Reciprocal Rank Fusion in Hybrid GraphRAG

Rajesh Kumar Gupta

06/29/2026

Abstract

Hybrid GraphRAG systems often combine semantic vector search with knowledge graph retrieval. These retrieval systems produce different types of relevance signals, and their raw scores are not directly comparable. Reciprocal Rank Fusion (RRF) provides a simple and robust way to combine their ranked outputs. Instead of merging raw scores, RRF combines rank positions and rewards evidence that is ranked highly by multiple independent retrievers.

Contents

1 Introduction

Retrieval-Augmented Generation (RAG) improves Large Language Model responses by grounding answers in retrieved evidence. In a Hybrid GraphRAG architecture, evidence may come from multiple retrieval systems.

Two common retrieval approaches are:

- **Vector Search**: retrieves text that is semantically similar to the user query.
- **Knowledge Graph Retrieval**: retrieves entities, relationships, facts, and evidence paths that are structurally relevant.

Both methods are useful, but they behave differently. Vector search is strong at finding related language, while graph retrieval is strong at finding explicit relationships and domain structure.

The challenge is that each retriever produces its own ranking and its own scoring mechanism. A vector similarity score and a graph relevance score do not live on the same mathematical scale. Therefore, directly averaging or comparing these scores can produce unreliable results.

Reciprocal Rank Fusion, or RRF, solves this problem by combining rankings rather than raw scores.

2 Why Hybrid Retrieval Is Needed

A single retrieval technique is often insufficient for complex enterprise questions.

For example, consider the question:

> How do PURE and Chubb differ in coverage for damage to a covered auto?

This question requires both semantic and structured understanding.

Vector search helps retrieve policy language that is semantically close to the question. It can find relevant paragraphs even when the wording differs from the user query.

Knowledge graph retrieval helps identify structured concepts such as:

- carrier names,
- coverage concepts,
- exclusions,
- limits,
- definitions,
- relationships between policies and evidence.

If only vector search is used, the system may retrieve similar text but miss important relationships. If only graph retrieval is used, the system may retrieve structured facts but miss surrounding policy language. Hybrid retrieval allows both systems to contribute evidence.

3 Why Rankings Are Better Than Raw Scores

Redis vector search may return scores such as:

$$0.93, \quad 0.91, \quad 0.88$$

A graph retriever may return a path confidence, traversal rank, SPARQL order, or another structural relevance signal such as:

$$0.61, \quad 1, \quad 0.004$$

These numbers are not directly comparable. A Redis similarity score of 0.91 does not necessarily mean the result is better than a graph result with score 0.61. They are produced by different retrieval models.

A score-averaging approach would require difficult normalization decisions:

- How should vector similarity be compared to graph traversal rank?
- Should graph scores and vector scores be weighted differently?
- Should the weighting change for definitions, comparisons, exclusions, or policy limits?
- How should noisy results from one retriever be handled?

RRF avoids these issues by ignoring raw scores and using only rank positions.

4 Hybrid GraphRAG Pipeline

A simplified Hybrid GraphRAG pipeline is shown below:

$$\text{User Query} \rightarrow \text{Hybrid Retrieval} \rightarrow \text{Redis Vector Search}$$

$$\text{User Query} \rightarrow \text{Hybrid Retrieval} \rightarrow \text{Knowledge Graph Retrieval}$$

The two ranked lists are then passed to the fusion layer:

$$\text{Redis Ranked List} + \text{Graph Ranked List} \rightarrow \text{Reciprocal Rank Fusion}$$

The output is a single unified evidence ranking:

$$\text{Unified Ranked Evidence} \rightarrow \text{LLM Context} \rightarrow \text{Grounded Answer}$$

Conceptually, the flow is:

$$\begin{array}{c} \text{User Query} \\ \downarrow \\ \text{Redis Vector Search and Knowledge Graph Retrieval} \\ \downarrow \\ \text{Independent Ranked Evidence Lists} \\ \downarrow \\ \text{Reciprocal Rank Fusion} \\ \downarrow \\ \text{Unified Ranked Evidence} \\ \downarrow \\ \text{LLM Answer Generation} \end{array}$$

5 Reciprocal Rank Fusion

Reciprocal Rank Fusion is a rank aggregation method. It combines the outputs of multiple retrieval systems into a single ranked list.

For an evidence item d, the RRF score is defined as:

$$\mathrm{RRF}(d) = \sum_{i=1}^{m} \frac{1}{k + \mathrm{rank}_i(d)} \tag{1}$$

where:

- d is the evidence item.
- m is the number of retrieval systems.
- $\mathrm{rank}_i(d)$ is the rank of item d in retrieval system i.
- k is a smoothing constant.

A commonly used value is:

$$k = 60$$

The smoothing constant prevents the top-ranked item from dominating too aggressively and keeps the contribution of lower-ranked items stable.

6 Mathematical Intuition

Each retriever contributes a reciprocal value based on where the evidence item appears in its ranked list.

If an item is ranked first, its contribution is:

$$\frac{1}{60+1} = \frac{1}{61} = 0.01639$$

If an item is ranked second, its contribution is:

$$\frac{1}{60+2} = \frac{1}{62} = 0.01613$$

If an item is ranked tenth, its contribution is:

$$\frac{1}{60+10} = \frac{1}{70} = 0.01429$$

The contribution decreases as the rank becomes lower. More importantly, an evidence item appearing in multiple ranked lists accumulates multiple contributions. This is how RRF rewards agreement between independent retrievers.

7 Worked Example

Assume Redis Vector Search and Neptune Graph Retrieval return the following evidence items.

