PropRAG: Guiding Retrieval with Beam Search over Proposition Paths

Anonymous ACL submission

Abstract

Retrieval Augmented Generation (RAG) has become the standard non-parametric approach for equipping Large Language Models (LLMs) with up-to-date knowledge and mitigating catastrophic forgetting common in continual learning. However, standard RAG, relying on independent passage retrieval, fails to capture the interconnected nature of human memory crucial for complex reasoning (associativity) and contextual understanding (sense-making). While structured RAG methods like HippoRAG 2 utilize knowledge graphs built from triples, we argue that the inherent context loss of knowledge triples limits fidelity. We introduce **PropRAG**, leveraging context-rich propositions and a novel LLM-free online beam search over proposition paths to find multi-step reasoning chains. PropRAG achieves state-of-the-art zero-shot Recall@5 and F1 scores on 2Wiki, HotpotQA, and MuSiQue, advancing non-parametric continual learning by improving evidence retrieval through richer representation and efficient reasoning path discovery.

1 Introduction

Large Language Models (LLMs) face challenges in continual learning, such as catastrophic forgetting (Cohen et al., 2024; Gu et al., 2024). Retrieval Augmented Generation (RAG) (Lewis et al., 2020) offers a non-parametric solution by retrieving external knowledge. However, conventional RAG systems (Karpukhin et al., 2020; Lee et al., 2025), retrieving evidence independently, struggle with multi-step queries requiring interconnected information for sense-making (Klein et al., 2006) and associativity (Suzuki, 2007).

Structured RAG methods, like HippoRAG 2 (Gutiérrez et al., 2025) using triple-based KGs, improve multi-hop retrieval but suffer "context collapse" from lossy triples (Section 3). Other advanced RAG strategies use online LLM calls during retrieval (Trivedi et al., 2022a; Jiang et al., 2024), introducing latency, cost, and consistency issues.

We introduce **PropRAG**, a novel RAG framework for dynamic, interconnected memory retrieval without online LLM inference during the search process. PropRAG features two key innovations:

1. **Propositions as High-Fidelity Knowledge Units:** Extracted offline by an LLM, propositions preserve contextual richness unlike triples (Section 3).
2. **LLM-Free Online Beam Search for Reasoning Path Discovery:** A novel algorithmic beam search discovers and scores paths of interconnected propositions from a pre-built graph using embeddings and graph structure, avoiding LLM inference costs and potential inconsistencies during evidence discovery (Section 5.3).

PropRAG's LLM-independent online path finding over richer propositions enhances sense-making and associativity. Experiments show substantial outperformance, especially on multi-hop QA, setting new state-of-the-art zero-shot RAG scores and advancing non-parametric continual learning.

2 Related Work

Retrieval Augmented Generation (RAG) frameworks (Lewis et al., 2020) augment LLMs by retrieving documents. Early methods like DPR (Karpukhin et al., 2020) used embedding similarity. Despite better embeddings (Izacard et al., 2022; Ni et al., 2022; Lee et al., 2025), standard RAG struggles with multi-document synthesis (Asai et al., 2020).

Multi-Hop RAG aims to address this. Iterative methods (Asai et al., 2020; Trivedi et al., 2022a) retrieve sequentially. Graph-based RAG (Edge et al., 2024; Sarthi et al., 2024) uses KGs. HippoRAG 2 (Gutiérrez et al., 2025) used Personalized PageRank over triple-based KGs, but triples are

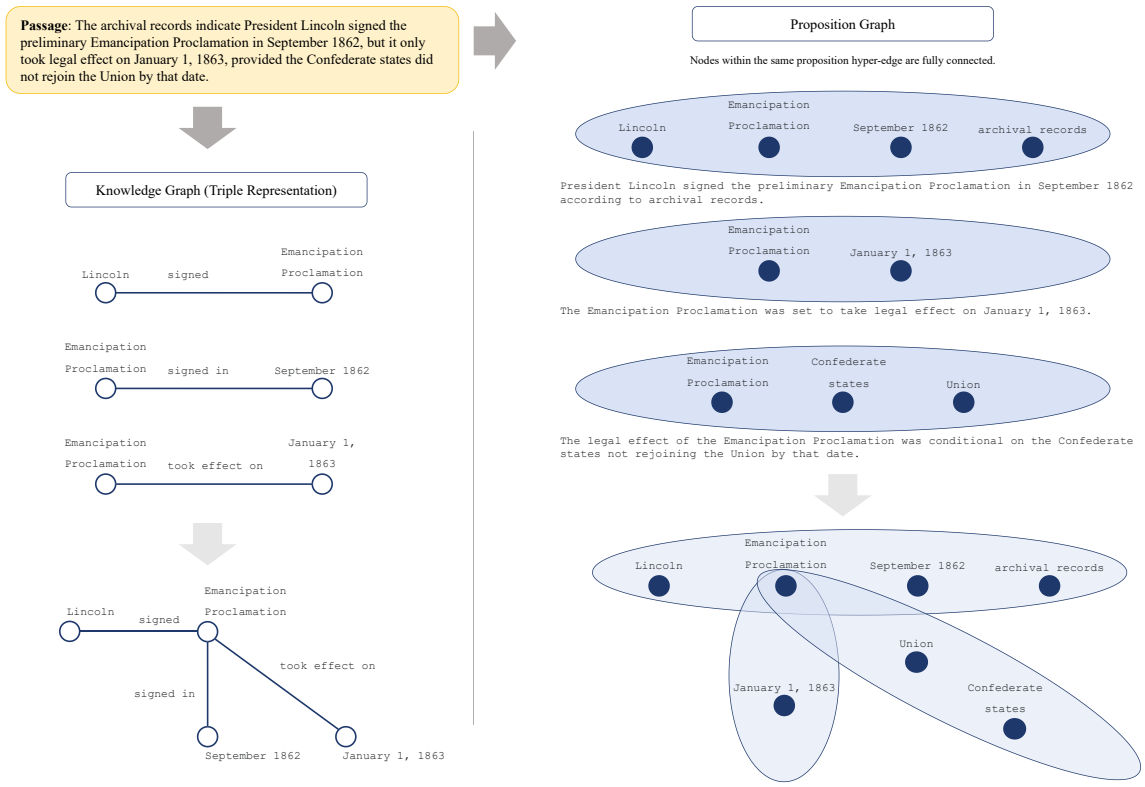

Figure 1: Comparison of Knowledge Graph vs Proposition Graph for a complex passage. Left: Traditional triple-based KG struggles to represent provenance ("archival records") and conditional clauses naturally, requiring sparse connections or complex reification. Right: PropRAG's proposition graph uses hyper-edges (fully connected cliques within shaded ovals) to link all entities co-occurring within each contextually rich proposition, preserving nuances like conditionality and provenance directly.

context-poor. PropRAG differs by using context-rich propositions and an explicit, LLM-free online beam search for path discovery on the proposition graph.

