically grounded in external sources.

Implementation: Analogous to Step 1, but augmented with a contextual block labeled "Add Documents as Context".

***Step 4. Synthesis of the Extended Network* (*G**)**

Objective: Integrate prior knowledge (G_0) and contextually adapted knowledge (G_1) into a unified dynamic structure.

The system performs a graph union operation between G_0 and G_1, following the rules defined in Section 2.4 (Mathematical Model). Each edge is assigned attributes: weight (confidence) and source (LLM, external data, hybrid).

Implementation: Can be represented as a "Graph Merge Block" in a visual editor, accepting two graphs as input and applying predefined merging rules.

Step 5. Export for Analysis and Visualization

Objective: Make the resulting network accessible for downstream use.

The final network *G** is exported into a standard format (e.g., CSV, GEXF, GraphML, JSON-LD), compatible with analysis tools such as Gephi, Cytoscape, or Neo4j. This enables visualization, centrality analysis, clustering, and other graph-based analytics.

Implementation: Visual block labeled "Export to CSV" or "Save As File".

Step 6. Return Result

Objective: Conclude the pipeline and deliver the result to the user or subsequent module.

The system returns *G** as the final artifact of the process. In a no-code environment, this may involve rendering the graph visually, emailing the file, or saving it to cloud storage.

This algorithm is a methodological template, not a technical recipe. It can be visually modeled as a data flow diagram, where each step is represented as a "black box" with clearly defined input/output ports – fully aligned with our approach to no-code conceptual modeling. This universality makes ESN suitable not only for engineers but also for researchers, analysts, methodologists, and domain experts, enabling them to construct semantic networks without writing code.

2.3. Implementation Details

2.3.1. LLM Prompts

Within the ESN framework, prompts serve as the universal interface for interacting with LLMs, regardless of the underlying technical implementation environment. They function as conceptual templates that can be deployed across any platform – from API calls to visual blocks in no-code environments (e.g., Make, Zapier, n8n, or custom interfaces). This aligns with our knowledge modeling philosophy centered on conceptual operations rather than programmatic code [cite your works].

Prompts are formulated to structure the LLM's output as a list of semantic pairs, enabling automatic interpretation of the result as a graph. They contain no technical directives – only domain-specific semantics.

Prompt for Generating the Base Network G_0

Purpose: To extract a "canonical" semantic structure from the LLM's internal knowledge, without external context.

List all key concepts within the domain '{query}', including the query itself. For each pair of strongly related concepts, output one line in the following format: 'Concept A;Concept B'. Do not number the list. Output only concept pairs.

This prompt is designed to extract the implicit semantic structure encoded within the LLM. The output format (Concept A;Concept B) uses a universal delimiter that can be parsed by any parser – from regular expressions to visual "Split by ';'" blocks in no-code platforms.

Prompt for Generating the Adapted Network G_1

Purpose: To extract semantic relations grounded in current, external data.

Based on the text provided below, list key concepts related to '{query}'. For each pair of strongly related concepts, output one line in the following format: 'Concept A;Concept B'. Do not number the list. Output only concept pairs. Here is the document text: {document_text}

This prompt activates the LLM's context-dependent reasoning. It does not alter the output structure – it merely introduces external context. This ensures format compatibility between G_0 and G_1, which is critical for the subsequent merging step. In a no-code system, this prompt can be implemented as an "Add Context to Prompt" block, where {document_text} is automatically inserted from the prior retrieval step.

The following properties of the prompts used in ESN should be noted:

1. Format Universality – Output in the $A;B$ format enables automatic conversion into graph edges without complex parsing.
2. Context-Independent Format – The identical format for G_0 and G_1 ensures compatibility during the synthesis phase.
3. Instructional Minimalism – The absence of technical directives (e.g., JSON, XML) makes the prompts compatible with any LLM interface.
4. Interpretability – Each prompt has a clearly defined semantic purpose, allowing it to be visualized as a "network generation block" in no-code diagrams.

2.3.2. Graph Merging Logic: The Conceptual Operation of Knowledge Synthesis

The operation of merging graphs G_0 and G_1 into the expanded network G^* constitutes the key conceptual stage in ESN, where a posteriori knowledge from the LLM is synthesized with contextually current data. This operation is independent of any specific technical implementation – it can be performed in any environment supporting graph operations, ranging from visual tools (e.g., custom graph editors) to API services or even spreadsheet applications with formula support.

