iG^{2}RAG: Information Gain Graph-based Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Existing Retrieval-Augmented Generation (RAG) methods predominantly retrieve documents by relying on surface-level similarity to the query or by aligning entities and relations to the graph. However, in multi-hop scenarios, such strategies frequently yield insufficient or redundant retrieval, failing to account for document complementarity and potential information gain, which are critical for effectively reducing query uncertainty. To address this issue, we introduce iG^{2}RAG, which constructs a novel **i**nformation **G**ain **G**raph as the foundation of the retrieval process. In the offline phase, we treat documents as nodes, mine similar neighbors to form subgraphs, and use a Large Language Model (LLM) to evaluate these neighbors, creating an information gain graph. During the online query phase, seed nodes are identified based on the query, and the Personalized PageRank (PPR) algorithm is applied to iteratively retrieve the optimal set of documents with high information gain. This process simulates foraging behavior, where the information gain graph acts as a map and PPR mimics the search for better food. Our experiments show that iG^{2}RAG outperforms baselines on multi-hop datasets, achieving state-of-the-art results and validating the framework's effectiveness.

1 Introduction

RAG has emerged as a critical approach to mitigating the hallucination problem in large language models (LLMs) (Lewis et al., 2020; OpenAI et al., 2024). Its central idea is to enhance the generation process by retrieving relevant evidence documents from external knowledge bases (Mei et al., 2025; Singh et al.). Although initially introduced several years ago, RAG remains indispensable in many LLM applications, as it improves performance even under constrained model capacity (Huang et al.). Notably, the evolution of RAG has shifted from relying on retrievers to fetch high similar documents

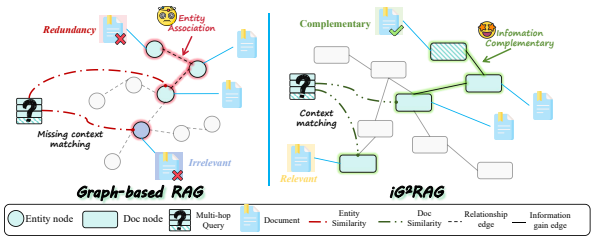

Figure 1: Graph-based RAG relies on query–entity triple matching, which is insufficient as many implicit relations cannot be directly linked. In contrast, iG^{2}RAG matches queries with documents and leverages information gain edges to capture complementary evidence.

toward the graph-based RAG paradigm, which leverages structured knowledge bases as a core feature, thereby demonstrating stringent requirements for the quality of retrieved documents (Gao et al., 2023; Peng et al., 2024).

However, when addressing multi-hop questions, both similarity-based RAG and graph-based RAG baselines exhibit inherent limitations, as such queries typically require diverse and complementary evidence to support accurate reasoning (Zhang et al., 2025a; Saleh et al., 2024). Similarity-based RAG tends to retrieve a set of documents with high surface-level semantic similarity to the query. While these documents are indeed relevant, they often contain substantial redundancy and contribute little new insight for subsequent generation (Zhuang et al., 2024). In contrast, graph-based RAG constructs nodes by extracting entities and relations from documents using an LLM and retrieves documents through structured element matching (Guo et al., 2024; Edge et al., 2024; Jimenez Gutierrez et al., 2024). Although this approach can partially reveal latent entity relations to support multi-hop reasoning, the organization of nodes neglects the contextual information of the original corpus (Xu et al., 2025). Consequently, prioritizing entity-level relational similarity can introduce approxi-

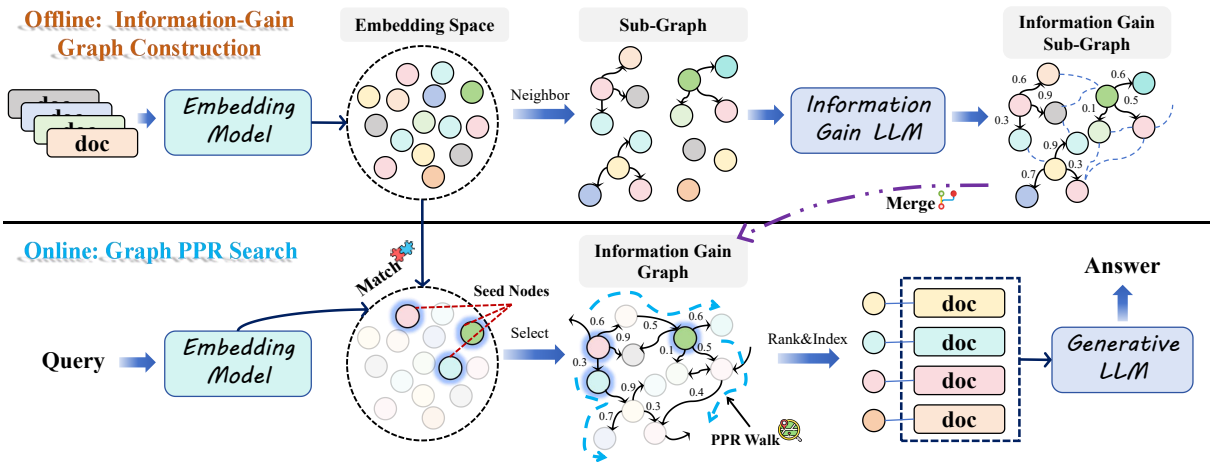

Figure 2: The framework of iG^{2}RAG. In the offline phase, we construct an information gain graph using document chunks as nodes and information gain as edge weights. In the online phase, we perform a Personalized PageRank (PPR) walk on the information gain graph, starting from seed nodes that are most similar to the query, to retrieve documents that maximize informational gain for the query.

mate but spurious links or semantically similar yet nonequivalent entities, leading to redundant or even misleading retrieval and weakening the focus on cross-document informational complementarity, as illustrated in Figure 1. Whether similarity-based or graph-based, the retrieved document sets may not provide sufficient support for the final answer. This reveals a core limitation of existing RAG approaches, namely their lack of consideration for document complementarity during retrieval (Zhang et al., 2002). In essence, these methods overlook whether retrieved documents provide complementary knowledge and effective information gain, thereby failing to reduce query uncertainty.

Consequently, we proposed an **i**nformational **G**ain **G**raph-based RAG framework, termed iG^{2}RAG, which aims to minimize query uncertainty through an optimized retrieval process and thereby ensure improved generation performance, as shown in Fig. 2. Specifically, we construct the corpus as a graph, where iG^{2}RAG differs from prior approaches by defining document chunks, rather than entities, as nodes. At the same time, instead of adopting traditional relations as edges, we determine edge weights based on the LLM's evaluation of informational gain between chunks, resulting in the construction of an informational gain graph. In this graph, each chunk computes its informational gain with its n nearest neighbors to assign the corresponding edge weights. On this basis, the key challenge lies in how to perform effective retrieval over the informational gain graph. Inspired by animal foraging theory, we adopt the classical random walk algorithm, namely Personalized PageRank (PPR) (Page et al., 1999), to efficiently traversal the graph structure. iG^{2}RAG first selects h chunks with the highest similarity to the query as seed nodes and then executes PPR on the informational gain graph until convergence. This process parallels the foraging behavior of animals, where they follow an "information scent" to explore promising paths in search of the most valuable food clusters (Pirolli and Card, 1999). In our framework, informational gain serves as the "information scent", while the PPR strategy guides the model to explore and aggregate the most valuable information within the graph. In summary, the main contributions of this paper are as follows:

- We propose iG^{2}RAG, a novel RAG framework grounded in an information gain graph, offering a fresh perspective for advancing research in next-generation RAG.
- We formulate an information-gain-driven graph representation that treats document chunks as nodes and weights edges by their potential to reduce query uncertainty, and we integrate PPR to identify complementary document sets with maximal informational value.
- We present comprehensive experiments on public multi-hop query-answer benchmarks, showing that iG^{2}RAG achieves substantial improvements over strong baselines, highlighting its advancement and effectiveness.

2 Related Work

RAG. Although RAG was first proposed in 2020 (Lewis et al., 2020), its value in LLM applications remains significant, even for models equipped with long context windows (Chang et al., 2024; Dong et al., 2025). Early research in this area primarily focused on more efficient methods for database chunking, such as MetaChunking (Zhao et al., 2024) and LateChunking (Günther et al., 2024). However, with the advancement of LLMs, rule-based data organization has gradually been superseded by the models' intrinsic capabilities for data comprehension and structuring. Concurrently, powerful dense retrievers like Qwen3-Embedding-8B (Zhang et al., 2025b) and NV-Embed-v2 (Lee et al., 2024), which are comparable in scale to LLMs, have significantly improved the efficiency of vectorization and matching. This has allowed traditional RAG pipelines to continue achieving excellent retrieval performance and enhance answer quality. Nevertheless, in multi-hop question scenarios, a sole reliance on surface-level similarity matching struggles to capture the latent connections between a query and the knowledge base, often leading to the retrieval of repetitive or redundant information.

Structured-Augmented RAG. To address multi-hop questions, structured-enhanced RAG improves retrieval by extracting entities or relations from the corpus to build knowledge graphs (Gao et al., 2025). Representative works include RAPTOR (Sarthi et al., 2024), which strengthens cross-corpus links via recursive tree-structured retrieval; GraphRAG (Edge et al., 2024) and its lightweight variant LightRAG (Guo et al., 2024), which employ LLMs for graph construction; and HippoRAG2 (Gutiérrez et al., 2025), which extends entity relations with phrase-based links and document nodes for a more refined pipeline. Yet these methods often rely on entity or phrase matching, where associations to the source text may be weak, leading retrieval to favor relationally similar but topically divergent documents, thereby introducing redundancy.

Such redundancy stems from overlooking document complementarity. An effective retrieval set should preserve thematic consistency while contributing novel information to reduce query uncertainty. This paper proposes a graph-based framework that improves question answering by emphasizing complementarity and information gain in retrieval.

3 Method

Ovreview. In this section, we will introduce the construction process of iG^{2}RAG. Section 3.1 will describe the offline construction of the information gain graph, Section 3.2 will introduce how to traverse the information gain graph using Personalized PageRank, and Section 3.3 will explain the advantages of incremental graph updating.