Beam Search Recent work has explored leveraging beam search to improve multi-hop retrieval. For instance, (Zhang et al., 2023) proposed Beam Retrieval, an end-to-end trainable framework where beam search is used during both training and inference to find optimal passage sequences. Their approach learns a scoring function via classification heads optimized across hops using ground-truth passage chains within a reading comprehension setting.

In contrast, PropRAG adopts a fundamentally different, zero-shot online retrieval strategy. While also employing beam search, PropRAG operates over a graph of contextually rich propositions, extracted offline using an LLM. Our beam search algorithmically discovers proposition paths based on pre-computed embedding similarity and graph connectivity, crucially avoiding online LLM inference costs, potential inconsistencies or task-specific training during the retrieval phase. The goal is to find semantically relevant reasoning chains (proposition paths) which then inform a final passage ranking, typically via Personalized PageRank seeded by the discovered paths. PropRAG thus decouples the complex reasoning path discovery from end-to-end training dependencies, focusing on leveraging richer knowledge units (propositions) and algorithmic path exploration for improved zero-shot multi-hop RAG performance.

Propositions as Retrieval Units Chen et al. (2024) showed propositions (atomic factoids) outperform passage/sentence retrieval. PropRAG integrates such propositions into a graph and uses beam search for multi-hop reasoning.

3 Propositions: Escaping the Tyranny of the Triple

The fidelity of knowledge representation is paramount for multi-hop reasoning. Traditional KG-based RAG often uses (Subject, Predicate, Object) triples, a lossy compression discarding cru-

cial nuances. Propositions—atomic, self-contained statements preserving context—offer a richer alternative. Key limitations of triples addressed by propositions include:

1. **Loss of Conditional Context:** Many facts are true only under specific conditions (temporal, spatial, etc.). Triples often discard this.
    - *Example Passage:* The experimental drug showed promise in Phase 2 trials, reducing tumor size significantly, but only in patients under 50 with the specific KRAS mutation."
    - *Triples might yield:* `(drug, showed, promise)`, `(drug, reduced, tumor size)`. The conditions "only in patients under 50" and "with specific KRAS mutation" are lost.
    - *Propositions preserve conditions:* E.g., `"The significant tumor size reduction by the experimental drug was observed only in patients under 50."`
2. **Awkward Representation of Clauses and Meta-Relations:** Representing complex relationships (provenance, causality, conditions within clauses) is unnatural in S-P-O format, often requiring complex reification that loses clarity.
    - *Example Passage:* "The archival records indicate President Lincoln signed the preliminary Emancipation Proclamation in September 1862, but it only took legal effect on January 1, 1863, provided the Confederate states did not rejoin the Union by that date."
    - *Triples struggle* with provenance ("Archival records indicate...") or the conditional clause.
    - *Propositions handle these naturally:* E.g., `"Archival records indicate President Lincoln signed..."`, `"The legal effect... was conditional on..."`. The proposition graph (Figure 1, right) uses hyper-edges preserving full context.
3. **Unary Predicates and Attributes:** Simple properties (e.g., "The ancient manuscript was fragile") are awkward in binary S-P-O. Triples like `(manuscript, is, fragile)` or `(manuscript, hasProperty, fragile)` treat attributes as entities or use weak/generic predicates.
    - *Propositions capture unary predication directly:* `"The ancient manuscript was fragile."` (Entity: ancient manuscript). The embedding of the full proposition preserves the nuanced relationship.
4. **N-ary Relationships (Events with Multiple Participants):** Real-world events often involve more than two entities (e.g., collaborations, transactions). Rigid S-P-O forces unnatural decomposition into multiple binary triples, fragmenting the event's holistic view.
    - *Example Passage (Collaboration):* "The groundbreaking research paper on quantum entanglement was co-authored by Alice, Bob, and Charlie in 2023."
    - *Triples fragment this:* `(Alice, co-authored, paper)`, `(Bob, co-authored, paper)`, etc., losing that they collaborated *together*.
    - *Proposition captures N-ary relation:* `"Alice, Bob, and Charlie co-authored the groundbreaking research paper on quantum entanglement in 2023."` (Entities: Alice, Bob, Charlie, paper, etc.). This is represented as a hyper-edge in the proposition graph.

PropRAG uses LLMs (Llama-3.3-70B-Instruct) **offline** for high-quality proposition extraction, leveraging NLU capabilities during indexing. The subsequent online retrieval does not involve further LLM calls for knowledge representation.

4 Problem Formulation: Finding the Reasoning Path

Traditional RAG aims to retrieve documents D_{ret} maximizing individual relevance $\text{sim}(\text{emb}(d), \text{emb}(q))$. Multi-hop KG-RAG like HippoRAG 2 (Gutiérrez et al., 2025) ranks nodes (entities/passages) by proximity to query seeds or graph centrality, but does not explicitly construct or evaluate multi-step reasoning paths. When KGs use context-poor triples, semantic richness is limited.

We reformulate multi-hop retrieval as finding an optimal *path* of interconnected propositions $P = (p_1, p_2, ..., p_k)$ that collectively answer query

q. Propositions p_i, p_{i+1} are linked by shared/synonymous entities. This connection occurs naturally in the proposition graph (Figure 1, right) where entities shared between proposition hyper-edges act as bridges. The objective is to find $P^* = \underset{P \in \text{ConnectedPaths}(\mathcal{P})}{\text{argmax}} \text{Score}(P, q)$, where $\text{Score}(P, q)$ measures path relevance, possibly via aggregated embeddings. As finding the global optimum is intractable, we use beam search as a heuristic.

5 Methodology: PropRAG

PropRAG implements path-finding via offline indexing and online two-stage retrieval (Figure 2).

5.1 Offline Indexing (LLM-assisted)

1. Use an LLM (e.g., Llama-3.3-70B-Instruct) to extract propositions and constituent entities from corpus $\mathcal{D}$.
2. Construct a proposition graph $G = (V, E)$ (detailed in Section A.1).
3. Compute and store embeddings for all passages, entities, and propositions.