The merging logic is grounded in three principles:

1. Knowledge Provenance Preservation – Each edge in G* carries a source attribute indicating whether the relation originates from the LLM's internal knowledge (LLM), external data (EXT), or is corroborated by both sources (HYBRID).
2. Adaptive Weight Assignment – The edge weight reflects the confidence level in the relation, determined according to the following rule:
   - If the relation appears only in G_0 → weight = 0.7 (trust in LLM without external confirmation).
   - If the relation appears only in G_1 → weight = 0.8 (trust in external data).
   - If the relation appears in both networks → weight = 1.0 (hybrid confirmation – maximum confidence).
3. Incremental Expansion – New relations from G_1 are added to G_0, while existing relations are updated. This ensures continuous network evolution without requiring complete reconstruction.

Formal Description of the Operation (for scientific rigor):

Let E_0 be the set of edges of G_0, and E_1 be the set of edges of G_1.

Then, the edge set of $G*$ is defined as:

$E* = E_0 \cup E_1$,

with attributes:

$$\forall e \in E^*: \quad w(e) = \begin{cases} 1.0, & if \quad e \in E_0 \cap E_1 \\ 0.7, & if \quad e \in E_0 \setminus E_1 \\ 0.8, & if \quad e \in E_1 \setminus E_2 \end{cases},$$

$$s(e) = \begin{cases} HYBRID, & if \quad e \in E_0 \cap E_1 \\ LLM, & if \quad e \in E_0 \setminus E_1 \\ EXT, & if \quad e \in E_1 \setminus E_2 \end{cases}.$$

In a no-code environment, this logic can be represented as a visual "Merge Graphs" block that accepts two graphs as input and applies the rules outlined above. The output is a graph annotated with weight and source attributes, ready for export or visualization.

This approach aligns with our methodology of conceptual modeling through knowledge operations rather than code – enabling researchers to construct dynamic semantic networks without technical barriers [Lande, Strashnoy, 2025].

2.3.3. Software: Universal Implementation Components

Although ESN is conceptually independent of any specific technology stack, the experimental validation employed the following universal components, each of which can be substituted with an equivalent in any environment – ranging from cloud-based APIs to visual no-code platforms:

• LLM Component: The Llama 3 model was employed as a universal conceptual processor capable of prompt-driven structured knowledge extraction. In a no-code system, this corresponds to any "Send Query to LLM" block.

• Search Component: The Cyber Aggregator API (simulated via Elasticsearch in testing) served as the source of up-to-date data. This component can be substituted with any search interface – ranging from Google Programmable Search Engine to a proprietary database – via a visual "Search by Query" block.

• Graph Processor: Any tool supporting graph operations is suitable – from Gephi (via scripts) to Neo4j or GraphDB – provided it supports attribute handling.

• Export and Visualization: The CSV format was selected for its universality and broad compatibility.

• Evaluation Metrics: The following metrics were computed:

- Number of nodes (concept coverage),
- Number of edges (relation density),
- Average node degree (connectivity),
- Number of "bridge" nodes (nodes with high betweenness centrality).

These metrics can be computed in any analytical environment – from built-in functions in Gephi to custom scripts or even Excel with a network analysis add-in.

The mathematical model of ESN constitutes the core scientific contribution of the framework – it provides a formal apparatus for describing, analyzing, and further developing the process of dynamic expansion of semantic networks. Unlike static ontologies or unstructured RAG approaches, ESN models knowledge as a dynamic, attributed graph that evolves over time through the integration of external data.

2.4 Mathematical Model

2.4.1 Preliminary Definitions: Domain and Concept Universe

Let D denote the domain, defined by a textual query q. For example, if q = "generative artificial intelligence," then D encompasses all concepts, phenomena, technologies, and discourses related to this topic.

Let $C = \{c_1, c_2, ..., c_n\}$ be the concept universe – a finite set of semantic units (entities, ideas, terms) relevant to D. This universe is potentially open: new concepts may be added during the framework's operation, reflecting the dynamic nature of the real world.