3.1 Offline: Information-Gain Graph Construction

Nodes. Both Naive RAG and graph-based RAG require tailored processing of the database. For iG^{2}RAG, the primary objective during the offline knowledge base construction phase is to create an information gain graph. However, in the absence of prior knowledge about user queries, it is infeasible to predetermine the information gain for specific queries, making it challenging to directly reduce uncertainty. Instead, during the offline phase, only document chunks are available from the knowledge base. Consequently, iG^{2}RAG constructs the information gain graph using these chunks as nodes ($v \in \mathcal{V}$). Intuitively, in the online phase, queries are matched against these chunks, implying that chunks indirectly represent the distribution of future queries.

Edges. After determining the nodes, we employ a large language model ($\mathcal{M}_i$) with semantic understanding and information gain evaluation capabilities to compute the information gain between nodes and assign edge weights ($\mathcal{E}$) accordingly. The choice of an LLM, rather than traditional information gain calculation methods, stems from the fact that documents in the knowledge base are typically not presented in the form of probability distributions, rendering classical formulas both impractical and lacking semantic guidance.

Information Gain. Under the setting where chunks serve as nodes and information gain as edge ($\mathcal{E}$) weights, a critical challenge arises: computing information gain for all pairs of chunks incurs prohibitively high computational costs and an inefficient construction process. To address this, we first operate in the embedding representation space ($\mathbf{E} \in \mathbb{R}^{N \times d}$) of the document to acquire the top-n representations in the neighborhood of each representation, forming a subgraph for it. Formally, let the node set be $\mathcal{V}$ and the similarity edges be $\mathcal{E}_{sim}$. The similarity subgraphs can be expressed

Algorithm 1 Information-Gain Graph Construction

Require: LLM $\mathcal{M}_i$, neighbor size n, Document chunks node $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$
Ensure: Information gain graph $\mathcal{G}_{info}(\mathcal{V}, \mathcal{E}_{info})$

```
Initialize E_info ← ∅
Compute embeddings: E ∈ R^{N×d}
for each chunk v_i ∈ V do
    Find top-n similar chunks Neighbors(v_i)
    Construct subgraph {Eq. (1)}
    Remove edge weights to get unweighted subgraph G^u_sub
    for each v_j ∈ Neighbors(v_i) do
        w_ij ← M_i(v_i, v_j) {Eq. (2)}
        E_info ← E_info ∪ {(v_i, v_j, w_ij)}
    end for
end for
Concatenate all subgraphs to form G_info(V, E_info) ∈ R^{N×n}
return G_info(V, E_info)
```

Algorithm 2 Graph PPR Search

Require: Query q, Information gain graph $\mathcal{G}_{info}(\mathcal{V}, \mathcal{E}_{info})$, seed nodes h, top-k documents, damping factor α, threshold ϵ
Ensure: Retrieved documents $\mathcal{D}$

```
Compute query embedding Embed_q
Compute similarity vector {Eq. (3)}
Construct personalization vector {Eq. (4)}
Normalize P to ensure ||P||_1 = 1
Initialize r^(0) ← P, t ← 0
Construct transition matrix T ← G_info
repeat
    r^(t+1) ← (1 − α) · T · r^(t) + α · P {Eq. (5)}
    t ← t + 1
until ||r^(t) − r^(t−1)||_1 < ε {Eq.(6)}
R ← r^(t) {Converged rank vector}
D ← Index(Sort(R, k)) {Eq. (7)}
return D
```

as a collection represented in:

$$\mathcal{G}(v, Neighbor(\mathcal{E}_{sim})) \in \mathbb{R}^{1\times n}, \tag{1}$$

where $Neighbor(*)$ denotes the set of neighboring nodes, obtained in this work using a simple top-n strategy, and $\mathcal{E}_{sim}$ represents the normalized similarity between pairs of nodes. Subsequently, we remove the weight value from the edges to obtain unweighted subgraphs ${\mathcal{G}^u}_{sub}$, which is essentially an adjacency matrix of size $1 \times n$. We then prompt the $\mathcal{M}_i$ to assign weights to the nodes and edges of ${\mathcal{G}^u}_{sub}$ by evaluating the information gain of the edges in these subgraphs. Specifically, we instruct the $\mathcal{M}_i$ to evaluate the information gain of edges across four levels:

$$w_{ij} = \mathcal{M}_i(e) = \begin{cases} [0.0, 0.2], & \text{Redundant} \\ [0.3, 0.5], & \text{Minor} \\ [0.6, 0.8], & \text{Significant} \\ [0.9, 1.0], & \text{Novel} \end{cases} \tag{2}$$

where e represents an edge between two nodes in ${\mathcal{G}^u}_{sub}$. Ultimately, we concatenate all nodes to form a large graph structure $\mathcal{G}_{info}(\mathcal{V}, \mathcal{E}_{info}) \in \mathbb{R}^{N\times n}$, where all edges are weighted by their information gain.

3.2 Online: Graph PPR Search

Based on the obtained $\mathcal{G}_{info}(\mathcal{V}, \mathcal{E}_{info})$, this work employs the classic Personalized PageRank (PPR) random walk algorithm to integrate knowledge via information gain. This differs from prior approaches which matches the query with entity nodes before performing the walk. In contrast, our method matches the query with chunk nodes to anchor the top-h relevant nodes and subsequently expands via random walks. Since information gain is established between chunks, directly matching the query enables the implicit transfer of information gain to the query, thereby reducing its uncertainty.

Personalized Seed Nodes. The PPR process involves random walks starting from a set of personalized seed nodes. We first identify these nodes by computing a similarity vector $S \in \mathbb{R}^{N\times 1}$, which scores each node's relevance to the query. This is calculated by taking the dot product of the query's embedding (Embed_q) and the node embeddings.

$$S = \text{Embed}_q \cdot \mathbf{E} \in \mathbb{R}^{N\times 1}. \tag{3}$$

Subsequently, the personalization vector $P \in \mathbb{R}^{N\times 1}$ is constructed by selecting the top-h similarity scores from S and nullifying the rest. This operation can be expressed as:

$$P_i = S \cdot \mathbb{I}(i \in \text{Sort}(S, h)) \in \mathbb{R}^{N\times 1}, \tag{4}$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $\text{Sort}(S, h)$ returns the set of indices for the top-h values in S. The resulting vector P is then normalized.

PPR Walk. Based on the obtained vector P, we employ it simultaneously as both the initial vector

and the restart vector for the PPR walk. PageRank is fundamentally a Markov chain random walk model, and PPR extends this by incorporating the personalized vector P. In this Markov chain, the transition matrix ($\mathcal{T}$) is denoted as $\mathcal{G}_{info}(\mathcal{V}, \mathcal{E}_{info})$. The specific walk (transition) process can be expressed by the following equation:

$$r^{(t+1)} = (1-\alpha) \cdot \mathcal{T} \cdot r^{(t)} + \alpha \cdot P, \quad (5)$$

where $r^{(t)}$ denotes the rank vector after the t-th iteration, which we refer to as $\mathbf{R} \in \mathbb{R}^{N \times 1}$, representing the probability distribution over node selections. The parameter α signifies the probability of restarting the walk using P as the restart vector. During the iterative process, we define the following convergence criterion:

$$\|r^{(t+1)} - r^{(t)}\|_1 < \epsilon, \quad (6)$$

where ϵ is a small threshold, typically set to 1×10^{-6}. Finally, we obtain the converged vector $\mathbf{R}$, which contains the selection probability for each node. These probabilities also represent the likelihood of integrating the respective node within the information gain graph. We sort $\mathbf{R}$, select the top-k nodes, and index their corresponding chunk knowledge, with the specific process outlined as follows:

$$\mathcal{D} = \text{Index}\left(\text{Sort}(\mathbf{R}, k)\right) \subseteq \{d_1, \ldots, d_k\}, \quad (7)$$

$$\mathcal{A} = \mathcal{M}_g\left(\mathcal{D} || \mathbf{q}\right), \quad (8)$$

here, Index$(*)$ indicates indexing the original document, and $\mathbf{q}$ represents the query, which is then input into $\mathcal{M}_g$ to generate the final response. By performing a PPR walk on the information gain graph, we obtain the combination of documents that maximizes information gain for the query. This process mirrors the foraging behavior of animals in nature: starting from an initial food source, they follow information cues of food (i.e., information gain) to progressively maximize their gains.

3.3 Incremental update

Unlike conventional chunk-based knowledge bases, graph-structured knowledge bases, such as those used in LightRAG and GraphRAG, require additional consideration of node duplication and interconnections during updates. In contrast, the graph constructed by iG^{2}RAG offers streamlined incremental update capabilities. Since the iG^{2}RAG knowledge graph uses document chunks as nodes and builds a graph with information gain-weighted edges based on neighboring documents, incremental updates can be efficiently performed. This involves matching newly added documents to their neighboring nodes through similarity matching and utilizing $\mathcal{M}_i$ for information gain analysis and evaluation. Similarity matching is performed via dot-product computation, allowing node updates without disrupting the database, as nodes are represented as chunks for lightweight incremental updates, shown in Appendix D, Algorithm 3.

Table 1: Statistics of datasets used in experiments

	MuSiQue	**2Wiki**	**HotpotQA**
Num. of queries	1000	1000	1000
Num. of documents	11656	6119	9811

4 Experimental Setup

Datasets. We evaluated our method on Musique, HotpotQA, and 2WikiMultiHopQA (2Wiki), which are curated versions provided by HippoRAG2. Each dataset includes supporting documents, distractors, and ground-truth answers to enable comprehensive evaluation of retrieval and question answering as shown in Table 1. According to HippoRAG(Jimenez Gutierrez et al., 2024), Musique presents greater reasoning challenges, whereas HotpotQA poses relatively lower demands.

Baselines. Our baseline experimental design encompasses three categories of comparison models. **Only LLM.** This category includes models without any external knowledge base, such as Qwen2.5-72B-Instruct and Llama3.3-70B-Instruct. **Naive RAG**. This category comprises models with dense retrieval capabilities, including BGE-M3 (Chen et al., 2024), text-embedding-v3[1], and Qwen3-Embedding-8B (Zhang et al., 2025b). **Structured-augmented RAG**. This category consists of methods that incorporate structured knowledge or graph augmentation, including **LightRAG** (Guo et al., 2024), **HippoRAG2** (Gutiérrez et al., 2025), and **RAPTOR** (Sarthi et al., 2024). More detailed implementations are listed in Appendix B.

