Ten-Week Research Roadmap
Interpretable RAG via Graph-Theoretic Knowledge Graphs

Prepared for Research Team
October 7, 2025

Abstract

This roadmap outlines a concrete, ten-week plan (five two-week milestones) to build an *interpretable* retrieval-augmented generation (RAG) system grounded in graph theory. Two core thrusts drive the work: (i) **Dynamic Query-Specific Knowledge Graphs (QS-KGs)**, constructed on-the-fly during retrieval to synthesize evidence across documents; and (ii) **Graph-Guided Generation**, where the model's output is structured by traversing paths in the QS-KG (e.g., for *acne treatment*: Mechanism of Action → Efficacy Data → Potential Side Effects). The roadmap fixes corpora and tasks, defines precise metrics and ablations, prescribes step-by-step implementation guidelines, and enumerates clear deliverables, risks, and reproducibility steps suitable for rapid execution by a single researcher, targeting a workshop-quality manuscript by Week 10.

Contents

1 Scope & Research Questions

Goal. Build a RAG system that (a) constructs a *dynamic, query-specific* knowledge graph at retrieval time; (b) exposes *interpretable evidence chains* via graph paths; and (c) *constrains* generation to follow graph-guided structures for completeness and faithfulness.

Core questions.

1. How accurately and efficiently can we induce a QS-KG from top-k retrieved passages for a single query?
2. Do graph-theoretic re-ranking and path extraction (e.g., Personalized PageRank, k-shortest paths, Steiner trees) improve recall, coverage, and *explainability* of evidence?
3. Can graph-constrained generation reduce omissions and hallucinations, and increase *structure compliance* and *attribution*?
4. What are the trade-offs between interpretability (path plausibility, attribution) and task metrics (EM/F1 for QA, factuality)?

2 Primary Corpus & Tasks (Fixed)

Primary corpus. A fixed Wikipedia snapshot or equivalent open encyclopedia dump (freeze a specific dump once at project start). Store raw text, page links, and metadata.

Primary tasks.

- **Multi-hop QA (Fixed for main figures).** Use a standard multi-hop QA set (e.g., Hotpot-style distractor setting) to evaluate retrieval, reasoning, and attribution within a *single-query* workflow.[1]
- **Structured Explanation Generation.** For selected topics (e.g., clinical-dermatology educational content), enforce a section blueprint via KG traversal: MECHANISM → EFFICACY → RISKS/SIDE EFFECTS (illustrative only; not medical advice).

Optional validation sets (stress-tests). Additional multi-hop or long-form QA sets to probe robustness (no new methods tuned on them; used only for sanity checks).

3 Milestone 1 (Weeks 1–2): Foundations & Design Spec

Objectives

- Complete a targeted literature review on RAG, graph-augmented retrieval, dynamic KG induction, and controllable/structured generation.
- Fix corpora, tasks, and evaluation splits; define the *QS-KG schema*.
- Produce a v1 design and experiment plan with explicit acceptance criteria.

[1] Use the development split for ablations and hold out a test split for final reporting.

Concrete Steps

1. **Lit review (12–15 core reads).** Cover: retrieval variants (sparse+dense, hybrid), graph-based retrieval, open IE/RE, coreference/NER canonicalization, graph centrality, Steiner trees, controllable generation and content planning.
2. **Tracking sheet.** Create an Excel with columns in Appx. **??** (*Paper Title, Year, Venue, 5–7 sentence Summary, Key Results, Limitations, Relevance (1–5)*); complete at least 15 entries.
3. **Corpus freeze.** Download and freeze the encyclopedia dump; document license; store in `data/` (Appx. **??**).
4. **Task selection.** Finalize the multi-hop QA dev/test splits; document inclusion/exclusion criteria.
5. **QS-KG schema.** Define node/edge types and fields: *Node*={id, surface, canonical_id, spans, doc_ids}; *Edge*={type, weight, evidence_span_ids}. Types: coMention, coref, hyperlink/citation, lexSem (cosine sim), rel (RE triple).
6. **Design doc v1.** 3–5 pages: architecture, modules, metrics, ablations, risks.

Deliverables

- 2–3 page review memo; Excel sheet with 15+ entries.
- Fixed corpus snapshot; fixed dev/test splits; QS-KG schema document.
- Design doc v1 with acceptance tests.