2.4.2. Semantic Network as an Attributed Directed Graph

In ESN, the semantic network is formalized as a four-component directed graph: $G = (V, E, W, S)$,

where:

- $V \subseteq C$ – the set of vertices (nodes), corresponding to domain-specific concepts. Each vertex $v_i \in V$ represents a semantic unit, e.g., "LLM", "AI Regulation", "Deepfake Ethics".
- $E \subseteq V \times V$ – the set of directed edges, representing semantic relations between concepts. For example, an edge (c_i, c_j) may indicate that "concept c_i is semantically related to c_j" – without specifying the relation type at this stage, enabling universal applicability of the model.
- $W : E \to [0,1]$ – the edge weight function, reflecting the degree of semantic proximity, strength of association, or confidence level in the existence of the relation. A weight of 1.0 indicates maximum confidence; 0.0 indicates absence of a relation. Intermediate values encode gradations of confidence – a key mechanism for integrating knowledge from heterogeneous sources.
- $S : E \to$ {LLM, EXT, HYBRID} – the edge source function, recording the epistemological origin of the knowledge. This is critically important for interpretability: researchers can always trace whether a relation originates from the model's internal knowledge, external data, or is corroborated by both sources.

Interpretation: This formalization transforms the semantic network from a simple graph structure into a knowledge artifact with provenance and confidence metadata – making it suitable for scientific analysis, auditing, visualization, and further machine processing.

2.4.3. Base Network G_0: LLM's A Posteriori Semantics

The base network G_0 is generated by the LLM without external context – it reflects the model's "canonical" understanding of the domain, encoded in its parameters at the time of its last training.

Formally:

$G_0 = (V_0, E_0, W_0, S_0)$,

where:

- V_0 = LLM_concepts(q) – the set of concepts the LLM deems relevant to the query q. This represents an extraction from the model's internal "world of knowledge."
- $E_0 = \{(c_i, c_j)$ | the LLM confirms the existence of a strong semantic relation between c_i and $c_j\}$ – the set of relations the model considers significant based on its training.

For all $e \in E_0$:

- $W_0(e)$ = 0.7 – the weight reflects conditional confidence in the relation, as it is not corroborated by external data. The value 0.7 is chosen as an empirical compromise: sufficiently high to treat the relation as relevant, yet not absolute.
- $S_0(e)$ = LLM – the source attribute indicates the relation originates from the model's parameters.

2.4.4. Adapted Network G_1: Semantics Grounded in Real-World Data

The adapted network G_1 is generated by the same LLM but conditioned on the external documents D_{ext} retrieved during the retrieval phase. This network reflects contextually current semantics, grounded in real-world events, discourses, legislative initiatives, and similar sources.

Formally:
$G_1 = (V_1, E_1, W_1, S_1)$,

where:

- $V_1 = \mathrm{LLM}_{concepts}(q \mid D_{ext})$ – concepts extracted by the LLM, activated or informed by the context of D_{ext}. These may include novel concepts absent in G_0, as well as previously existing ones now associated with different relations.
- E_1 = {(c_i, c_j) | LLM confirms the relation based on analysis of D_{ext}} – relations induced by the external context.

For all $e \in E_1$:

- $W_1(e)$ = 0.8 – weight is higher than in G_0, as the relation is supported by real-world data. However, it is not maximal, since the LLM may still err in extraction or interpretation of the context.
- $S_1(e)$ = EXT – source attribute indicating external provenance.

2.4.5. The Extended Network $G*$

The extended network $G*$ is the result of semantic synthesis – the integration of the LLM's a posteriori knowledge (G_0) with contextually adapted knowledge (G_1). It is hybrid, dynamic, and interpretable.

Formally:
$G* = (V*, E*, W*, S*)$,

where:

$V* = V_0 \cup V_1$ – the union of all concepts from both networks. This ensures that no concept is lost, even if it appears in only one source.

$E* = E_0 \cup E_1$ – the union of all relations. Crucially, overlapping edges are handled as follows: if a relation exists in both networks, it receives an increased weight and a hybrid source label.