Metrics. We evaluate each method using standard RAG metrics, including Exact Match (EM), F1, and Recall@10. EM and F1 reflect QA performance, while Recall@10 measures retrieval coverage. For multi-hop questions, however, higher

[1]Closed-source embedding model from https://bailian.console.aliyun.com/

Recall does not necessarily improve QA performance, as often only one among the retrieved relevant documents serves as the actual source of the answer.

5 Main Results

In this section, we evaluate iG^{2}RAG against alternative methods in terms of QA quality (EM and F1) and retrieval quality (Recall@10), as shown in Table 2 and Table 3. iG^{2}RAG consistently achieves the highest EM and F1 scores with both Qwen2.5-72B-Instruct and Llama-3.3-70B-Instruct, indicating that its retrieved documents provide stronger support for query answering. Among dense retrievers, Qwen3-Embedding-8B performs competitively with structured methods and even surpasses them on Musique, reflecting the advantage of preserving full semantic similarity. Yet, its reliance on surface matching limits its ability to capture complementary information. In contrast, iG^{2}RAG combines an information gain graph with PPR to assemble a complementary document set, yielding superior performance. For example, with Qwen2.5-72B-Instruct, iG^{2}RAG outperforms the next-best structured method by 1.88%, 3.4%, and 1.93% in F1 score on Musique, 2Wiki, and HotpotQA, respectively, underscoring its strength in retrieving documents with genuinely high informational gain rather than superficial entity links.

For retrieval quality, iG^{2}RAG is slightly outperformed by HippoRAG2 on Musique, but its recall is more critical, retrieving not just relevant documents but those that provide complementary evidence, thereby optimizing downstream QA. By contrast, HippoRAG2 and other baselines emphasize relevance alone, neglecting complementarity. As dataset difficulty decreases, iG^{2}RAG further improves recall while sustaining its QA advantages.

Overall, the superior performance of iG^{2}RAG stems from the integration of the information gain graph with PPR, which enables richer feedback of essential information to the LLM. This mechanism yields substantial information gain and significantly reduces semantic uncertainty in query answering.

6 Discussions

In this section, we will discuss in detail the ablation studies for iG^{2}RAG, experiments on node reranking, hyperparameter selection, and a qualitative analysis.

6.1 Ablation Study

We performed ablation studies on iG^{2}RAG with four variants: Naive RAG, similarity matrix graph with PPR walks (Sim w/ PPR), information gain graph with a greedy strategy (iG² w/ GS), and information gain graph with PPR (iG^{2}RAG). As shown in Table 4, the combination of the information gain graph and PPR walks achieves the best QA performance. PPR on the similarity graph provides limited improvement, as its lack of explicit directionality leads to convergence on redundant information. The iG² w/ GS variant shows that the information gain graph significantly boosts performance. Further analysis reveals that PPR outperforms GS by iteratively identifying optimal gain combinations. Overall, iG^{2}RAG outperforms Naive RAG across all datasets, with F1 score improvements of 2.45%, 6.85%, and 2.96% on Musique, 2Wiki, and HotpotQA, respectively, demonstrating the effectiveness of the information gain graph.

6.2 Number of gain edges

To evaluate the impact of graph connectivity in iG^{2}RAG, we conducted a sensitivity analysis on the top-n hyperparameter, which determines the number of edges linked to each node. Specifically, we explored this effect on the most challenging *Musique* dataset, comparing Qwen2.5-72B-Instruct and Llama3.3-70B-Instruct under top-n values of 3, 5, and 7 (Table 5). Results indicate that both models exhibit comparable performance across settings, with top-n = 5 yielding slightly more balanced outcomes. However, a clear trade-off exists, as lowering the top-n risks omitting extended reasoning paths, while increasing the top-n may introduce redundant edges without improving retrieval quality. Therefore, in this work we set top-n = 5.

6.3 Number of Seed Nodes

We also investigated the impact of using different numbers of seed nodes on *Musique* dataset. As shown in Table 6, using 5 seed nodes yields the optimal QA performance when retrieving 10 relevant documents, although its Recall@10 is lower than that achieved with 7 seed nodes. This indicates that an excessive number of seed nodes can easily fall into the trap of surface similarity, whereas leaving adequate room for the PPR algorithm to explore allows for better discovery of the optimal combination of information gain. Furthermore, using too few seed nodes fails to provide effective initial con-

Table 2: Performance of Qwen2.5-72B-Instruct on EM, F1 and Recall@10 across three Multi-Hop Datasets. "Dense Retriever" denotes a Naive RAG baseline, and "Structure-Augmented RAG" represents more advanced structured-enhanced RAG methods. In the table, **bold** values indicate the best performance, and underlined values indicate the second-best performance.

Methods/Metrics	Musique			2Wiki			HotpotQA		
	EM	F1	Recall@10	EM	F1	Recall@10	EM	F1	Recall@10
Qwen2.5-72B-Instruct	0.0510	0.1357	-	0.2650	0.3081	-	0.2560	0.3464	-
Dense retriever									
BGE-M3 (Chen et al., 2024)	0.2380	0.3411	0.6189	0.4420	0.5077	0.7465	0.5290	0.6619	0.9035
text-embedding-v3	0.2550	0.3630	0.6366	0.4290	0.4967	0.7382	0.5090	0.6415	0.8790
Qwen3-Embedding-8B (Zhang et al., 2025b)	0.2720	0.3891	0.7035	0.4510	0.5163	0.7563	0.5390	0.6708	0.9385
Structure-Augmented RAG									
LightRAG (Guo et al., 2024)	0.1572	0.2503	0.3316	0.3869	0.4855	0.7503	0.4230	0.5741	0.9639
RAPTOR (Sarthi et al., 2024)	0.2660	0.3864	0.6868	0.4670	0.5366	0.7442	0.5440	0.6740	0.9185
HippoRAG2 (Gutiérrez et al., 2025)	0.2640	0.3841	**0.7169**	0.5040	0.5807	0.8405	0.5433	0.6723	**0.9545**
iG^{2}RAG (Ours)	**0.3000**	**0.4079**	0.7057	**0.5310**	**0.6147**	**0.9062**	**0.5630**	**0.6933**	0.9520

Table 3: Performance of Qwen2.5-72B-Instruct on EM, F1 and Recall@10 across three Multi-Hop Datasets.

Methods/Metrics	Musique			2Wiki			HotpotQA		
	EM	F1	Recall@10	EM	F1	Recall@10	EM	F1	Recall@10
Llama-3.3-70B-Instruct	0.0690	0.1667	-	0.292	0.3534	-	0.297	0.4063	-
Dense retriever									
BGE-M3 (Chen et al., 2024)	0.2620	0.3856	0.6892	0.4500	0.5215	0.7523	0.5300	0.6621	0.9385
text-embedding-v3	0.2650	0.3958	0.6406	0.5520	0.6312	0.8548	0.5330	0.6752	0.9215
Qwen3-Embedding-8B (Zhang et al., 2025b)	0.2703	0.4040	0.6976	0.5600	0.6453	0.8495	0.5490	0.6924	0.9385
Structure-Augmented RAG									
LightRAG (Guo et al., 2024)	0.1446	0.2318	0.2848	0.3410	0.4037	0.5565	0.3303	0.4234	0.3750
RAPTOR (Sarthi et al., 2024)	0.2770	0.4135	0.6918	0.4670	0.5366	0.7445	0.5480	0.6921	0.9175
HippoRAG2 (Gutiérrez et al., 2025)	0.2680	0.3984	**0.7198**	0.5610	0.6465	0.8505	0.5480	0.6936	**0.9585**
iG^{2}RAG (Ours)	**0.2940**	**0.4328**	0.7069	**0.6080**	**0.6891**	**0.9095**	**0.5640**	**0.7144**	0.9545

Table 4: QA Performance for Ablation Study. GS performs a 1-hop search based on the neighboring nodes.

Methods	Musique		2Wiki		HotpotQA	
	EM	F1	EM	F1	EM	F1
Naive	0.2720	0.3891	0.4510	0.5163	0.5390	0.6708
Sim w/ PPR	0.2630	0.3834	0.4740	0.5462	0.537	0.6640
iG^2 w/ GS	0.2720	0.3864	0.5210	0.5998	0.5530	0.6852
iG^2 w/ PPR (iG^{2}RAG)	**0.3000**	**0.4079**	**0.5310**	**0.6147**	**0.5640**	**0.6933**

Table 5: QA Performance on Varying top-n.

Model	top-n	EM	F1	Recall@10
Qwen2.5-72B-Instruct	3	0.2910	0.4011	0.7002
	5	**0.3000**	0.4079	0.7057
	7	0.2870	**0.4297**	**0.7072**
Llama3.3-70B-Instruct	3	0.2870	0.4297	0.6994
	5	**0.2940**	**0.4328**	**0.7069**
	7	0.2830	0.4251	0.7025

ditions, leaving the PPR algorithm without a solid foundation to build upon.

6.4 Rerank Study

The selection of seed nodes influences the effectiveness of PPR walks (Jimenez Gutierrez et al., 2024). To mitigate model capability biases, we used Qwen2.5-72B-Instruct and Llama-3.3-70B-Instruct to rerank seed nodes before inference. As shown in Table 7, reranking nodes prior to PPR improves document recall, both HippoRAG2 and iG^{2}RAG, while significantly enhancing RAG's response capability (F1 score). Comparing iG^{2}RAG with HippoRAG2, iG^{2}RAG consistently achieves the best F1 score, though its Recall@10 may be slightly lower in some cases. This supports our view that document recall alone does not guarantee effective information acquisition. By focusing on the information gain graph, iG^{2}RAG retrieves documents that truly reduce query uncertainty, thereby achieving superior QA performance even with lower recall.