5.2 Online Retrieval (LLM-free)

PropRAG's online retrieval component executes a two-stage, LLM-free process to identify relevant reasoning paths (Figure 2). This staged approach balances broad exploration with focused path refinement.

5.2.1 Stage 1: Coarse-grained Subgraph Focusing

The objective of this stage is to efficiently prune the search space to a highly relevant subgraph of the proposition graph G.

1. **Initial Candidate Identification:** The top-N_{prop} propositions most semantically similar to the query q are retrieved. Entities $\mathcal{E}(p)$ within these propositions are extracted.
2. **Seed Selection ($S_{initial}$):** From the extracted entities, the top-N_{entity} unique entities, ranked by their initial relevance scores (Appendix A.6), form the set $S_{initial}$. These seeds are assigned uniform weights for the subsequent PPR to foster diverse graph exploration.
3. **Exploratory Graph Traversal (PPR):** Personalized PageRank (PPR) is performed on the full graph G. Reset probabilities are concentrated exclusively on nodes in $S_{initial}$ (1.0 for seeds, 0 otherwise). A high damping factor (e.g., 0.75) is employed to encourage wider traversal from these initial seeds.
4. **Focused Subgraph (G_{sub}) Generation:** The top-K passages identified by the initial PPR, along with all entities connected to them in G, constitute the focused subgraph G_{sub}, which serves as the operational graph for fine-grained reasoning.

5.2.2 Stage 2: Fine-grained Path Discovery and Ranking

Within G_{sub}, this stage explicitly discovers, scores, and ranks multi-proposition reasoning paths.

1. **Beam Search Path Exploration:** The beam search algorithm (Section 5.3) systematically explores G_{sub} to identify connected proposition paths up to a maximum length L_{max}. It maintains a beam of the top-B paths based on query relevance.
2. **Refined Seed Set Construction ($S_{final} = S_{explore} \cup S_{exploit}$):** The final set of seed entities for ranking integrates relevance from two sources:
   - *Exploration Seeds ($S_{explore}$):* Top-$B_{initial}$ entities from the top-$P_{initial}$ query-similar propositions within G_{sub}.
   - *Exploitation Seeds ($S_{exploit}$):* Top-B_{beam} entities identified as central to the top-P_{beam} high-scoring paths discovered by beam search.
3. **Exploitative Path-informed Ranking (PPR):** A second PPR iteration is confined to G_{sub}. The reset probability vector $\mathbf{r}$ combines normalized initial query-entity similarity scores ($\mathbf{s}_{init}$) and aggregated entity scores derived from beam search paths ($\mathbf{s}_{beam}$), as detailed in Appendix A.6. Specifically, for $e \in S_{final}$, $score_{final}(e) = \max(s'_{explore,e}, s'_{exploit,e})$; passage nodes d also receive a base score $score_{final}(d) = \lambda_{passage} \cdot score_{dpr}(d, q)$. A lower damping factor (e.g., 0.45) is used to capitalize on these refined relevance signals, yielding a focused ranking of the top 5 evidence passages.
4. **Final Evidence Ranking:** Passages are ranked according to their final PPR scores, promoting those linked to entities within salient reasoning paths.

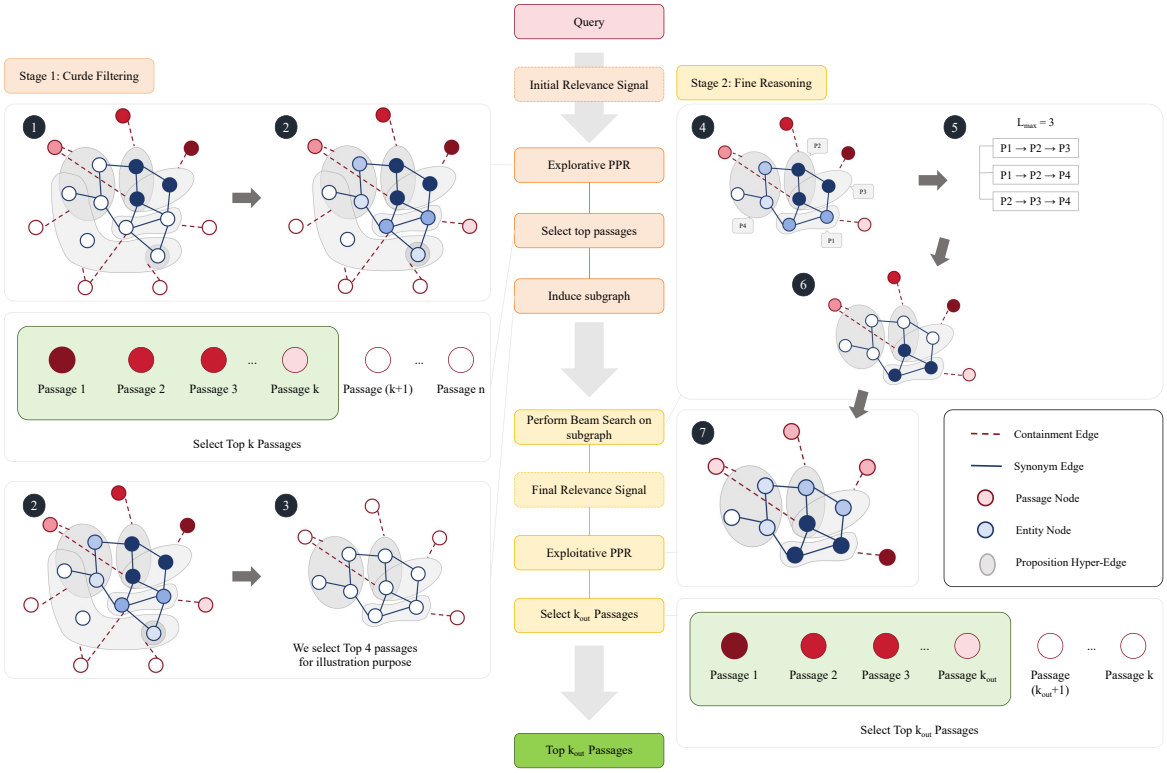

Figure 2: The two-stage retrieval process of PropRAG. Stage 1 (Crude Filtering) uses explorative PPR (high damping factor = 0.75) on the full graph to identify top $k = 50$ passages and induce a relevant subgraph. Stage 2 (Fine Reasoning) performs beam search on the subgraph to discover reasoning paths, generates final relevance signals, applies exploitative PPR (low damping factor = 0.45) on the subgraph using refined seeds, and selects the final top $k_{out} = 5$ passages.