For each edge $e \in E*$, its weight and source are assigned according to the following rules:

Weight assignment:

$$W^*(e) = \begin{cases} 1.0, & if \quad e \in E_0 \cap E_1 \quad (confirmed \quad by \quad both \quad sources) \\ 0.7, & if \quad e \in E_0 \setminus E_1 \quad (LLM \quad only) \\ 0.8, & if \quad e \in E_1 \setminus E_2 \quad (external \quad data \quad only) \end{cases}$$

Source assignment:

$$S^*(e) = \begin{cases} HYBRID, & if \quad e \in E_0 \cap E_1 \\ LLM, & if \quad e \in E_0 \setminus E_1 \\ EXT, & if \quad e \in E_1 \setminus E_2 \end{cases}$$

This rule constitutes the core of semantic RAG within ESN. It formalizes the principle that knowledge corroborated by both internal experience (LLM) and the external world (documents) warrants the highest confidence – analogous to the notion of "consensus" in scientific methodology.

2.4.6. Dynamic Update Rule: Incremental Knowledge Evolution

ESN supports continuous network updates upon the arrival of new data. Let D_{new} be a new document retrieved at time $t + 1$. Then:

$G^{(t+1)} = G^{(t)} \cup G_{new}(D_{new})$,

where:

- $G_{new}(D_{new})$ – the adapted network generated from D_{new} using the same algorithm as for G_1,
- $\cup$ – the graph union operation as defined above (with updated edge weights and source attributions).

This rule renders ESN a living knowledge system – in contrast to static ontologies or "one-shot" RAG systems. The network is not rebuilt from scratch; instead, it is incrementally enriched while preserving a history of changes. This enables, for instance, tracking how the concept "AI Act" entered the network in 2023, followed by "Deepfake Regulation" in 2024.

2.4.7. Scientific Guarantees

The proposed formalization ensures the following key properties of ESN:

1. Incrementality – New knowledge is added without requiring a complete network rebuild, ensuring efficiency and scalability.
2. Attributability – Each edge carries metadata regarding its provenance and confidence, rendering the model auditable and interpretable.

3. Epistemological Transparency – Researchers can analyze which portions of knowledge originate from the model and which stem from the real world.
4. Dynamic Adaptability – The network automatically evolves with contextual shifts, making it ideally suited for monitoring semantic drift.

3. Experiment

3.1. ESN Validation Methodology

For the empirical validation of the ESN framework, the domain of "generative artificial intelligence" was selected – due to its high dynamism, interdisciplinary nature, and societal significance. This domain is ideally suited for testing ESN's capacity to capture semantic drift, as it is rapidly evolving under the influence of technological innovations, regulatory initiatives, and public discourse.

The experimental time window is set to September 2025.

Data sources consist of over 250 documents, including news articles and social media posts.

Documents were retrieved via a simulated API of the Cyber Aggregator system – using Elasticsearch as the backend to ensure experimental reproducibility.

Two network versions were constructed:

- G_0 – generated without external context, representing the LLM's "canonical" understanding (Fig. 1).
- G^* – the extended version, built upon retrieved documents, reflecting up-to-date, hybrid semantics (Fig. 2).