Table 6: Performance for Different top-h.

top-h	Qwen2.5-72B			Llama3.3-70B		
	EM	F1	Recall@10	EM	F1	Recall@10
1	0.2300	0.3338	0.5396	0.2450	0.3694	0.5418
3	0.3000	0.4079	0.7057	0.2650	0.3998	0.6660
5	**0.3000**	**0.4079**	0.7057	**0.2940**	**0.4328**	0.7069
7	0.2900	0.3998	**0.7215**	0.2870	0.4267	**0.7227**

6.5 Top-k Study

By studying the top-k hyperparameter (using Qwen2.5-72B-Instruct), we can investigate the effectiveness of the information gain mechanism in iG^{2}RAG. As the performance of the Naive method

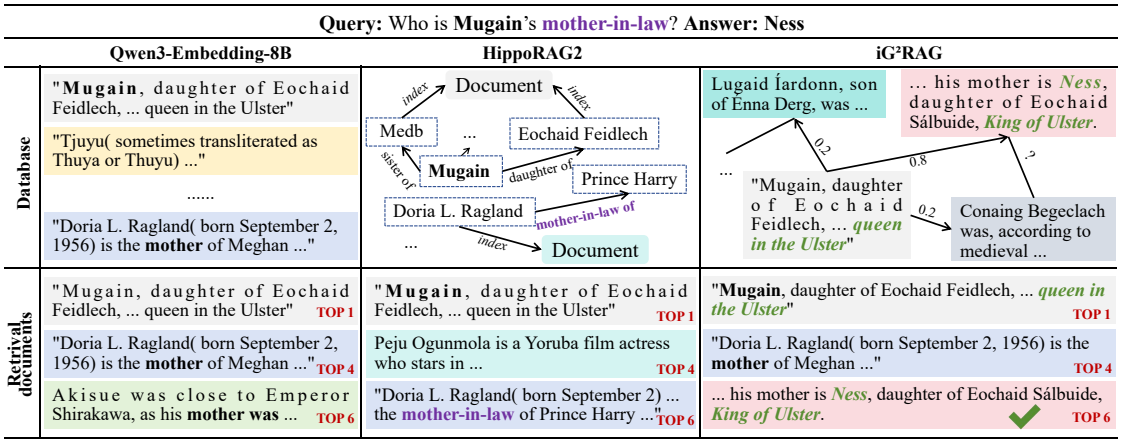

Figure 3: Case study illustrating the advantages of iG^{2}RAG. Qwen3-Embedding-8B embeds the corpus into a vector store, while graph-based RAG constructs an entity-relation database to index source documents. In contrast, iG^{2}RAG builds an information gain graph with documents as nodes, capturing informational complementarity across documents. **Black bold** text indicates similarity sources, **purple bold** denotes relational similarity, and ***green italic*** highlights complementary relations.

Table 7: Performance for Rerank Study.

Mode	Methods	Musique		2Wiki		HotpotQA	
		F1	Recall@10	F1	Recall@10	F1	Recall@10
Qwen2.5-72B-Instruct							
w/o Rerank	HippoRAG2	0.3841	0.7169	0.5807	0.8405	0.6723	0.9545
	iG^{2}RAG	0.4079	0.7057	0.6147	**0.9062**	0.6933	0.9520
w/ Rerank	HippoRAG2	0.4072	**0.7402**	0.5849	0.8720	0.6875	**0.9635**
	iG^{2}RAG	**0.4138**	0.7163	**0.6292**	0.9055	**0.7056**	0.9610
Llama-3.3-70B-Instruct							
w/o Rerank	HippoRAG2	0.3984	0.7198	0.6465	0.8505	0.6936	0.9585
	iG^{2}RAG	0.4328	0.7069	0.6891	**0.9095**	0.7144	0.9545
w/ Rerank	HippoRAG2	0.4324	**0.7369**	0.6518	0.8407	0.7076	**0.9640**
	iG^{2}RAG	**0.4522**	0.7218	**0.6908**	0.9073	**0.7174**	0.9610

Table 8: Performance for Different top-k Values.

top-k	Musique			2Wiki			HotpotQA		
	EM	F1	Recall@10	EM	F1	Recall@10	EM	F1	Recall@10
Naive									
10	0.2830	0.3957	0.7667	0.4590	0.5297	0.7900	0.5430	0.6728	0.9610
20	0.2720	0.3891	0.7035	0.4510	0.5163	0.7563	0.5390	0.6708	0.9385
iG^{2}RAG									
10	0.3000	0.4079	0.6992	0.5310	0.6147	0.9062	**0.5630**	**0.6933**	0.9520
15	0.2930	0.4051	0.7356	**0.5360**	**0.6185**	0.9197	0.5610	0.6917	0.9635
20	**0.3030**	**0.4105**	**0.7597**	0.5290	0.6164	0.9223	0.5550	0.6927	**0.9655**

in Table 8 shows, simply increasing the amount of retrieved content does not effectively yield crucial supporting evidence. Instead, it introduces unnecessary noise that interferes with the generation process. From the performance of iG^{2}RAG, we can see that although increasing top-k expands the search space for the PPR algorithm, the effective evidence is often sparse. Consequently, an overly large exploration space does not lead to significant performance improvements, because the critical evidence has already been efficiently identified in advance by PPR on the information graph. Conversely, this demonstrates that iG^{2}RAG achieves high efficiency in acquiring key evidence by leveraging information gain.

6.6 Qualitative Analysis

We provide a qualitative case study showing that iG^{2}RAG recalls crucial documents more effectively than a strong dense retriever and HippoRAG2. For the query "*Who is Mugain's mother-in-law?*", both Qwen3-Embedding-8B and HippoRAG2 initially retrieve the document about "*Mugain*" but subsequently return results focused only on "*mother*" or "*mother-in-law*," which are insufficient to answer the question. In contrast, iG^{2}RAG links the "*Mugain*" document to a complementary one via its information gain graph and retrieves it through PPR, enabling the correct reasoning path.

7 Conclusion

In this paper, we presented iG^{2}RAG, a structured-enhanced RAG framework that integrates an information gain graph with the Personalized PageRank algorithm to advance multi-hop question answering. Within iG^{2}RAG, large language models are employed in the offline stage to assign gain values between document nodes, while the online retrieval stage leverages PPR to identify optimal document combinations that maximize informational gain for a given query. This process parallels animal foraging, where information gain serves as an "informational scent" guiding the search toward the most valuable evidence. Extensive experiments on multi-hop QA benchmarks validate the effectiveness and superiority of iG^{2}RAG, highlighting its potential as a new paradigm for retrieval-augmented reasoning.

Limitations

Although iG^{2}RAG outperforms existing baseline methods in overall QA and document retrieval capabilities, it still has several limitations:

- The information gain graph in iG^{2}RAG relies heavily on the strong comprehension capability of large language models. This reliance inevitably incurs substantial corpus processing costs during the initial development phase, representing a common challenge faced by graph-based RAG approaches.

- Although iG²RAG demonstrates strong performance on multi-hop questions, the optimal choices of edge number in the information gain graph (top-n) and seed node number in the PPR process (top-h) are not fixed, but rather correlated with the intrinsic distribution of different datasets. Therefore, a key challenge for iG²RAG lies in how to pre-bind top-n and top-h to dataset characteristics, thereby reducing manual intervention.

Despite these limitations, these issues also provide clear directions for future research and optimization.

References

Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chin-Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Mahashweta Das, and 1 others. 2024. Main-rag: Multi-agent filtering retrieval-augmented generation. *arXiv preprint arXiv:2501.00332*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Guanting Dong, Jiajie Jin, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. 2025. Rag-critic: Leveraging automated critic-guided agentic workflow for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3551–3578.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Yunfan Gao, Yun Xiong, Yijie Zhong, Yuxi Bi, Ming Xue, and Haofen Wang. 2025. Synergizing rag and reasoning: A systematic review. *arXiv preprint arXiv:2504.15909*.

Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

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. *arXiv preprint arXiv:2502.14802*.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, and 1 others. Deep research agents: A systematic examination and roadmap, 2025. *URL https://arxiv.org/abs/2506.18096*.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, and 1 others. 2025. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review*, 106(4):643.

Ahmmad OM Saleh, Gökhan Tür, and Yucel Saygin. 2024. Sg-rag: Multi-hop question answering with large language models through knowledge graphs. In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 439–448.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented generation: A survey on agentic rag, 2025. *URL https://arxiv.org/abs/2501.09136*.

Tianyang Xu, Haojie Zheng, Chengze Li, Haoxiang Chen, Yixin Liu, Ruoxi Chen, and Lichao Sun. 2025. Noderag: Structuring graph-based rag with heterogeneous nodes. *arXiv preprint arXiv:2504.11544*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025a. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models.

Yi Zhang, Jamie Callan, and Thomas Minka. 2002. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88.

Jihao Zhao, Zhiyuan Ji, Yuchen Feng, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Meta-chunking: Learning text segmentation and semantic completion via logical perception. *arXiv preprint arXiv:2410.12788*.

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. Efficientrag: Efficient retriever for multi-hop question answering. *arXiv preprint arXiv:2408.04259*.

Appendix

A Appendix: LLM Prompts

We mainly present the graph construction prompt of iG^{2}RAG and the QA prompt of the RAG pipeline. This prompt primarily assigns information gain evaluation levels to the LLM, provides few-shot examples, and requires the LLM to generate outputs in a structured format.

A.1 LLM Prompts of iG^{2}RAG

We present the prompts used for building the information gain graph of iG^{2}RAG and the reasoning process in Figure 4. The main objective of this prompt is to guide the LLM in evaluating the informational gain of the *evaluation documents* relative to the *baseline document*, with the assessment categorized into four levels. Furthermore, the output is structured to facilitate subsequent parsing.

A.2 QA Prompts of RAG Pipeline

The prompt shown in Figure 5 is used in all RAG pipelines of this paper. We employed a unified QA prompt to ensure fair evaluation, preventing biases that may arise from inconsistent output length distributions across models.

A.3 LLM Prompts of Others

In addition to maintaining a unified final-answer prompt across all methods, each variant of the structured retrieval-augmented generation employs its own default prompts for knowledge graph construction. These include prompts for entity, relation, and phrase extraction, as well as prompts for graph comprehension and summarization.

B Appendix: Detailed Implementation

B.1 LLM and Embedding Models settings

In this paper, all LLMs use a temperature coefficient of 0, with the maximum model output length set to 8192 tokens and the model's context length opened up to 32648 tokens. For the Embedding models, the context length is only opened up to 8192, the encoding format adopts the float type, and the embedding dimension of the Qwen3-Embedding-8B model used is 4096, while that of the other models is 1024.