Risks & Mitigations

- *Scope creep* → Keep primary figures on one QA benchmark; all else validation-only.
- *Tooling sprawl* → Prefer one NER/coref/RE stack; freeze versions for reproducibility.

4 Milestone 2 (Weeks 3–4): Dynamic QS-KG Induction

Objectives

- Implement an on-the-fly pipeline that, for a given query q, retrieves top-k passages and induces a weighted, typed multigraph $G_q = (V, E, w)$.
- Produce a compact *subgraph* S_q anchored on query entities with a strict node/edge budget.

Concrete Steps & Guidelines

1. **Hybrid candidate retrieval.** BM25 (sparse) + dense encoder (hybrid scoring). Keep $k \in \{20, 50, 100\}$ as a controllable parameter and log both rankings.
2. **Span & entity layer.** Run sentence splitting, NER, and coreference on the k passages. Create nodes for *canonical* entities and for *claim spans* (key sentences).

3. **Relation candidates.** Induce edges with scores:

$$w(u,v) = \lambda_{\text{co}} \underbrace{\mathrm{PMI}(u,v)}_{\text{coMention}} + \lambda_{\text{sem}} \cos(f(u), f(v)) + \lambda_{\text{rel}}\, s_{\text{RE}}(u \to v) + \lambda_{\text{link}} \mathbf{1}\{\text{hyperlink/cite}\}$$

   Normalize each component to $[0,1]$; store evidence spans per edge.

4. **Query anchoring.** Identify anchor set A_q from query mentions; compute Personalized PageRank (PPR) vector

$$r = \alpha P^{\top} r + (1-\alpha) e_{A_q}, \quad \alpha \in [0.8, 0.95]$$

   where P is the row-normalized adjacency over w.

5. **Subgraph extraction.** Build S_q by growing from A_q with (i) PPR rank, (ii) edge-threshold τ, and (iii) budget $|V(S_q)| \leq N_{\max}$, $|E(S_q)| \leq M_{\max}$. Keep $N_{\max} \in \{30, 50\}$.

6. **Graph JSON & cache.** Emit `S_q.json` (schema fixed), and memoize by a content hash of q and retrieval seeds (TTL cache enabled).

7. **Gold mini-set (25 queries).** Hand-curate light annotations (anchor nodes; 1–2 plausible links) for qualitative checks of S_q path plausibility.

Acceptance Tests & Deliverables

- $\geq 95\%$ of `S_q.json` files validate against the schema; average build time < 1.5s for $k \leq 50$ on CPU.
- Visual dumps (PDF) of S_q for 10 random dev queries with node labels and top-10 edges by w.
- Module tests for NER/coref/RE and PPR; reproducibility seeds recorded.

Risks & Mitigations

- *Noisy edges* → Use edge-type-specific thresholds; keep RE optional but logged.
- *Latency* → Lower k, raise τ, or precompute doc-side spans and embeddings.

5 Milestone 3 (Weeks 5–6): Graph-Ranked Retrieval & Evidence Chains

Objectives

- Re-rank evidence using graph centrality; select a budgeted set of passages maximizing coverage of graph nodes relevant to q.
- Extract interpretable *evidence chains* as paths/subgraphs connecting anchors to answer candidates.

Concrete Steps & Guidelines

1. **Centrality re-ranking.** Use PPR scores on S_q to rank nodes; map nodes to supporting passages; re-rank passages by (node score aggregate + baseline retrieval score).

2. **Budgeted maximum coverage.** Select B passages via greedy facility-location objective

$$\max_{X:|X|\leq B} \sum_{u\in V(S_q)} \max_{p\in X} \mathrm{rel}(u,p),$$

   where rel is node–passage support; report the $(1-1/e)$ greedy guarantee.

3. **Path extraction.** Compute k-shortest paths (edge weights $1/w$) from A_q to top answer nodes; also compute a Steiner *treelet* covering anchors and top-m nodes (prize-collecting if needed).

4. **Answer candidate scoring.** For multiple-choice QA, compute option log-likelihoods conditioned on the *selected evidence set*; record margins and evidence paths.

5. **Logging.** Persist: selected passages, covered nodes, paths (node/edge ids + evidence spans), and timing.