5.3 Beam Search for Reasoning Paths

PropRAG employs a beam search algorithm, adapted from sequence generation methodologies, to heuristically discover high-relevance reasoning paths within the proposition graph. This approach systematically explores sequences of propositions $P = (p_1, ..., p_L)$ that maximize a relevance score $\text{Score}(P, q)$ with respect to the input query q, performing a bounded-width search.

Core Algorithmic Steps:

- **Initialization:** The beam is initialized with paths of length 1, corresponding to the propositions most semantically similar to the query.
- **State Representation:** Each hypothesis in the beam comprises a partial path $P = (p_1, ..., p_k)$, its cumulative relevance score $\text{Score}(P, q)$, and the entities $\mathcal{E}(p_k)$ in its terminal proposition p_k.
- **Path Expansion:** Paths are extended by identifying candidate next propositions p_{next}. These candidates are primarily drawn from propositions connected to the current path's terminal proposition p_k via shared or synonymous entities within G_{sub}. To mitigate local optima and encourage diverse exploration, the top-3 initial query-relevant propositions are also considered as potential p_{next} ("jump points"), irrespective of direct graph connectivity to p_k.
- **Path Scoring:** The relevance of an expanded path $P_{new} = (P, p_{next})$ is estimated as $\text{Score}(P_{new}, q) \approx \text{sim}(\text{emb}(P_{new}), \text{emb}(q))$. An efficient average proposition embedding is used for initial scoring. The top-M candidates from this initial scoring are then re-evaluated using a more robust score derived from an embedding of the concatenated text of all propositions in P_{new}.
- **Selection and Pruning:** From all generated candidate expansions, the top-B (beam width) paths with the highest scores are retained for the subsequent iteration; others are pruned.
- **Termination Criteria:** The search concludes when paths reach the maximum length L_{max} or

Query: What year did the Governor of the city where the basilica named after the same saint as the one that Mantua Cathedral is dedicated to die?
Gold Answer: 1952
Top 5 Initial Propositions (Score - Text):

- 0.4275 - Mantua Cathedral is a Roman Catholic cathedral dedicated to Saint Peter.
- 0.3851 - Mantua Cathedral is the seat of the Bishop of Mantua.
- 0.3474 - Mantua Cathedral is located in Mantua, Lombardy, northern Italy.
- 0.3372 - Foligno Cathedral is dedicated to the patron saint of the city, Felician of Foligno (San Feliciano).
- 0.3135 - No successor was appointed to the post of Governor of Vatican City after Marchese Camillo Serafini's death in 1952.

Beam Search Depth 2/3 (Top 3 Paths):

- 0.4792 - Mantua Cathedral... dedicated to Saint Peter. → St. Peter's Basilica is located in Vatican City.
- 0.4756 - Mantua Cathedral... dedicated to Saint Peter. → Alfredo Ormando died on 23 January 1998 in Rome.
- 0.4705 - Mantua Cathedral... dedicated to Saint Peter. → The Italian name for St. Peter's Basilica is...

Beam Search Depth 3/3 (Top 3 Paths):

- **0.5989 - Mantua Cathedral... dedicated to Saint Peter. → St. Peter's Basilica is located in Vatican City. → No successor was appointed... after Marchese Camillo Serafini's death in 1952.**
- 0.5674 - Mantua Cathedral... dedicated to Saint Peter. → St. Peter's Basilica is located in Vatican City. → Marchese Camillo Serafini held the post of Governor... until his death in 1952.
- 0.5458 - Mantua Cathedral... dedicated to Saint Peter. → St. Peter's Basilica is located in Vatican City. → The post of Governor of Vatican City was not mentioned...

Observation: The crucial link "St. Peter's Basilica is located in Vatican City" (low initial relevance) is found during beam search (Depth 2), enabling discovery of the full reasoning path by Depth 3.

Figure 3: Example beam search process ($L_{max} = 3$) for a MuSiQue query. Full text of propositions abridged for fit.

when no valid expansions can be generated for any hypothesis in the beam.

This online beam search operates entirely on precomputed embeddings and the graph structure, ensuring efficient inference by **avoiding any LLM calls**. Figure 3 offers a conceptual illustration of this path discovery process.

6 Experiments

6.1 Setup

Datasets: We evaluate on NaturalQuestions (NQ) (Wang et al., 2024), PopQA (Mallen et al., 2023), and for multi-hop reasoning: 2WikiMultihopQA (2Wiki) (Ho et al., 2020), HotpotQA (Yang et al., 2018), and MuSiQue-Ans (Trivedi et al., 2022b). We use 1000-query samples and corpora from Gutiérrez et al. (2025) for comparability and limitation of experimental cost.

Baselines: Comparisons include classic retrievers (BM25, Contriever, GTR), large embedding models (GTE-Qwen2, GritLM, NV-Embed-v2), and structure-augmented RAG (RAPTOR, GraphRAG, LightRAG, HippoRAG, HippoRAG 2). Baseline results are primarily from Gutiérrez et al. (2025). Citations for all baselines are in the full version/appendix.

Implementation: PropRAG uses Llama-3.3-70B-Instruct (AI@Meta, 2024) for offline proposition extraction and QA, and NV-Embed-v2 (7B) (Lee et al., 2025) for embeddings. Key parameters: Beam width $B = 4$, max path length $L_{max} = 3$. Details in Appendix A.2.

Metrics: Passage Recall@5; QA F1 score and Exact Match (EM) per MuSiQue scripts (Trivedi et al., 2022b).

6.2 Results and Discussion

Tables 1 (Recall@5) and 2 (F1 Score) present PropRAG's performance.

Overall Performance: PropRAG ($L_{max} = 3, B = 4$) achieves a state-of-the-art average F1 of 64.9%, outperforming HippoRAG 2 by 2.0 points and NV-Embed-v2 by 6.9 points. This highlights the synergy of context-rich propositions and LLM-free online path discovery.

Impact of Two-Stage Proposition Retrieval ($L_{max} = 1$): Even with no path expansion ($L_{max} = 1$), PropRAG's two-stage process using propositions significantly improves over baselines (e.g., +4.0% MuSiQue F1 over HippoRAG 2). This demonstrates the benefit of focused subgraph search with propositions, even before extensive beam search. A direct comparison of single-stage proposition vs. triple retrieval will be explored in ablation studies.