B.2 Hyperparameters of Structure-Augmented RAG

iG^{2}RAG. Unless otherwise specified for analyzing the impact of parameter variations, we keep

Information Gain Graph Prompt

You are an expert analyst specializing in textual information assessment. Your task is to meticulously evaluate the "information gain" of a "Paragraphs to Evaluate" relative to a "Baseline Paragraph".
Primary Goal: The purpose of this evaluation is to identify paragraphs that add new, significant, and complementary information to the baseline, helping to build a more comprehensive understanding of the core topic.
Definition of Information Gain: Information gain is a score from 0.0 to 1.0 that quantifies how much new, relevant, and significant information the "Paragraph to Evaluate" provides beyond what is already present in the "Baseline Paragraph". Relevance is determined by how closely the new information relates to the main subject and key entities of the baseline.
Evaluation Rubric (Score from 0.0 to 1.0):
0.0 - 0.2 (Redundant / Trivial): The paragraph largely repeats information already in the baseline, or adds only trivial, irrelevant details (e.g., rephrasing, common knowledge filler).
0.3 - 0.5 (Minor Addition): The paragraph adds some new, relevant information, but it is of low significance. It might be a minor detail, a specific example of a point already made, or a slight expansion of an existing concept.
0.6 - 0.8 (Significant Addition): The paragraph introduces substantial new facts, perspectives, or details that are highly relevant and complementary to the baseline. This new information significantly enriches the understanding of the core topic.
0.9 - 1.0 (Highly Novel & Critical): The paragraph provides entirely new information that is critical and highly relevant to the baseline's topic. It might introduce a new dimension, a surprising counter-argument, or key data that fundamentally changes or completes the picture presented in the baseline.

Example

Baseline paragraph:
FC Barcelona
Barcelona is one of three founding members of the Primera División that have never been relegated from the top division, along with Athletic Bilbao and Real Madrid. In 2009, Barcelona became the first Spanish club to win the continental treble consisting of La Liga, Copa del Rey, and the UEFA Champions League, and also became the first football club to win six out of six competitions in a single year, completing the sextuple in also winning the Spanish Super Cup, UEFA Super Cup and FIFA Club World Cup. In 2011, the club became European champions again and won five trophies. This Barcelona team, which reached a record six consecutive Champions League semi-finals and won 14 trophies in just four years under Pep Guardiola, is considered by some in the sport to be the greatest team of all time. In June 2015, Barcelona became the first European club in history to achieve the continental treble twice.
</baseline_content>
Paragraphs to evaluate:
<chunk_id>doc_3_78ee7a4c1f82</chunk_id>
<content>
FC Barcelona
Barcelona is the only European club to have played continental football every season since 1955, and one of three clubs to have never been relegated from La Liga, along with Athletic Bilbao and Real Madrid. In 2009, Barcelona became the first club in Spain to win the treble consisting of La Liga, Copa del Rey, and the Champions League. That same year, it also became the first football club ever to win six out of six competitions in a single year, thus completing the sextuple, comprising the aforementioned treble and the Spanish Super Cup, UEFA Super Cup and FIFA Club World Cup. In the 2014–15 season, Barcelona won another historic treble, making them the first club in European football to win the treble twice.
</content>
<reasoning>
Most facts are already in the baseline (treble in 2009, sextuple, treble in 2015). The only new fact is "only European club to have played continental football every season since 1955", which is minor new info. Therefore, low gain.
</reasoning>
<info_gain>0.2</info_gain><chunk_id>doc_6_6953a7af7c5b</chunk_id>
<content>
FC Barcelona
FC Barcelona had a successful start in regional and national cups, competing in the Campionat de Catalunya and the Copa del Rey. In 1902, the club won its first trophy, the Copa Macaya, and participated in the first Copa del Rey, losing 1–2 to Bizcaya in the final. Hans Gamper — now known as Joan Gamper — became club president in 1908, finding the club in financial difficulty after not winning a competition since the Campionat de Catalunya in 1905. Club president on five separate occasions between 1908 and 1925, he spent 25 years in total at the helm. One of his main achievements was ensuring Barça acquire its own stadium and thus generate a stable income.
</content>
<reasoning>
Almost entirely new historical details about early years, trophies, and leadership. Very little overlap with baseline's modern achievements.
</reasoning><info_gain>0.9</info_gain>
Expected JSON output for example:
{"data": [{"chunk_id": "doc_3_78ee7a4c1f82", "info_gain": 0.2}, {"chunk_id": "doc_6_6953a7af7c5b", "info_gain": 0.9},]}

Now your task:

You will receive a new baseline paragraph and new paragraphs to evaluate in the exact same format.
Instructions:
For each paragraph, compare with the baseline paragraph. In your internal reasoning, follow the same explanation style as the example above, but DO NOT include explanations in the final output. Output only the JSON with chunk_id and info_gain. Keep <chunk_id> exactly as provided. Do not change order. Values must be numbers with one decimal place.
The JSON should contain the `chunk_id` and `info_gain` of each evaluated paragraph, with no omissions allowed.
Baseline paragraph:
{baseline}
Paragraphs to evaluate:
{chunks_text}
Output (Only JSON):

Figure 4: LLM prompts used in iG^{2}RAG for constructing the information gain graph and reasoning process.

QA Prompt

Based on the following retrieved content, please answer the user's question.

User Question: {query}
Retrieved Content:
{pasg_content}
Instructions:
1. Analyze the content to find relevant information.
2. Provide only the final answer without any explanation or analysis process.
3. If multiple pieces contain relevant information, synthesize them appropriately.
4. Only the final answer, no other content. The answer may be entities such as names of people, organizations, places, relationships, times, dates, etc.
Answer (Only the final answer, no other content):

Figure 5: QA prompt used in the RAG pipeline.

all hyperparameters consistent across experiments. Specifically, the number of information gain edges (*top-n*) is fixed at 5, and the number of seed nodes used in the random walk (*top-h*) is also set to 5. The restart probability in Personalized PageRank, denoted as α, for Qwen2.5-72B-Instruct, α is set to 0.3 for Musique and HotpotQA, and 0.5 for 2Wiki. For Llama3.3-70B-Instruct, α is set to 0.5 for Musique, 0.2 for HotpotQA, and 0.3 for 2Wiki.

RAPTOR, constructs a hierarchical semantic tree by recursively clustering and abstracting textual units: starting from fine-grained leaf nodes (e.g., sentences or paragraphs), semantically similar nodes are iteratively grouped based on embedding similarity, and a LLM generates abstractive summaries to form parent nodes, resulting in a multi-level tree that enables retrieval at varying granularities. In our implementation, the tree depth is set to 5, and a pruning threshold of 0.5 is applied to the cosine similarity between node embeddings, only pairs exceeding this threshold are merged, thereby balancing abstraction fidelity and retrieval precision.

LightRAG, is a graph-based RAG method that constructs a knowledge graph by extracting entities from documents and linking them based on co-occurrence and contextual relationships to capture structured semantic dependencies. For LightRAG, we adopt the "local" query model setting, which restricts retrieval to the immediate neighborhood of relevant entities in the graph, thereby focusing on contextually coherent and relationally grounded evidence.

HippoRAG2, is a retrieval framework that structures knowledge into a graph of phrase nodes, enabling associative search across concepts. For HippoRAG2, the number of documents initially retrieved is set to 100, with each node forming up to 10 links. The PPR restart probability α is fixed at 0.5.

For experimental settings not explicitly specified in the baseline methods, we adhere to their default implementations. Furthermore, apart from the final answer prompt, which is kept identical across all methods, the prompts used for constructing knowledge graphs in each structured-augmented RAG variant also follow their respective defaults.

B.3 Selection of the parameter α

The PPR damping factor α, also reflects the restart probability of the random walk on the graph. To investigate whether its specific value has a significant impact on the performance of iG^{2}RAG, we conducted experiments with multiple values for α. As shown in Tables 9 and 10, the variation in α has a relatively minor impact on model performance, regardless of the dataset or the large language model used. However, when α is set to zero, which represents no random walk (a 100% restart probability), the outcome depends entirely on the initial seed node selection. This leads to a significant drop in both QA and retrieval performance. This finding demonstrates that while the specific non-zero value of α may not strongly influence the results, a value of zero is completely ineffective, different non-zero values primarily affect the convergence speed of the PPR algorithm. Therefore, this also highlights the soundness of the graph in iG^{2}RAG, as it can

Table 9: Performance of Qwen2.5-72B-Instruct on three multi-hop datasets under different damping factors (α). Dark orange indicates the optimal option, while light orange indicates the suboptimal one.

α	Musique			HotpotQA			2Wiki		
	EM	F1	Recall@10	EM	F1	Recall@10	EM	F1	Recall@10
0.8	0.2830	0.3931	0.7027	0.5500	0.6865	0.9505	0.5310	0.6085	0.9005
0.7	0.2920	0.3998	0.7059	0.5590	0.6927	0.9525	0.5300	0.6120	0.9028
0.6	0.2860	0.3955	0.7050	0.5620	0.6936	0.9520	0.5300	0.6120	0.9050
0.5	0.2900	0.4002	0.7069	0.5610	0.6907	0.9520	0.5310	0.6147	0.9062
0.4	0.2920	0.4010	0.7064	0.5600	0.6915	0.9525	0.5260	0.6116	0.9060
0.3	0.3000	0.4079	0.7057	0.5630	0.6933	0.9520	0.5270	0.6147	0.9060
0.2	0.2930	0.4052	0.7006	0.5590	0.6921	0.9520	0.5290	0.6152	0.9065
0.0	0.2430	0.3505	0.6098	0.5170	0.6439	0.8795	0.4380	0.5024	0.7210

Table 10: Performance of Llama3.3-70B-Instruct on three multi-hop datasets under different damping factors (α).

α	Musique			HotpotQA			2Wiki		
	EM	F1	Recall@10	EM	F1	Recall@10	EM	F1	Recall@10
0.8	0.2940	0.4291	0.6835	0.5590	0.7042	0.9510	0.5710	0.6578	0.9055
0.7	0.2870	0.4232	0.7096	0.5610	0.7074	0.9510	0.5730	0.6593	0.9073
0.6	0.2840	0.4247	0.7085	0.5590	0.7024	0.9540	0.5900	0.6759	0.9093
0.5	0.2940	0.4328	0.7069	0.5630	0.7104	0.9540	0.5933	0.6755	0.9095
0.4	0.2840	0.4222	0.7060	0.5610	0.7116	0.9540	0.6060	0.6853	0.9093
0.3	0.2930	0.4290	0.7058	0.5580	0.7100	0.9540	0.6080	0.6891	0.9095
0.2	0.2910	0.4292	0.7064	0.5640	0.7144	0.9545	0.6040	0.6845	0.9097
0.0	0.2610	0.3880	0.6098	0.5200	0.6623	0.8795	0.4960	0.5657	0.7220

achieve excellent performance without requiring strong intervention from the α parameter. Additionally, across the three datasets, the optimal choices for α with Qwen2.5-72B-Instruct were 0.3, 0.3, and 0.5, respectively, while for Llama3.3-70B-Instruct, they were 0.5, 0.2, and 0.3, respectively.