Metrics & Checks

- **Retrieval:** Recall@k, % nodes covered from a weak silver set (entity overlap with answer), average path length, node coverage under budget B.
- **Interpretability:** Path plausibility (human 3-point scale on the 25-query mini-set), edge-type composition, and redundancy (distinctiveness of top-k paths).

Deliverables

- Re-ranked evidence selector with coverage objective; path-extraction library.
- Tables/figures: retrieval recall vs. budget; path stats; ablation on edge types on/off.

6 Milestone 4 (Weeks 7–8): Graph-Guided (Constrained) Generation

Objectives

- Convert paths/subgraphs into *section blueprints* that drive generation order and content boundaries.
- Enforce *attribution* and *structure compliance* during decoding.

Concrete Steps & Guidelines

1. **Blueprint compiler.** Map a path $\pi = (v_1, \ldots, v_T)$ to a section sequence $(s_1, \ldots, s_T)$ with titles from node types (e.g., MECHANISM → EFFICACY → SIDE EFFECTS). Allow a small fanout by grouping adjacent nodes of the same type.

2. **Section-conditional prompts.** For each section s_t, build a prompt with *only* the evidence spans attached to nodes/edges included in s_t. Require inline citations like "[#1]" that map to passage ids.

3. **Controller loop.** Generate sections sequentially: for $t = 1..T$, (i) retrieve evidence for s_t, (ii) generate y_t, (iii) run a light *citation check* that each factual claim references at least one allowed span (retry once if not).

4. **Stop conditions.** Enforce max tokens per section and overall; hard stop if attribution ratio (claims with cites / total claims) $< \gamma$.
5. **Fallbacks.** If no valid path exists, use the PPR-ranked node list to form a *flat blueprint.*

Structure & Faithfulness Metrics

- **Structure Compliance (SC).** Fraction of required sections present and in the prescribed order; compute sequence F1 vs. blueprint.
- **Attribution Rate (AR).** # factual statements with citations / total factual statements (use simple claim segmentation).
- **Faithfulness Error (FE).** Share of cited claims contradicted by their evidence (binary contradiction classifier on claim–span pairs).

Deliverables

- Graph-to-blueprint compiler; sectioned generator; citation checker; JSON outputs with section headers, text, and evidence ids.
- Example report for the *acne treatment* template showing ordered traversal and citations (illustrative).

7 Milestone 5 (Weeks 9–10): Evaluation, Ablations & Paper

Objectives

- Run complete evaluations and ablations; finalize figures/tables and a workshop-style manuscript.

Exact Ablations (one factor at a time)

1. **Retrieval budget:** $k \in \{20, 50, 100\}$; coverage budget $B \in \{5, 10, 20\}$.
2. **Edge types:** on/off per type (coMention, coref, hyperlink, lexSem, rel); re-estimate w.
3. **Graph methods:** centrality (PPR vs. degree), path extraction (k-shortest vs. Steiner).
4. **Generation:** graph-guided vs. standard RAG; with vs. without citation checker; blueprint fanout size.
5. **Model scale:** small vs. medium instruction-tuned LMs for generation (same prompts).

Primary Metrics

- **Task:** EM/F1 on QA; answer logit margin.
- **Retrieval:** Recall@k, coverage of answer-supporting entities, average path length.
- **Interpretability:** Path plausibility (3-point human rating on 50 items), *AR*, *SC*, and *FE*.
- **Efficiency:** QS-KG build time, tokens used, sections generated.

Manuscript & Artifacts

- 6–8 page paper draft with method diagram, retrieval/coverage plots, structure/attribution results, and qualitative examples.
- Open-source code with `README`, config files, and `results/` (logs, tables, figures).

Risks & Mitigations

- *No gains vs. strong RAG* → Emphasize interpretability: path quality, attribution, and structure metrics; introduce section-completeness tasks where gains are expected.
- *Hallucinations persist* → Tighten evidence gating; raise citation thresholds; increase blueprint granularity.

Reproducibility Checklist

- Fixed corpus snapshot & hashes; seeds for retrieval and decoding.
- Version-locked NER/coref/RE/embedding models.
- Deterministic data loaders; cached `S_q.json` artifacts; experiment tracking (CSV).