Benefit of Multi-Step Beam Search ($L_{max} > 1$): Explicit path discovery via beam search further boosts performance. Compared to $L_{max} = 1$, $L_{max} = 3$ increases average F1 by +0.9% (to 64.9%) and MuSiQue Recall@5 by +2.7% (to 78.3%). This confirms that exploring 2-3 hop paths uncovers crucial evidence. $L_{max} = 3$ is optimal.

Simpler QA Performance: On NQ and PopQA, PropRAG remains robust (e.g., 62.5% NQ F1, 56.4% PopQA F1 with $L_{max} = 3$), showing no

Table 1: Passage Retrieval Performance (Recall@5). Baselines from Gutiérrez et al. (2025). Best overall in bold, best PropRAG variant also bolded if different.

Method	NQ	PopQA	MuSiQue	2Wiki	HotpotQA
Simple Baselines					
BM25	56.1%	35.7%	43.5%	65.3%	74.8%
Contriever	54.6%	43.2%	46.6%	57.5%	75.3%
GTR (T5-base)	63.4%	49.4%	49.1%	67.9%	73.9%
Large Embedding Models					
GTE-Qwen2-7B	74.3%	50.6%	63.6%	74.8%	89.1%
GritLM-7B	76.6%	50.1%	65.9%	76.0%	92.4%
NV-Embed-v2 (7B)	75.4%	51.0%	69.7%	76.5%	94.5%
Structure-Augmented RAG					
RAPTOR	68.3%	48.7%	57.8%	66.2%	86.9%
HippoRAG	44.4%	53.8%	53.2%	90.4%	77.3%
HippoRAG 2	78.0%	51.7%	74.7%	90.4%	96.3%
PropRAG (Ours)					
$L_{max} = 1$	**78.4%**	56.3%	75.6%	92.0%	95.7%
$L_{max} = 2$	78.1%	56.1%	77.6%	93.4%	97.2%
$L_{max} = 3$	77.9%	**56.2%**	**78.3%**	**94.1%**	**97.4%**
$L_{max} = 4$	77.8%	56.0%	77.6%	93.7%	97.0%

Table 2: End-to-End QA Performance (F1 Score) with Llama-3.3-70B-Instruct Reader. Baselines from Gutiérrez et al. (2025). Best overall in bold, best PropRAG variant also bolded if different.

Method	NQ	PopQA	MuSiQue	2Wiki	HotpotQA	Avg
No Retrieval (Parametric)						
Llama-3.3-70B-Instruct	54.9%	32.5%	26.1%	42.8%	47.3%	40.7%
Simple Baselines						
Contriever	58.9%	53.1%	31.3%	41.9%	62.3%	49.5%
GTR (T5-base)	59.9%	56.2%	34.6%	52.8%	62.8%	53.3%
Large Embedding Models						
GTE-Qwen2-7B	62.0%	56.3%	40.9%	60.0%	71.0%	58.0%
GritLM-7B	61.3%	55.8%	44.8%	60.6%	73.3%	59.2%
NV-Embed-v2 (7B)	61.9%	55.7%	45.7%	61.5%	75.3%	58.0%
Structure-Augmented RAG						
RAPTOR	50.7%	56.2%	28.9%	52.1%	69.5%	51.5%
GraphRAG	46.9%	48.1%	38.5%	58.6%	68.6%	52.1%
LightRAG	16.6%	2.4%	1.6%	11.6%	2.4%	6.9%
HippoRAG	55.3%	55.9%	35.1%	71.8%	63.5%	56.3%
HippoRAG 2	**63.3%**	56.2%	48.6%	71.0%	75.5%	62.9%
PropRAG (Ours)						
$L_{max} = 1$	62.2%	56.1%	52.6%	73.5%	75.7%	64.0%
$L_{max} = 2$	61.9%	56.1%	53.4%	74.9%	76.0%	64.4%
$L_{max} = 3$	62.5%	**56.4%**	**53.9%**	**75.3%**	**76.1%**	**64.9%**
$L_{max} = 4$	62.8%	56.0%	53.0%	75.3%	76.1%	64.7%

degradation on tasks with less multi-hop dependency.

6.3 Ablation Studies

Ablations (Tables 3 and 4) on PropRAG (default $L_{max} = 3, B = 4$) validate key design choices.

Beam Width (B): Increasing B from 1 to 4 boosts average R@5 by +1.4% and F1 by +0.6%. $B = 4$ offers a strong balance, while $B = 5$ shows slight F1 gains at the cost of some recall consistency.

Propositions vs. Triples (Stage 1 PPR): To isolate the benefit of propositions before multi-step beam search, we compare PropRAG using only its first stage PPR (effectively, PPR on the proposition-based graph with parameters similar to HippoRAG 2's single PPR stage) against HippoRAG 2 (no filter, using triples and PPR, results from Gutiérrez et al. (2025) Table 4). PropRAG (Stage 1 PPR only) achieves an average Recall@5 of 87.2% compared to 86.4% for HippoRAG 2 (+0.8%). Specifically on MuSiQue, PropRAG (Stage 1) gets 75.4% vs.

Table 3: Ablation Study on Beam Width (B) using $L_{max} = 3$. Default $B = 4$. Performance shown as Recall@5 / F1 Score.

Beam Width (B)	MuSiQue (R@5 / F1)	2Wiki (R@5 / F1)	HotpotQA (R@5 / F1)	Average (R@5 / F1)
1 (Greedy Search)	76.6% / 52.9%	92.1% / 74.5%	97.0% / 76.2%	88.5% / 67.9%
2	77.4% / 52.9%	**94.4% / 75.8%**	97.4% / 75.7%	89.7% / 68.1%
3	78.0% / 53.1%	94.2% / 75.7%	**97.5% / 76.2%**	89.9% / 68.3%
4 (Default)	**78.3%** / 53.9%	94.1% / 75.3%	97.4% / 76.1%	**89.9%** / 68.5%
5	77.8% / **54.4%**	93.7% / 75.4%	97.4% / 76.1%	89.6% / **68.6%**
6	77.8% / 53.9%	93.2% / 74.6%	97.2% / 75.7%	89.4% / 68.1%

Table 4: Full Ablation Study Results (Recall@5). PropRAG uses $L_{max} = 3, B = 4$ unless noted. HippoRAG 2 (no filter) results from Gutiérrez et al. (2025) Table 4.