B.4 Evaluation Metrics

We evaluate each method using three standard metrics in RAG: Exact Match (EM), F1, and Recall@10.

Exact Match. The EM score evaluates whether the predicted answer exactly matches at least one ground-truth answer after normalization:

$$\text{EM} = \frac{1}{N}\sum_{i=1}^{N} \max_{y \in Y_i} \mathbb{I}(\text{norm}(\hat{y}_i) = \text{norm}(y)), \quad (9)$$

where N is the total number of questions, $\hat{y}_i$ is the prediction for the i-th question, Y_i is the set of ground-truth answers, $\text{norm}(\cdot)$ denotes text normalization, and $\mathbb{I}(\cdot)$ is the indicator function.

F1 Score. The F1 score is computed at the character level and averaged across all samples:

$$\text{F1} = \frac{1}{N}\sum_{i=1}^{N} \frac{2 \cdot \text{Precision}(\hat{y}_i, y_i) \cdot \text{Recall}(\hat{y}_i, y_i)}{\text{Precision}(\hat{y}_i, y_i) + \text{Recall}(\hat{y}_i, y_i)}, \quad (10)$$

where N is the total number of questions, $\hat{y}_i$ is the prediction for the i-th question, y_i is its ground-truth answer, and both Precision and Recall are computed based on character-level overlap between $\hat{y}_i$ and y_i.

Recall@10. Recall@10 evaluates how many ground-truth documents are covered within the top-10 retrieved results:

$$\text{Recall@10} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|D_i^{\text{rel}} \cap D_i^{(10)}\right|}{\left|D_i^{\text{rel}}\right|}, \quad (11)$$

where N is the total number of questions, D_i^{rel} is the set of all ground-truth documents relevant to the i-th question, and $D_i^{(10)}$ denotes the set of top-10 retrieved documents. The numerator counts

Table 11: Performance comparison between Qwen2.5-72B-Instruct and Llama3.3-70B-Instruct on 2Wiki and HotpotQA with different top-n settings.

Models	**Top-n**	**2Wiki**			**HotpotQA**		
		EM	F1	Recall@10	EM	F1	Recall@10
Qwen2.5-72B-Instruct	3	0.5310	0.6106	0.8968	0.5430	0.6784	0.9515
	5	0.5310	0.6147	0.9095	0.5630	0.6933	0.9520
	7	0.5440	0.6233	0.9107	0.5500	0.6862	0.9506
Llama3.3-70B-Instruct	3	0.5850	0.6666	0.8945	0.5680	0.7104	0.9495
	5	0.6080	0.6891	0.9095	0.5640	0.7144	0.9545
	7	0.6000	0.6841	0.9175	0.5700	0.7166	0.9595

the number of relevant documents successfully retrieved in the top-10.

C Appendix: Supplementary discussion

In the main discussion Section 6, we investigated the effects of edge construction in the information gain graph and node selection in the PPR stage using the Musique dataset. This choice was motivated by the fact that Musique represents a more challenging multi-hop dataset, where performance fluctuations are more indicative. To examine whether these conclusions generalize to a broader range of datasets, we further conducted experiments on HotpotQA and 2Wiki.

C.1 Number of gain edges

In Section 6.2, we have already concluded that too few edges lead to insufficient exploitation of informational gain, while too many edges introduce redundancy. As shown in Table 11, for the relatively less challenging multi-hop datasets 2Wiki and HotpotQA, the models can better capture the logical connections within the datasets, making it meaningful to establish additional information gain edges. These edges refine the granularity of inter-document relationships, thereby enabling the PPR process to more effectively integrate document combinations that maximize information gain. However, the performance trend suggests that there is no fixed top-n value that consistently yields the best results. Smaller top-n values fail to sufficiently exploit information gain relationships, while larger top-n values are not necessarily optimal. Instead, selecting a moderate top-n in alignment with dataset characteristics provides more stable performance for iG^{2}RAG.

C.2 Number of seed nodes

In Section 6.3, we have concluded that too few seed nodes fail to provide an effective initialization probability for PPR, whereas an excessively large set of random seeds reduces the exploration space of PPR. As shown in Table 12, the conclusion drawn in Section 6.3 still holds for the *2Wiki* and *HotpotQA* datasets. Although some numerical fluctuations are observed, the overall trend first increases and then decreases, reflecting the stability of using a moderate number of seed nodes.

Overall, as shown in Table 11 and Table 12, the performance trends for both top-n and top-h exhibit a rise-then-fall pattern. Although in the case of the HotpotQA dataset, Llama3.3-70B-Instruct tends to favor larger values, this reflects a form of synergy between external knowledge and the model's internal knowledge. At the same time, this highlights an important phenomenon, namely that the performance peak varies with knowledge distributions of different characteristics, and such variation is inherently coupled with the parameterized knowledge of the model.

D Appendix: Incremental Update for iG^{2}RAG Graph

As detailed in Algorithm 3, the incremental updating process in iG^{2}RAG is highly efficient. It only requires using the saved embeddings of existing documents to identify the neighboring nodes for a new document, after which an LLM evaluates the information gain of these neighbors. This process does not alter the previously established graph structure at all; instead, it seamlessly incorporates the new nodes. To further investigate the efficiency of this process, we measured the number of graph edges, time, token consumption, and F1 scores during incremental updates. We started with a base of

Table 12: Performance comparison between Qwen2.5-72B-Instruct and Llama3.3-70B-Instruct on 2Wiki and HotpotQA with different top-*h* settings.

top-*h*	Qwen2.5-72B-Instruct			Llama3.3-70B-Instruct		
	EM	F1	Recall@10	EM	F1	Recall@10
2Wiki						
1	0.4570	0.5428	0.7117	0.5070	0.5897	0.7097
3	0.5430	0.6250	0.9110	0.6060	0.6837	0.9130
5	0.5310	0.6147	0.9062	0.6080	0.6891	0.9095
7	0.5300	0.6116	0.8908	0.5690	0.6529	0.8902
HotpotQA						
1	0.4930	0.62740	0.8055	0.5060	0.6539	0.8100
3	0.5430	0.67920	0.9360	0.5590	0.7056	0.9370
5	0.5620	0.6936	0.95200	0.5590	0.7024	0.9540
7	0.5480	0.68310	0.9590	0.5670	0.7092	0.9565

1,000 documents, incrementally added documents in batches of 500 up to a total of 2,500, and used a set of 100 questions to test QA performance at each stage.

The results in Table 13 show that the initial construction time for each dataset was approximately 25 minutes, consuming around 3 million tokens. When adding new documents, the additional time and token consumption for each 500-document batch were roughly proportional to the initial cost, demonstrating the efficiency and stability of the incremental updating process. Moreover, it can be observed that as the cardinality of the graph increases, the time required for incrementally updating 500 documents consistently remains around 15 minutes. This indicates the efficiency of the incremental updating mechanism, which is largely independent of the size of the original graph and instead depends primarily on the number of newly added documents. Regarding the F1 score, the Musique dataset showed improved performance as more documents were added, reaching a saturation point at 2,000 documents. For the other two datasets, however, performance saturation occurred with the initial 1,000 documents, a result directly related to the size of those datasets. Further document additions to these datasets led to fluctuations in the F1 score, which can be attributed to the imbalance between the limited number of queries and the increased information contained in the constructed graph, providing the model with more information than it can effectively align, thus introducing interpretive bias.

E Appendix: Dense Retriever Flexibility

As shown in Table 14, as the capabilities of the embedding model improve, both the F1 score and Recall of the RAG pipeline increase significantly. Furthermore, iG^{2}RAG consistently outperforms the Dense mode. This demonstrates the flexibility of iG^{2}RAG when paired with different retrievers.

F Appendix: Details of case study

Musique Case. When asking *"Who is the child of the Victim of Romance performer?"*, the query is first matched with documents based on similarity, yielding seed nodes. Among these, the document *"Victim of Romance\nVictim of Romance ..."* is most relevant to the query. However, relying solely on similarity for seed nodes lacks additional supplementary information from this document. iG^{2}RAG instead enables PPR to discover the document *"Wilson Phillips\nWilson Phillips is an ..."* within the information gain graph. Consequently, the final input to the generative model is sufficiently informative, effectively reducing the query's uncertainty. In the document *"Victim of Romance\nVictim of Romance ..."*, it is explicitly stated that *"Victim of Romance"* was performed by *"Michelle Phillips"*, while the document *"Wilson Phillips\nWilson Phillips is an ..."* further supplements that *"Chynna Phillips"* is the daughter of *"John and Michelle Phillips"*, details as shown in Fig 6.

HotpotQA Case. When asked *"Who is the film dedicated to that Paramount Classics and MTV Films co-purchased the rights to?"*, the initial

Table 13: Performance vs. incremental update across three datasets.