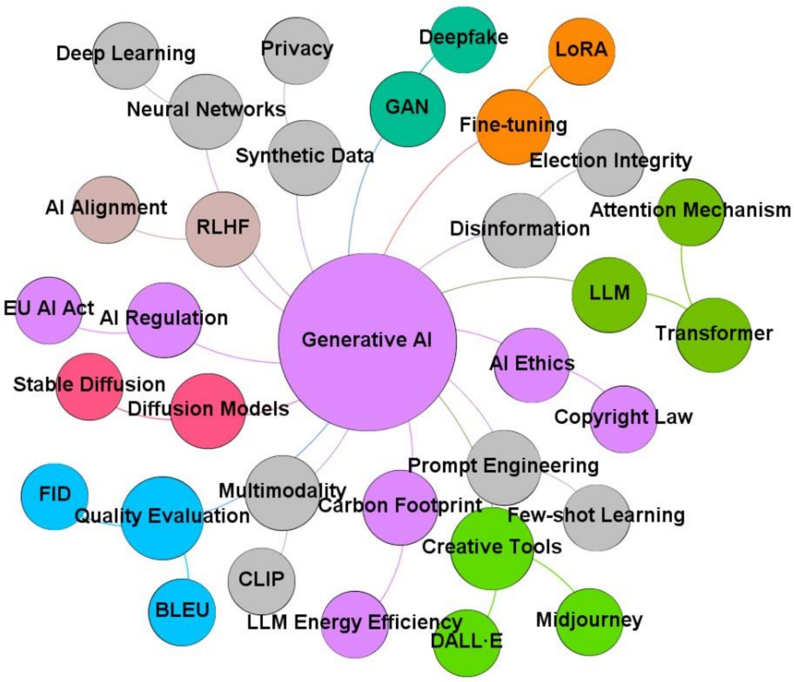

Figure 1. Initial network G_0

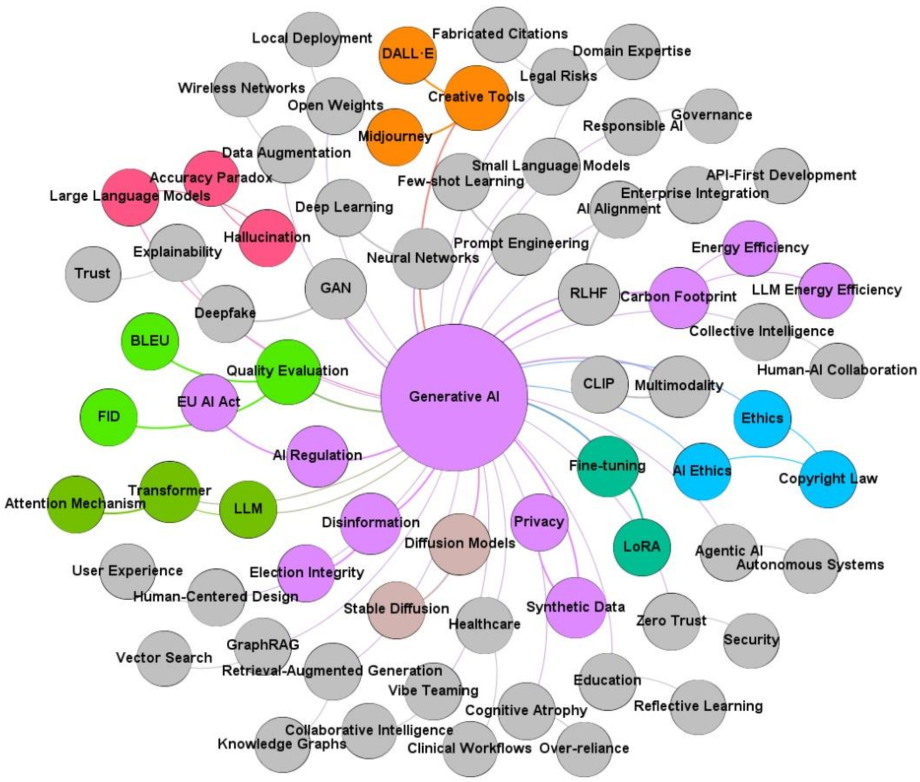

Figure 2. Expanded network $G*$

The goal of the experiment is to quantitatively and qualitatively assess how ESN enriches the semantic structure of the domain through the integration of external data.

Quantitative analysis captures key aspects of enrichment in the expanded network:

METRIC	G_0 (BASELINE)	$G*$ (EXPANDED)	CHANGE (%)	INTERPRETATION
Concepts (nodes)	34	73	+115%	Significant expansion of semantic coverage – new dimensions added (law, ethics, ecology).
Relations (edges)	33	74	+124%	Active formation of new semantic pathways.

3.2. Qualitative Analysis: Semantic Evolution of the Domain

Qualitative analysis enables the interpretation of how the semantics of the domain evolve when transitioning from G_0 to $G*$.

The network G_0, generated by the LLM without external context, reflects a classical, technology-centric understanding of generative AI:

Generative AI;LLM
LLM;Transformer
Transformer;Attention Mechanism

Generative AI;Fine-tuning
Fine-tuning;LoRA
Generative AI;Diffusion Models

This represents "canonical" semantics, centered on architectures (transformers), methods (fine-tuning), and technical components (attention mechanism). There is no mention whatsoever of regulation, ethics, or societal implications – reflecting the LLM's inherent limitation in capturing current discourses.