Configuration	MuSiQue	2Wiki	HotpotQA	Avg
Full PropRAG ($L_{max} = 3, B = 4$)	**78.3%**	**94.1%**	**97.4%**	**89.9%**
Comparison Baselines				
HippoRAG 2 (Triples, PPR, no filter)	73.0%	90.7%	95.4%	86.4%
PropRAG (Stage 1 PPR only, same parameters as HippoRAG 2)	75.4%	90.4%	95.9%	87.2%
Retrieval Strategy Ablations ($L_{max} = 3, B = 4$)				
PropRAG (Exploration Seeds Only)	75.6%	92.0%	95.6%	87.7%
PropRAG (Exploitation Seeds Only)	77.9%	91.4%	97.6%	89.0%
PropRAG (Allow all unconnected candidates in beam search)	77.4%	92.9%	96.6%	89.0%

73.0% for HippoRAG 2 (+2.4%). This confirms that the richer context in propositions provides a stronger foundation for graph-based retrieval even in a simpler, single-PPR setup.

Graph Guidance in Beam Search: The "Allow unconnected candidates in beam search" ablation tests the importance of graph structure guiding the beam search. In this setting, candidate propositions for path expansion are chosen based purely on embedding similarity to the query or previous step, without requiring graph connectivity (beyond the initial top-3 query-relevant jumps). This configuration achieves an average Recall@5 of 89.0%, which is 0.9% lower than the full PropRAG (89.9%) that primarily considers graph-connected propositions. The drop is notable on MuSiQue (77.4% vs. 78.3%) and 2Wiki (92.9% vs. 94.1%). This demonstrates that leveraging the explicit connections in the proposition graph effectively guides the beam search towards more relevant reasoning paths, rather than relying solely on semantic similarity which can be noisy.

Seed Strategy: Balanced seeding (exploration + exploitation seeds) outperforms using only one type, yielding the best average R@5 (89.9%).

These ablations confirm the contributions of propositions, graph-guided beam search, two-stage retrieval, and balanced seeding.

6.4 Qualitative Analysis

Figure 3 qualitatively shows beam search identifying a crucial, low-initial-relevance intermediate proposition, enabling the discovery of the full reasoning path without online LLM intervention.

7 Conclusion

PropRAG represents a significant advancement in RAG by shifting from context-poor triples to richer propositions and introducing a novel, LLM-free online beam search mechanism for discovering multi-step reasoning paths. This dual approach demonstrably improves the quality of retrieved evidence, particularly for complex multi-hop queries. Our experiments show that PropRAG sets new state-of-the-art results for zero-shot RAG systems on several challenging benchmarks, enhancing both retrieval recall and end-to-end QA F1 scores. The framework's ability to perform sophisticated evidence gathering without incurring online LLM inference costs is a key advantage. PropRAG underscores the value of explicit, algorithmic modeling of reasoning processes over high-fidelity, pre-structured knowledge, offering a promising direction for developing LLMs with more robust, associative, and dynamic non-parametric memory.

Limitations

PropRAG's primary limitations include the computational overhead of beam search, which, while LLM-free online, is more intensive than simpler retrieval methods. The system's performance is sensitive to the quality of the offline proposition extraction phase; errors or omissions here can propagate. Although online LLM calls are avoided during retrieval, the initial proposition generation relies on an LLM, and its quality can influence downstream results. Furthermore, the graph construction process, particularly the accuracy of entity linking and synonymy detection, plays a crucial role and can be a source of error. The current path scoring relies on embedding similarity, which might not capture all semantic nuances required for perfect path evaluation.

References

AI@Meta. 2024. Llama 3 model card.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations (ICLR)*.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense x retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177.

Roni Cohen, Eran Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics (TACL)*, 12:283–298.

Derek Edge, Hoang Trinh, Nicholas Cheng, Joshua Bradley, Allen Chao, Ajay Mody, Shayne Truitt, and Jason Larson. 2024. From local to global: A graph rag approach to query-focused summarization. http://arxiv.org/abs/2404.16130. ArXiv:2404.16130.

Jia-Chen Gu, Hao-Xiang Xu, Jia-Yu Ma, Pu Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 16801–16819.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. http://arxiv.org/abs/2502.14802. ArXiv:2502.14802.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 6609–6625.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions of the Association for Computational Linguistics (TACL)*, 10:641–655.

Zhouyu Jiang, Mengshu Sun, Lei Liang, and Zhiqiang Zhang. 2024. Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach. http://arxiv.org/abs/2407.13101. ArXiv:2407.13101.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Gary Klein, Brian Moon, and Robert R. Hoffman. 2006. Making sense of sensemaking 1: Alternative perspectives. *IEEE Intelligent Systems*, 21(4):70–73.

Chang-Bin Lee, Rishav Roy, Mengjiao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Nv-embed-v2: Improved techniques for training llms as generalist embedding models. http://arxiv.org/abs/2405.17428. ArXiv:2405.17428.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9802–9822.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9844–9855.

Priyanka Sarthi, Sameer Abdullah, Abhilash Tuli, Sakshi Khanna, Abhijit Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*.

Wendy A. Suzuki. 2007. Making new memories: the role of the hippocampus in new associative learning. *Annals of the New York Academy of Sciences*, 1097:1–11.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022b. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics (TACL)*, 10:539–554.

Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. 2024. Rear: A relevance-aware retrieval-augmented framework for open-domain question answering. *arXiv preprint arXiv:2402.17497*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2369–2380.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. 2023. End-to-end beam retrieval for multi-hop question answering. *arXiv preprint arXiv:2308.08973*.

A Appendix

A.1 Proposition Graph Construction Details

The PropRAG Proposition graph $G = (V, E)$ is constructed to facilitate reasoning over interconnected propositions. The vertex set V comprises two main types of nodes:

- V_{entity}: Nodes representing entities extracted from the text corpus.
- $V_{passage}$: Nodes representing the original text passages from which propositions and entities were derived.