Nodes	Musique				2Wiki				HotpotQA			
	Edges	Time (min)	Tokens	F1	Edges	Time (min)	Tokens	F1	Edges	Time (min)	Tokens	F1
1000	3648	25.63	3143081	0.3306	2481	25.39	3091916	0.6429	2858	25.88	3204260	0.7186
1000 + 500	5079	16.03	1587661	0.4435	3953	13.31	1614195	0.6126	4403	13.27	1619992	0.7156
1500 + 500	6679	13.50	1576503	0.4667	5474	13.33	1570036	0.6290	5980	13.33	1606094	0.7019
2000 + 500	8173	14.02	1582141	0.4536	7065	13.66	1546862	0.6376	7550	13.34	1609052	0.7119

Algorithm 3 Incremental Update

Require: Existing information gain graph $\mathcal{G}_{info}(\mathcal{V}, \mathcal{E}_{info})$, New document chunks $\mathcal{V}_{new}$, LLM $\mathcal{M}_i$, Embeddings $\mathbf{E}$, Neighbor size n
Ensure: Updated graph $\mathcal{G}'_{info}(\mathcal{V}', \mathcal{E}'_{info})$

```
Compute embeddings for new chunks: E_new ∈ R^{|V_new| × d}
Concatenate embeddings: E' ← E ⊕ E_new
for each new chunk v_new ∈ V_new do
    Find top-n similar chunks from existing V using dot-product: Neighbors(v_new) ⊆ V
    Construct subgraph: G_sub ← {v_new} ∪ Neighbors(v_new)
    for each v_j ∈ Neighbors(v_new) do
        Compute information gain: w_{new,j} ← M_i(v_new, v_j)
        Add edge: E'_info ← E'_info ∪ {(v_new, v_j, w_{new,j})}
    end for
end for
Update nodes: V' ← V ∪ V_new
Update edges: E'_info ← E_info ∪ E'_info
return G'_info(V', E'_info)
```

seed nodes primarily match prominent fields such as *"Paramount Classics"* and *"MTV Films"*, yet these alone are insufficient to substantially reduce the query's uncertainty. Through PPR expansion, iG^{2}RAG is able to retrieve the passage *"Hustle & Flow ..."*, which, when connected to *"Beneath (2007 film) Beneath is a straight-to-DVD ..."*, provides the critical complementary information that the film is dedicated to *"Sam Phillips"*, details as shown in Fig 7.

2Wiki Case. When requesting *"Who is Mugain's mother-in-law?"* (a detailed case in Figure 3), the query first retrieves relevant documents from the seed nodes via the key field *"Mugain"*. From this document, it can be learned that *"Mugain"* is the wife of *"Conchobar mac Nessa"*, and that *"Conchobar mac Nessa"* is the *"king of Ulster"*. The documents obtained through PPR expansion provide further information on *"Conchobar mac Nessa"*, from which it is revealed that his mother is *"Ness"*. Therefore, *"Mugain's mother-in-law"* is *"Ness"*, details as shown in Fig 8.

Table 14: Performance on different embedding models.

Embedding Models	Dense		iG^{2}RAG	
	F1	Recall	F1	Recall
BGE-M3	0.3411	0.6189	0.3577	0.6279
text-embedding-v3	0.3630	0.6366	0.3743	0.6427
Qwen3-Embedding-8B	0.3891	0.7035	0.4138	0.7163

Table 15: Differences and p-values for F1 Scores on Datasets

Metric	Musique		2Wiki		HotpotQA	
	Diff	p-value	Diff	p-value	Diff	p-value
F1	0.0345	0.0004	0.029	0.0117	0.0168	0.041

G Appendix: Statistical Significance Test

To verify that the improvements of iG^{2}RAG over HippoRAG2 are not simply due to random variation, we conducted statistical significance testing using a paired permutation test on the F1 scores. The permutation test works by repeatedly shuffling the pairwise results of the two methods and recalculating the performance difference, thereby estimating how likely it is to observe a difference at least as large as the actual one under the assumption that both methods perform equivalently. The resulting probability is reported as the p-value. By convention, a threshold of 0.05 is used: if the p-value is below this level, it means that the chance of the observed improvement arising purely from random fluctuations is less than 5%. In other words, a low p-value provides strong evidence that the performance gain is systematic and meaningful. As shown in Table 15, all datasets yield p-values well below 0.05, confirming that the improvements of iG^{2}RAG are statistically significant rather than random artifacts.

Table 16: Graph Statistics for Different Methods on Datasets

Method	Musique		2Wiki		HotpotQA	
	Nodes	Edges	Nodes	Edges	Nodes	Edges
iG^{2}RAG	11656	58281	6119	30595	9811	49055
HippoRAG2	89364	2031466	45947	1287189	83854	1861013

H Appendix: Graph Statistics

In Table 16, we present the number of graph nodes and edges in iG^{2}RAG compared to other methods on Qwen2.5-72B-Instruct setting. This demonstrates that iG^{2}RAG is more efficient and more advantageous for management and storage. Additionally, in Figure 9, we illustrate the local graph structure centered on the node "*Lionel Messi After a year a...*".

I Appendix: Detailed hardware and software settings

The iG^{2}RAG framework relies on the OpenAI API request format [2]. For deployment, we use vLLM [3] to run the large language model (LLM) on four NVIDIA A800 Tensor Core GPUs and expose it as an API that is compatible with the OpenAI interface. Similarly, for the embedding model, we deploy it on a single NVIDIA GeForce RTX 4090 using vLLM and also interact with it using the OpenAI API format. We chose vLLM for deployment and the OpenAI interface for requests to facilitate easy extension to different models, a benefit derived from the continuous development and support of both communities.

[2] https://github.com/openai/openai-python
[3] https://github.com/vllm-project/vllm

iG²RAG

Query: Who is the child of the Victim of Romance performer? Answer: Chynna Phillips

Seed Nodes

Victim of Romance\n***Victim of Romance is singer and songwriter Michelle Phillips***' first and only solo album, and was released in February 1977 (see 1977 in music). The record was unsuccessful and Phillips (previously with The Mamas & the Papas) then favored her acting career. The front cover photography was by Terry O'Neill.

The Last Song\nAt seventeen, Veronica ``Ronnie "Miller (Miley Cyrus) remains as rebellious as she was the day her parents divorced and her father moved to North Carolina three years prior. Once a classical piano child prodigy under the tutelage of her father, Steve Miller (Greg Kinnear), Ronnie now ignores the instrument and has not spoken with her father since he left. While Juilliard School has been interested in her since she was young, Ronnie refuses to attend.

Louis Chedid\nLouis Chedid is **the son of** the writer Andrée Chedid and the father of Matthieu Chedid (better known as -M-).

American Idol\nPhillips became the winner, beating Sanchez. Prior to the announcement of the winner, season five finalist Ace Young proposed marriage to season three runner-up Diana DeGarmo on stage – which she accepted.

Lina Medina\nMedina has never revealed the father of the child nor the circumstances of her impregnation. Escomel suggested that she might not actually know herself, as she \"couldn't give precise responses\". Lina's father was arrested on suspicion of child sexual abuse, but he was released due to lack of evidence and the biological father was never identified. **Her son grew up healthy**. He died in 1979 at the age of 40. In young adulthood, Medina worked as a secretary in the Lima clinic of Lozada, which gave her an education and helped put her son through high school. She married Raúl Jurado, who fathered her second son in 1972. As of 2002, they lived in a poor district of Lima known as \"Chicago Chico\". She refused an interview with Reuters that year, just as she had turned away many reporters in years past.

PPR to Expand

The Official Story\nThe film deals with the story of an upper middle class couple who lives in Buenos Aires with an illegally adopted child. The mother comes to realize that her daughter may be the child of a desaparecido, a victim of the forced disappearances that occurred during Argentina's last military dictatorship (1976 - 1983), which was marred by widespread human rights violations and a genocide.

The Last Song\nKelly Preston as Kim Miller, Ronnie and Jonah's mother who raised her children in New York City after her divorce.

Wilson Phillips\nWilson Phillips is an American vocal group consisting of Carnie Wilson, Wendy Wilson, and ***Chynna Phillips, the daughters***, respectively, of Brian Wilson of The Beach Boys and ***of John and Michelle Phillips*** of The Mamas & the Papas.

Matthieu Chedid\nMatthieu Chedid was born in Boulogne-Billancourt, Hauts-de-Seine, France, the son of French singer Louis Chedid, and the grandson of the Egyptian-born French writer and poet of Lebanese descent Andrée Chedid who has written lyrics for him. His sister is the music video and concert director Emilie Chedid.

Je dis aime\nJe dis aime (1999) is the second studio album by French singer-songwriter Matthieu Chedid, in his persona as -M-, described by reviewers as a \"conceptual icon to rival Bowie's Ziggy Stardust and Aladdin Sane\". The album manages to take a remarkable variety of musical directions and pull them together into a consistent whole. Another reviewer describes the album as sounding like a 'French Lenny Kravitz' and notes the 'vintage 70s sound and textures'.

Figure 6: Details of Musique case study. It details the selection of seed documents (Seed Nodes) for iG^{2}RAG in Musique and the documents obtained through the random walk expansion of PPR (PPR to Expand). Among them, documents with the same background color are indicated as complementary documents, even if they are not retrieved during seed selection, they can be obtained through PPR expansion to supplement multi-hop questions. In the document nodes, **black bold text** indicates fields with a relatively large contribution to similarity, while ***red italic bold text*** indicates fields that complement each other in terms of facts.

iG²RAG

Query: Who is the film dedicated to that Paramount Classics and MTV Films co-purchased the rights to? Answer: Sam Phillips

Seed Nodes

MTV Films\nMTV Films is the motion picture production arm of the American cable television channel MTV. Founded in 1996, it has produced films based on MTV programs such as \"Beavis and Butt-head Do America\" and \"\", as well as other adaptations and original projects. Its films are released by fellow Viacom division Paramount Pictures. On August 21, 2006, Nickelodeon Movies and MTV Films became full labels of the Paramount Motion Pictures Group. It currently has 40 films with four direct-to-video titles.

Paramount Vantage\nParamount Vantage (originally known as Paramount Classics) was the specialty film division of Paramount Pictures (which, in turn, has Viacom as its parent company), charged with producing, purchasing, distributing and marketing films, generally those with a more \"art house\" feel than films made and distributed by its parent company.

Beneath (2007 film)\nBeneath is a straight-to-DVD thriller-horror film co-produced in a first time partnership between Paramount Classics (a Viacom subsidiary) and MTV Films (although both co-purchased the rights to \"Hustle & Flow\" in 2005). The film is directed by the newcomer Dagen Merrill, who co-wrote the script with Kevin Burke, and the list of producers include Sean Covel and Chris Wyatt (\"Napoleon Dynamite\", \"Think Tank\"), as well as Troy Craig Poon. In Paramount Classics's first horror movie, which marks the company's expansion from acquisitions into the production arena, the cast includes Nora Zehetner (\"Brick\", \"May\", \"R.S.V.P.\", \"Everwood\") and Matthew Settle (\"U-571\", \"Band of Brothers\"). Shooting started 2005 in Vancouver, the film was released on DVD August 7, 2007. It was the first direct-to-video title produced by MTV Films. ✓

Paramount Famous Productions\nParamount Famous Productions is a made-for-home entertainment division of Paramount Pictures. It develops home entertainment sequels to films from Paramount Pictures, Paramount Vantage, **MTV Films**, DreamWorks Pictures (pre-2005 library), Nickelodeon Movies, and other Paramount-related properties. The name also revives the \"Famous\" moniker previously used by the Paramount-owned Famous Studios.