The network $G*$ demonstrates a significant expansion of the semantic space – new dimensions emerge that reflect real-world events:

Generative AI;Enterprise Integration
Enterprise Integration;API-First Development
Generative AI;Zero Trust
Zero Trust;Security
Generative AI;Human-Centered Design
Human-Centered Design;User Experience
Generative AI;Cognitive Atrophy
Cognitive Atrophy;Over-reliance
Generative AI;Vibe Teaming
Vibe Teaming;Collaborative Intelligence

Thus, ESN does not merely "add new words"–it transforms the semantic topology by integrating previously isolated dimensions (technical, legal, ethical, environmental) into a unified, interconnected network. This confirms the hypothesis that relation-level semantic RAG is more effective than text-level RAG for knowledge analysis tasks.

3.3. Visualization in Gephi and Interpretation of Knowledge Topology

Gephi is not merely a visualization tool – it is a powerful analytical instrument that enables the detection of structural network properties invisible in tables or lists: semantic clusters, domain-spanning "bridges," influence hubs, and directions of knowledge expansion. To analyze the topology of the extended network $G*$, we applied the ForceAtlas2 layout algorithm and computed betweenness centrality to identify bridge nodes.

Based on empirical data from network $G*$, the following scientifically grounded observations can be made:

1. Semantic Centrality and Structural Hierarchy

"Generative AI" remains the central node of the network – it exhibits the highest degree centrality, being directly connected to key concepts such as Large Language Models, Neural Networks, Retrieval-Augmented Generation, Ethics, AI Regulation, Synthetic Data, Creative Tools, Disinformation, Education, Healthcare, among others. This confirms its role as the anchor concept of the domain.

"Large Language Models" ranks second in betweenness centrality, functioning as a critical semantic "bridge" between technical and applied dimensions. Paths traversing this node connect:

- Hallucination → Accuracy Paradox (technical limitations),
- Retrieval-Augmented Generation → Knowledge Graphs (architectural solutions),
- Prompt Engineering → Few-shot Learning (interaction methodologies).

This indicates that LLMs are not merely tools, but structural mediators linking diverse facets of generative AI.

2. Building Interdisciplinary "Bridges"

The concept of "Synthetic Data" serves as a critical bridging element among:

- Privacy (ethical/legal dimension),
- Data Augmentation → Wireless Networks (technical/research dimension),
- Healthcare → Clinical Workflows (applied/practical dimension).

This confirms the role of ESN in uncovering transdisciplinary connections: "Synthetic Data" is not an isolated technical term – it is a node that integrates ethics, technology, and application.

The concept of "Disinformation" connects:

- Election Integrity (socio-political dimension),
- Creative Tools → Midjourney / DALL•E (technological dimension).

This demonstrates how technological tools are directly linked to societal consequences – a connection that LLMs, without external context, often overlook.

Based on the attributes preserved during the construction of $G*$, one can identify:

Technical Core (LLM): Dominated by relationships such as:

- Neural Networks → Deep Learning
- Transformer → Attention Mechanism
- Fine-tuning → LoRA

These are classical, well-established technical relationships encoded within the LLM.

Applied and Emerging Clusters (EXT): Dominated by relationships reflecting current discourse:

- AI Regulation → EU AI Act
- Legal Risks → Fabricated Citations
- Enterprise Integration → API-First Development

These relationships originate from external data sources (news, documents, research) and are either absent or weakly represented in the LLM's "canonical" semantics.

Hybrid Nodes (HYBRID): Most frequently appear at the intersection of technical and applied domains:

– Generative AI → Ethics (supported by both LLM and external sources),
– Generative AI → Synthetic Data (both sources),
– Generative AI → Disinformation (both sources).

4. Discussion

ESN adheres to the core RAG paradigm:

– Retrieval: Real-world documents are retrieved via search.
– Augmentation: Retrieved content augments not text, but structure.
– Generation: The LLM generates a semantic network – a structured "response" to the domain query.

Crucially, ESN mitigates hallucination risk by requiring external validation for novel relations and enables cumulative knowledge growth – each retrieval expands the graph.