The edge set E includes the following key types, designed to capture relationships within and between propositions, and to link entities back to their source contexts:

- **Entity Clique Edges (Implicit Proposition Hyper-edge):** For each proposition p extracted from the corpus, which contains a set of entities $\mathcal{E}(p)$, we add undirected edges connecting all pairs of distinct entities $\{e_i, e_j\}$ such that $e_i, e_j \in \mathcal{E}(p)$ and $e_i \neq e_j$. This forms a clique (a fully connected subgraph) among all entities co-occurring within that single proposition. This clique structure implicitly represents the proposition p as a hyper-edge, contextually linking all its constituent entities together, rather than relying on potentially ambiguous predicate-labeled edges between only two entities as in traditional triple stores.
- **Passage Containment Edges:** An undirected edge connects each entity node $e \in V_{entity}$ to the passage node $d \in V_{passage}$ corresponding to the text passage from which entity e (and its associated propositions) were originally extracted. These edges ground entities and propositions in their source documents.
- **Synonymy Edges:** An undirected edge connects two distinct entity nodes $e_i, e_j \in V_{entity}$ if their pre-computed embeddings are highly similar, i.e., $\text{sim}(\text{emb}(e_i), \text{emb}(e_j)) \geq \tau_{syn}$, where τ_{syn} is a predefined similarity threshold. These edges help bridge different textual mentions of the same underlying concept.

This graph structure allows for traversal algorithms (like PPR and beam search) to navigate through the rich context embedded in propositions (via the entity cliques/hyper-edges) and to connect entities back to their original passages, facilitating comprehensive evidence aggregation.

A.2 Implementation Details

PropRAG leverages Llama-3.3-70B-Instruct for offline proposition extraction (and as the final QA reader for experiments) and NV-Embed-v2 (7B) as the base embedding model for passages, entities, and propositions, ensuring consistency with the HippoRAG 2 baseline setup. Default parameters used in PropRAG experiments are as follows:

- Beam width for path discovery (B): 4
- Maximum path length for beam search (L_{max}): 3
- Initial PPR damping factor (Stage 1, exploration): 0.75
- Final PPR damping factor (Stage 2, exploitation): 0.45
- Number of passages in subgraph (K): 50
- Number of top paths for exact scoring (beam search internal re-ranking) (M): 40
- Number of top initial seeds for final PPR ($B_{initial}$): 5
- Number of top propositions to select seeds from for final PPR ($P_{initial}$): B (Beam width)
- Number of top beam-derived seeds for final PPR (B_{beam}): 5
- Number of top beam-derived paths to select seeds from for final PPR (P_{beam}): 5
- Synonymy embedding similarity threshold (τ_{syn}): 0.8
- Number of initial propositions for seeding Stage 1 PPR (N_{prop}): 20
- Number of initial entities from the top-N_{prop} propositions for seeding Stage 1 PPR (N_{entity}): 40
- Weight for passage direct retrieval score in final PPR ($\lambda_{passage}$): 0.05

These parameters were determined based on empirical performance on development sets or adopted from common practices in related research where applicable. The choice of $L_{max} = 3$ was based on achieving the best average F1 score across development datasets.

A.3 LLM Prompts

This section details the prompts used for entity and proposition extraction with Llama-3.3-70B-Instruct, crucial for the offline indexing phase of PropRAG.

A.3.1 Entity Extraction Prompt

This prompt is designed for inclusive entity identification. Unlike strict Named Entity Recognition (NER) often used for triple extraction, this step aims to capture a broader set of concepts relevant for constructing rich propositions. It explicitly asks the LLM to identify named entities, dates, important generic entities, and entities involved in predicate relations. This provides a comprehensive list for the subsequent proposition generation phase, which only uses entities from this pre-identified set. (The prompt is shown in Figure 4)

A.3.2 Proposition Extraction Prompt

This prompt guides the LLM to decompose a passage into atomic, yet contextually complete, propositions. It strictly uses the entities identified in the previous step (Figure 4). The core focus is on maintaining high fidelity by preserving complex relationships, conditions, and the full context, which are often lost or oversimplified in traditional triple extraction processes. (The prompt is shown in Figure 5)

A.4 Proposition Graph Statistics

The proposition graphs constructed for each dataset vary in size and complexity, reflecting the nature of the underlying corpora. Table 5 provides key statistics for the graphs used in our experiments. These include the number of extracted propositions, the number of passage nodes (corresponding to unique passages in the corpus subset), the number of unique entity nodes identified, and the total number of edges in the constructed graph (encompassing entity clique edges, passage containment edges, and synonymy edges).

A.5 Cost and Efficiency

The offline indexing phase of PropRAG involves LLM-based proposition and entity extraction, as well as embedding computation. For embedding, we run a float16 version of NV-Embed-v2 on an NVIDIA RTX 4090 GPU. For proposition and entity extraction, we utilize the Llama-3.3-70B-Instruct model via Nebius AI Studio's API endpoint. Processing each passage for proposition and entity extraction takes approximately 2 seconds with this setup. As a concrete example, indexing the 11,656 passages from the MuSiQue dataset completed within approximately 40 minutes, at a monetary cost of around $4 USD using the API.

The token cost for the offline LLM-based proposition extraction is an important consideration. Table 6 compares the input and output token counts for PropRAG on the MuSiQue dataset against those reported for other structure-augmented RAG methods by Gutiérrez et al. (2025) for their respective offline knowledge structuring phases.

PropRAG's token cost for proposition extraction is higher than methods like HippoRAG 2 (which uses OpenIE for triple extraction, often less LLM-intensive) or RAPTOR (which focuses on summarization). This is attributable to the detailed instructions and the generation of full-sentence propositions, which are richer but require more tokens. However, PropRAG's costs are considerably lower than methods like LightRAG and GraphRAG, which may involve more extensive LLM-based processing for their graph construction or summarization steps. The trade-off is between the upfront offline cost of generating high-fidelity propositions and the downstream benefits in retrieval accuracy and the avoidance of online LLM calls during retrieval. The online retrieval phase of PropRAG, involving PPR and beam search, is entirely LLM-free and computationally efficient, relying on precomputed embeddings and graph operations.

A.6 Entity Score Calculation from Paths

After the beam search identifies a set of high-scoring proposition paths (as detailed in Section 5.3), PropRAG determines the importance of individual entities based on their participation in these paths. This entity scoring is crucial for generating the final set of seed nodes (S_{final}) used in the Stage 2 PPR (Section 5). The scoring process adheres to the following principles:

1. **Path Score Inheritance:** Each proposition within an identified path is considered to have the same relevance score as the overall path it belongs to.