Jackass: The Movie\nJackass: The Movie is a 2002 American reality comedy film directed by Jeff Tremaine with the tagline \"Do not attempt this at home.\" It is a continuation of the stunts and pranks by the various characters of the MTV television series \"Jackass\", which had completed its unique series run by this time. The film was produced by **MTV Films** and Dickhouse Productions and released by **Paramount** Pictures.

PPR to Expand

Jackass Number Two\nJackass Number Two is a 2006 American reality comedy film. It is the sequel to \"\" (2002), both based upon the MTV series \"Jackass\". Like its predecessor and the original TV show, the film is a compilation of stunts, pranks and skits. The film stars the regular \"Jackass\" cast of Johnny Knoxville, Bam Margera, Chris Pontius, Steve-O, Ryan Dunn, Dave England, Jason \"Wee Man\" Acuña, Preston Lacy and Ehren McGhehey. Everyone depicted in the film plays as themselves. All nine main cast members from the first film returned for the sequel. The film was directed by Jeff Tremaine, who also directed \"\" and produced \"Jackass\".

Hustle & Flow\n***Hustle & Flow is a 2005 American independent drama film written and directed by Craig Brewer and produced by John Singleton and Stephanie Allain.*** It was released on July 22, 2005. Terrence Howard stars as a Memphis hustler and pimp who faces his aspiration to become a rapper. The film is dedicated to Sun Records founder Sam Phillips. ✓

Ryan Dunn\nRyan Matthew Dunn (June 11, 1977 – June 20, 2011) was an American stunt performer, television personality, comedian, actor, writer, musician, and one of the stars of the MTV reality stunt show \"Jackass\".

Dave England\nDave England (born December 30, 1969) is an American stunt performer, and former professional snowboarder. He is best remembered as one of the stars of the MTV reality stunt show \"Jackass\".

Chris Pontius\nChristopher Andrew \"Chris\" Pontius (born July 16, 1974) is an American stunt performer, actor, musician, and a cast member of the MTV reality stunt show \"Jackass\" and also co-hosted its spinoff \"Wildboyz\" with fellow cast member Steve-O.

Figure 7: Details of HotpotQA case study.

iG²RAG

Query: Who is Mugain's mother-in-law? Answer: Ness

Seed Nodes

Mugain\n***Mugain***, daughter of Eochaid Feidlech, is a legendary queen in the Ulster Cycle of Irish mythology; ***characterized as the \"Strumpet wife of Conchobar mac Nessa\", the king of Ulster***. Also styled Mumain , she bore him a son named Glaisne. She was also a sister of Medb by paternity. Her epithet, \"Aitinchairchech\", literally means \"having gorse-like body hair\", or perhaps more specifically pubic hair. When Cúchulainn returned to Emain Macha after his first foray, his fury was so great the Ulstermen feared he would destroy them. Mugain led her maidens out, and they bared their breasts in front of him. Cúchulainn averted his eyes, and the Ulstermen were able to wrestle him into a barrel of cold water, which exploded from the heat of his body. They put him in a second barrel, and the water boiled; and finally a third barrel, which merely warmed up to a pleasant temperature. Her affair with Aed, Conchobar's poet, led to the death of Lóegaire Búadach. The Ulstermen took her life, out of the love of her, though they seldom engaged in femicide.\np.23

Tjuyu\nTjuyu(sometimes transliterated as Thuya or Thuyu) was an Egyptian noblewoman and the **mother of** queen Tiye, and the wife of Yuya. She is the grandmother of Akhenaten, and great grandmother of Tutankhamun.

Minamoto no Chikako\nShe **was the mother of** Prince Morinaga.

Doria Ragland\nDoria L. Ragland(born September 2, 1956) **is the mother of** Meghan, Duchess of Sussex, the **mother- in- law** of Prince Harry, and the maternal grandmother of Archie Mountbatten- Windsor, who is seventh in line to the British throne. She is a former makeup artist, business owner, yoga instructor, and former social worker who worked in the mental health sector from 2015 to 2018.

Maria Thins\nMaria Thins(c. 1593 – 27 December 1680) was the **mother- in- law** of Johannes Vermeer and a member of the Gouda Thins family.

PPR to Expand

Conchobar mac Nessa\nConchobar mac Nessa ***(son of Ness) is the king of Ulster in the Ulster Cycle of Irish mythology***. He rules from Emain Macha (Navan Fort, near Armagh). He is usually said to be the son of the High King Fachtna Fáthach, although in some stories his father is the druid Cathbad, and he is usually known by his matronymic, \"mac Nessa\": his mother is Ness, daughter of Eochaid Sálbuide, King of Ulster.

Nefertiti\nNeferneferuaten Nefertiti (c. 1370 – c. 1330 BC) was an Egyptian queen and the Great Royal Wife of Akhenaten, an Egyptian Pharaoh. Nefertiti and her husband were known for a religious revolution, in which they worshipped one god only, Aten, or the sun disc. With her husband, she reigned at what was arguably the wealthiest period of Ancient Egyptian history. Some scholars believe that Nefertiti ruled briefly as Neferneferuaten after her husband's death and before the accession of Tutankhamun, although this identification is a matter of ongoing debate. If Nefertiti did rule as Pharaoh, her reign was marked by the fall of Amarna and relocation of the capital back to the traditional city of Thebes. Nefertiti had many titles including Hereditary Princess (iryt-p`t); Great of Praises (wrt-Hzwt); Lady of Grace (nbt-im3t), Sweet of Love (bnrt-mrwt); Lady of The Two Lands (nbt-t3wy); Main King's Wife, his beloved (Hmt-nswt-‘3t mryt.f); Great King's Wife, his beloved (Hmt-nswt-wrt mryt.f), Lady of All Women (Hnwt-Hmwt-nbwt); and Mistress of Upper and Lower Egypt (Hnwt-Shm’w-mhw). She was made famous by her bust, now in Berlin's Neues Museum. The bust is one of the most copied works of ancient Egypt. It was attributed to the sculptor Thutmose, and it was found in his workshop.

Shuttarna II\nShuttarna II (or Šuttarna) was a king of the Hurrian kingdom of Mitanni in the early 14th century BC. Shuttarna was a descendant and probably a son of the great Mitannian king Artatama I. He was an ally of the Egyptian Pharaoh Amenhotep III and the diplomatic dealings of the kings are briefly recorded in the Amarna letters. Shuttarna's daughter Kilu-Hepa (sometimes spelled Gilukhepa) was given to Amenhotep III in marriage to seal the alliance between the two royal houses in the Pharaoh's 10th regnal year, taking with her a great dowry. During the reign of Shuttarna, the kingdom of Mitanni reached its height of power and prosperity. From Alalakh in the west, Mitanni shared its border with Egypt in northern Syria, approximately by the river Orontes. The heart of the kingdom was in the Khabur River basin where the capital Washshukanni was situated. Assyria as well as Arrapha in the east were vassal kingdoms of Mitanni. The Hittites attempted to invade the northern border lands of Mitanni, but were defeated by Shuttarna. He was succeeded by his son, Tushratta, or possibly Artashumara, under dubious circumstances.

Fujiwara no Hiroko\n, also known as, was an empress consort of Emperor Go- Reizei. She was the eldest daughter of Fujiwara no Yorimichi and Fujiwara no Gishi. Fujiwara no Morozane was her brother by the same mother.

Margaretha van Godewijk\nMargaretha van Godewijk (30 August 1627, Dordrecht – 2 November 1677, Dordrecht), was a Dutch Golden Age poet and painter.

Figure 8: Details of 2Wiki case study.

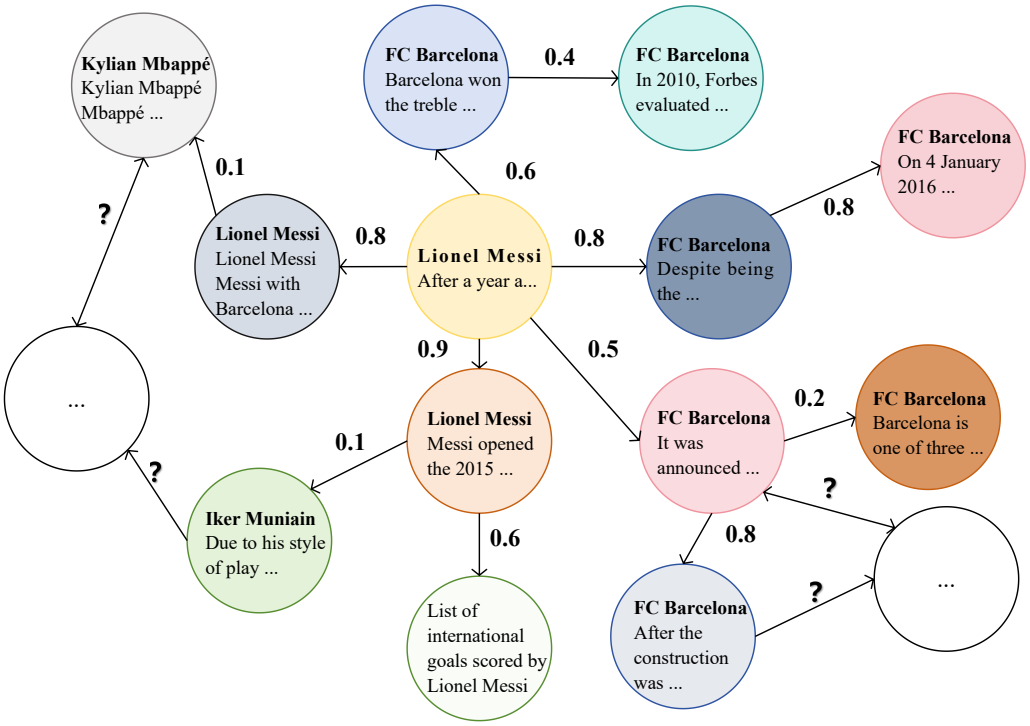

Figure 9: Informational graph structure centered on the node "*Lionel Messi After a year a...*".