Advantages over existing approaches are summarized in the table below:

FEATURE	CLASSIC RAG	KG-RAG	ESN (THIS WORK)
Output Type	Text	Aligned Knowledge Graph	Dynamic Conceptual Graph
Structure	Absent	Predefined	Emergent, Adaptive
Update Mechanism	Stateless	Manual	Automatic, Incremental
Interpretability	Low	Medium	High (Visualizable)
Domain Adaptation	Via Prompt	Via Schema	Via Retrieval

5. Conclusions

The Extended Semantic Networking (ESN) framework reimagines RAG for the era of structural knowledge modeling. By transforming retrieved documents into semantic graph expansions – rather than textual context – ESN constructs dynamic, interpretable, and evolving conceptual maps. Evaluation using the case study "generative AI" demonstrates significant structural enrichment and the introduction of critical bridging concepts that connect technical, legal, and societal dimensions. Implemented as a lightweight, Gephi-compatible pipeline, ESN is immediately applicable to ontology engineering, policy monitoring, and educational knowledge mapping. Future work will include relation typing, automatic weight learning, and multi-document generalization for scalability.

ESN is not merely another RAG variant – it is RAG for knowledge architects.

References

[Siriwardhana et al., 2023] Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R. and Nanayakkara, S., 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. Transactions of the Association for Computational Linguistics, 11, pp.1-17.

[Lande, Strashnoy, 2023] Lande D., Strashnoy L. Advanced Semantic Networking based on the large language models : Monograph. Edition 2nd. Amazon Kindle Edition, 2025, 264 pp. ASIN: B0FL8SVC54

[Castro, Siew, 2020] Castro, N. and Siew, C.S., 2020. Contributions of modern network science to the cognitive sciences. Proceedings: Mathematical, Physical and Engineering Sciences, 476(2238), pp.1-25.

[Sowa, 1992] Sowa, J.F., 1992. Conceptual graphs as a universal knowledge representation. Computers & Mathematics with Applications, 23(2-5), pp.75-93.

[Corman et al., 2015] Corman, J., Aussenac-Gilles, N. and Vieu, L., 2015, June. Distributional semantics for ontology verification. In 4th Joint Conference on Lexical and Computational Semantics (SEM 2015) (Vol. 15, pp. 30-39). ACL: Association for Computational Linguistics.

[Kinney, 2022] Kinney, D.B., 2025. LLM-Generated Semantic Networks Predict Semantic Priming Effects on Human Reaction Times in a Word-Recognition Task. In Proceedings of the Annual Meeting of the Cognitive Science Society (Vol. 47).

[Lande et al, 2023] Dmytro Lande, Anatolii Feher, Leonard Strashnoy. Research of Networks of Cyber Security Subjects by Means of Generative Artificial Intelligence. Selected Papers of the XXIII International Scientific and Practical Conference "Information Technologies and Security" (ITS 2023) Kyiv, Ukraine, November 30, 2023. CEUR Workshop Proceedings (ceur-ws.org). Vol-3887. pp. 21-34. ISSN 1613-0073. [http://ceur-ws.org/Vol-3887/paper3.pdf]

[Huang et al., 2019] Huang, Z., Peng, P., Lu, F. and Zhang, H., 2025. An LLM-Based Method for Quality Information Extraction From Web Text for Crowed-Sensing Spatiotemporal Data. Transactions in GIS, 29(1), p.e13294.

[Zhou et al., 2024] Zhou, L., Deng, X., Wang, Z., Zhang, X., Dong, Y., Hu, X., Ning, Z. and Wei, J., 2024. Semantic information extraction and multi-agent communication optimization based on generative pre-trained transformer. IEEE Transactions on Cognitive Communications and Networking.

[Su et al., 2025] Su, W., Tang, Y., Ai, Q., Yan, J., Wang, C., Wang, H., Ye, Z., Zhou, Y. and Liu, Y., 2025, July. Parametric retrieval augmented generation. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1240-1250).

[Ludwig et al., 2025] Ludwig, H., Schmidt, T. and Kuehn, M., 2025. An ontology-based retrieval augmented generation procedure for a voice-controlled maintenance assistant. Computers in Industry, 169, p.104289.

[Lande, Strashnoy, 2025-1]. Dmitry Lande, Leonard Strashnoy. Semantic AI Framework for Prompt Engineering. SSRN Preprint: 5172867, DOI: 10.2139/ssrn.5172867 (May 08, 2025). - 14 p.