2. **Entity Score Aggregation:** An entity's total score is determined by summing the scores of all propositions (and thus, all paths) in which it appears. If an entity is part of multiple high-scoring paths or multiple propositions within a

Entity Extraction Prompt

Instruction: Your task is to extract entities from the given paragraph. Respond with a JSON dictionary only, with a "entities" key that maps to an non-empty list of entities. All named entities and dates must be included in the list. All generic entities important to the theme of the passage must be included in the list. All entities that is involved in a predicate relation to the above entities must be included in the list. All dates must be included in the list.

Demonstration:

Example Paragraph: Radio City Radio City is India's first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.

Example Output:

```
{"entities":
    ["Radio City", "India", "private FM radio station", "3 July 2001", "Hindi",
     "English", "New Media", "May 2008", "PlanetRadiocity.com", "music portal",
     "news", "videos", "songs"]
}
```

Input Format:

```
Passage: ${passage}
```

Figure 4: LLM prompt for Entity Extraction. This prompt aims for comprehensive entity identification beyond standard NER.

Proposition Extraction Prompt

Instruction:
Your task is to analyze text passages and break them down into precise, atomic propositions using a specified list of named entities. A proposition is a fully contextualized statement that expresses a single unit of meaning with complete specificity about the relationships described.
For each proposition:

1. Extract a complete, standalone statement that preserves the full context
2. Use ONLY the entities provided in the `named_entities` list - do not introduce new entities
3. Ensure each proposition contains only ONE claim or relationship
4. Be extremely specific about which entities are involved in each relationship
5. Maintain clear causal connections between related statements

Respond with a JSON object containing a list of propositions, where each proposition is an object with:
- `"text"`: The full proposition text as a complete, contextualized statement
- `"entities"`: An array of entities from the `named_entities` list that appear in that proposition
Critical Guidelines:

- ONLY use entities from the provided `named_entities` list
- Make relationships explicit and specific - clarify exactly which entities relate to which other entities
- Clarify precisely which entity a modifier applies to (e.g., specify which product had "80% improvement")
- Establish clear connections between related facts (e.g., "Adobe optimized their applications FOR THE M1 CHIP")
- Connect comparative statements to their specific reference points (e.g., "Adobe's applications on the M1 chip improved by 80% compared to Intel-based Macs")
- Preserve temporal information and causal relationships between events
- Make each proposition stand alone with all necessary context
- Include ALL relevant entities from the `named_entities` list in both the proposition text and entities array
- Ensure the collection of propositions captures ALL meaningful information in the passage

---

Demonstration:
Passage: In 2020, after Apple launched the M1 chip, major software companies like Adobe optimized their applications, improving performance by up to 80% compared to Intel-based Macs.
Named entities:

```
["Apple", "M1 chip", "2020", "Adobe", "Adobe's applications", "Intel-based
    ↪ Macs", "80% performance improvement"]
```

```
{
  "propositions": [
    {
      "text": "Apple launched the M1 chip in 2020.",
      "entities": ["Apple", "M1 chip", "2020"]
    },
    {
      "text": "Adobe optimized their applications specifically for the M1 chip
          ↪ after its launch.",
      "entities": ["Adobe", "Adobe's applications", "M1 chip"]
    },
    {
      "text": "Adobe's applications running on the M1 chip improved performance
          ↪ by up to 80% compared to the same applications running on Intel-
          ↪ based Macs.",
      "entities": ["Adobe", "Adobe's applications", "M1 chip", "80% performance
          ↪ improvement", "Intel-based Macs"]
    }
  ]
}
```

Input Format:

```
Passage: ${passage}
Named entities: ${entities_json_list}
```

Figure 5: LLM prompt for Proposition Extraction. This prompt emphasizes contextual completeness and adherence to pre-identified entities.

Table 5: Statistics of Constructed Proposition Graphs per Dataset.

Statistic	NQ	PopQA	MuSiQue	2Wiki	HotpotQA
# Propositions	55536	57624	59028	30099	53566
# Passage Nodes	9633	8676	11656	6119	9811
# Entity Nodes	62368	73577	76928	43444	75608
# Total Edges	1.27M	1.17M	1.34M	0.86M	1.31M

Table 6: Offline LLM Token Costs (Input/Output) for Knowledge Structuring on MuSiQue Dataset (Millions of Tokens). Baseline data from Gutiérrez et al. (2025).

Method	Input Tokens (M)	Output Tokens (M)
RAPTOR	1.7	0.2
HippoRAG 2	9.2	3.0
PropRAG (Ours)	**16.5**	**4.6**
LightRAG	68.5	18.3
GraphRAG	115.5	36.1

single path, its score accumulates, reflecting its centrality and repeated relevance.

3. **Emphasis on Connecting Entities:** The scoring mechanism gives additional weight to entities that form crucial links within a reasoning path, particularly for synonymous connections.
    - **Synonymous Connections Boost:** When a proposition P_A (containing entity E_A) connects to proposition P_B (containing entity E_B) via a synonymous link where $E_A \approx E_B$, the *connected entity* (E_B in P_B) receives an additional score increment equivalent to the path's score. This effectively elevates the importance of E_B, treating it as a strong continuation of a central concept from P_A. The rationale is that E_B is vital for identifying the passage associated with P_B. The original connecting entity (E_A in P_A) contributes its score through its presence in P_A but does not receive this specific connection-based score enhancement itself. If P_A was connected from a preceding proposition, its own central entities would have been accounted for similarly.
    - **Exact Connections:** Entities that are shared exactly between two consecutive propositions in a path (forming an exact connection) naturally contribute to the score aggregation through their appearance in both propositions. Their role as direct bridges is thus inherently emphasized by the summation of scores from both propositions they are part of.
4. **Initial Proposition Entities:** For entities appearing in the very first proposition of a path (which do not have a preceding "connection" within that path), their initial relevance is captured through the "exploration seeds" ($S_{initial}$). Many entities from these initial top query-relevant propositions are directly considered as exploration seeds. This ensures their potential importance is factored into the final seed set, even if they don't benefit from the connection-based score enhancements that apply to entities deeper within a path.

Following the aggregation of scores for all entities involved in the discovered paths, the entities are ranked by their total accumulated scores. This ranked list is then used to select the top-B_{beam} "exploitation seeds." These exploitation seeds, rich in path-derived relevance, are combined with the "exploration seeds" ($S_{initial}$) to form the final seed set S_{final} for the concluding PPR stage, ensuring a comprehensive and robust final ranking of evidence passages.