Redis Rank	Evidence Item
1	Evidence A
2	Evidence B
3	Evidence C
4	Evidence D

Table 1: Redis ranked evidence list

Graph Rank	Evidence Item
1	Evidence B
2	Evidence C
3	Evidence E
4	Evidence F

Table 2: Graph ranked evidence list

7.1 Evidence B

Evidence B appears in both retrieval systems.

$$\text{Redis Rank} = 2$$

$$\text{Graph Rank} = 1$$

Therefore:

$$\begin{aligned} \text{RRF(Evidence B)} &= \frac{1}{60+2} + \frac{1}{60+1} && (2)\\ &= \frac{1}{62} + \frac{1}{61} && (3)\\ &= 0.01613 + 0.01639 && (4)\\ &= 0.03252 && (5) \end{aligned}$$

7.2 Evidence A

Evidence A appears only in Redis.

$$\text{Redis Rank} = 1$$

Therefore:

$$\begin{aligned} \text{RRF(Evidence A)} &= \frac{1}{60+1} && (6)\\ &= \frac{1}{61} && (7)\\ &= 0.01639 && (8) \end{aligned}$$

7.3 Evidence C

Evidence C appears in both retrieval systems.

$$\text{Redis Rank} = 3$$

$$\text{Graph Rank} = 2$$

Therefore:

$$\begin{aligned} \text{RRF(Evidence C)} &= \frac{1}{60+3} + \frac{1}{60+2} && (9)\\ &= \frac{1}{63} + \frac{1}{62} && (10)\\ &= 0.01587 + 0.01613 && (11)\\ &= 0.03200 && (12) \end{aligned}$$

8 Final Fused Ranking

The final fused ranking is based on the total RRF score.

Evidence	**Redis Rank**	**Graph Rank**	**RRF Score**
Evidence B	2	1	0.03252
Evidence C	3	2	0.03200
Evidence A	1	–	0.01639
Evidence D	4	–	0.01562

Table 3: Final RRF fused ranking

Evidence A was the top Redis result, but Evidence B receives the highest final score because it is ranked highly by both Redis and Graph Retrieval.

This is the core principle of RRF:

Evidence supported by multiple independent retrievers should rise.

9 Deduplication and Evidence Fusion

In Hybrid GraphRAG, the same source evidence may be retrieved by both Redis and the knowledge graph.

For example, Redis may retrieve a policy chunk because it is semantically similar to the user query. The graph retriever may retrieve the same chunk because it is attached to a relevant coverage concept or evidence path.

Without deduplication, the final context may contain repeated evidence. Therefore, the fusion layer should merge overlapping evidence using a stable canonical key.

A canonical key may be based on:

- document identifier,
- page number,
- chunk identifier,
- source element identifier,
- snippet hash,

- bounding box metadata.

After deduplication, the fused item can preserve metadata from both retrieval systems. If an item is found by both Redis and graph retrieval, it can be labeled as hybrid evidence.

10 How RRF Supports LLM Grounding

The purpose of RRF is not only to rank evidence. Its larger role is to improve the quality of the context passed to the LLM.

A strong evidence ranking helps the LLM:

- focus on the most relevant policy language,
- use structurally grounded facts,
- avoid unsupported claims,
- reduce hallucinated citations,
- generate answers backed by traceable evidence.

Instead of sending two disconnected result sets to the LLM, the system sends one unified evidence digest.

$$\text{Redis Results} + \text{Graph Results} \rightarrow \text{RRF} \rightarrow \text{Unified Evidence Context}$$

This keeps the prompt cleaner and makes the final answer more reliable.

11 Why RRF Works

RRF works well in Hybrid GraphRAG because it has three important properties.

11.1 It Avoids Score Normalization

Redis and graph retrieval scores are not on the same scale. RRF does not require them to be normalized, compared, or averaged.

11.2 It Rewards Agreement

If multiple retrievers rank the same evidence highly, that evidence receives multiple reciprocal contributions and moves upward in the final ranking.

11.3 It Reduces Noise

A noisy result from one retriever usually appears in only one list. Since it does not receive support from another retriever, it is less likely to dominate the final ranking.

12 Benefits of RRF in Hybrid GraphRAG

The major benefits of Reciprocal Rank Fusion are:

- **Simple**: The algorithm is easy to implement and explain.
- **Robust**: It works even when retrieval scores are incompatible.

- **Explainable**: The final ranking can be traced back to source ranks.
- **Retriever-agnostic**: It can combine vector search, graph retrieval, keyword search, or structured query results.
- **Grounding-friendly**: It prioritizes evidence that has support from multiple retrieval paths.
- **Production-friendly**: It avoids fragile score tuning across retrieval systems.

13 Practical Takeaway

RRF is not just a ranking trick. In a Hybrid GraphRAG system, it acts as the bridge between semantic retrieval and structured graph retrieval.

Vector search answers the question:

Which text is semantically similar to the query?

Graph retrieval answers the question:

Which entities, relationships, and evidence paths are structurally relevant?

RRF answers the final fusion question:

Which evidence is consistently important across retrievers?

This makes it especially useful for domains where accuracy, traceability, and explainability matter.

14 Conclusion

Reciprocal Rank Fusion provides a simple and effective way to combine heterogeneous retrieval systems in Hybrid GraphRAG. Instead of comparing incompatible raw scores, RRF aggregates rank positions.

The result is a unified evidence ranking that rewards agreement between independent retrievers. This improves the quality of the context provided to the LLM, strengthens grounding, reduces hallucination risk, and produces more trustworthy responses.

For enterprise and regulated document workflows, this kind of evidence fusion is essential. It helps ensure that final answers are not only relevant, but also traceable to reliable source evidence.