Visual and Graph-Augmented Evaluation of RAG Systems – Improving Retrieval, Re-Ranking, and Data Quality with the pRAG Framework

Alexander Kreß*, Alexander Lawall*, Thomas Zöller*
*IU International University of Applied Science
Erfurt, Germany
alexander.kress@iu-study.org, alexander.lawall@iu.org, thomas.zoeller@iu.org

***Abstract*—Retrieval-Augmented Generation systems require evaluation methods that expose weaknesses within individual components rather than relying solely on end-to-end metrics. This paper introduces pRAG, a visual and graph-augmented framework that provides interpretable, component-level insights into retrieval, re-ranking, and context selection without using LLM-based judgment. pRAG combines specialized recall metrics, ranking indicators, and timing analysis to identify bottlenecks-most notably in the retrieval stage. We further propose a scalable synthetic dataset pipeline to reduce the need for manually curated ground truth. Experiments show that synthetic question-answer pairs achieve comparable or superior evaluation quality, with human experts preferring them in most cases. The extension of pRAG with knowledge graph relations enables the analysis of contextual dependencies between chunks and supports more informed reasoning. Our results highlight that hybrid retrieval strategies and optimized retrieval sizes significantly improve RAG performance. Overall, pRAG offers an efficient and interpretable methodology for evaluating complex RAG pipelines and supports scalable assessment in specialized domains.**

Keywords-Retrieval-Augmented Generation (RAG); Component-Level Evaluation; Synthetic Data Generation; Knowledge Graph Integration; Retrieval Optimization.

I. Introduction

This work is an extended version of *Retrieval Performance in RAG Systems: A Component-Level Evaluation Framework*, published at GPTMB 2025 [1]. Retrieval-Augmented Generation (RAG) systems are important for improving the factuality and reliability of Large Language Model (LLM) outputs, especially in domain-specific applications. Despite their adoption, evaluating these systems remains challenging, particularly when considering their multi-component nature and varying performance across different use cases [2].

A. Motivation and Problem Statement

The evaluation of RAG systems faces several challenges that current LLM approaches fail to adequately address. While LLMs alone can be evaluated using established benchmarks, RAG systems introduce additional complexities due to their multi-stage architecture spanning document processing, retrieval, re-ranking, and generation components [3][4]. As noted by [3], dynamic data environments further complicate evaluation, as the underlying knowledge sources often change over time.

Current evaluation frameworks typically produce aggregate metrics that mask the performance of individual components, making it difficult to identify specific bottlenecks or optimization opportunities [3][5]. Manual evaluation methods are becoming increasingly inefficient, necessitating automated approaches that can scale with system complexity. Additionally, temporal aspects of RAG performance — such as latency variations across different technical configurations — are rarely incorporated into evaluation methodologies despite their critical importance in real-world applications [3][6].

Established benchmark datasets like HotpotQA [7] and MS MARCO [8] have proven inadequate for evaluating modern RAG systems [9], as they fail to capture the nuanced retrieval and generation scenarios encountered in specialized domains. The availability of ground truth data will become rare in the future [10]. While synthetic dataset generation offers promising alternatives [11], systematic approaches for validating these datasets and incorporating them into holistic evaluation frameworks remain underdeveloped.

B. Research Gap

Despite the proliferation of evaluation methods for RAG systems, major gaps persist in current approaches. Existing frameworks like RAGAS [12] rarely provide granular insights into component-level performance, instead focusing on end-to-end evaluation that obscures the contribution of individual technical elements [3][4]. As [13] observes, the various technical alternatives available at each stage of the RAG pipeline create a complex evaluation space that remains largely unexplored. As [14] points out, this gap cannot be closed by asking LLMs for reasoning.

The role of re-ranking models [15][16] and hybrid retrieval techniques like the combination of embeddings and BM25 [17] in RAG performance is inadequately addressed by current evaluation approaches. Furthermore, while the importance of synthetic datasets for evaluation is increasingly recognized [18], methodologies for generating and validating these datasets remain ad-hoc and not standardized. These gaps demand a comprehensive evaluation framework that addresses both the technical complexity of RAG systems and the practical challenges of meaningful assessment [11].

C. Research Questions

This paper addresses the primary research question: "Which technical concepts are necessary to successfully evaluate RAG systems?". Secondary research questions are investigated to explore this question more comprehensively:

1) "How can we effectively evaluate the retrieval component in RAG systems?"

2) "How can synthetic datasets be efficiently generated and validated for RAG evaluation?"
3) "What approaches show promise for evaluating the entire RAG pipeline?"

D. Contributions

This paper makes several contributions to the field of RAG system evaluation:

- The introduction of a methodology for generating and validating synthetic evaluation datasets that can scale efficiently across domains and use cases.
- The development of a visualization approach (pRAG) for assessing retrieval component performance that incorporates both quality metrics and temporal analysis.
- The provision of empirical findings from applying the framework to a real-world RAG system designed for technical documentation question answering.
- The integration of Knowledge Graph logic into the pRAG visualization and evaluation framework

The framework addresses critical gaps in existing evaluation approaches by offering a more granular, component-specific assessment methodology that can adapt to the evolving landscape of RAG system design.

E. Paper Structure

The remainder of this paper is organized as follows: Section II describes the methodology, including the architecture of the evaluation framework, synthetic dataset generation approach, and component-specific assessment techniques. Section III presents the results of applying the framework to a case study RAG system and discusses key findings and implications. Section IV provides a detailed evaluation and discussion of the results, analyzing the role of ground truth, retrieval performance, and the effectiveness of the proposed framework. Finally, Section V concludes the paper and outlines directions for future work.

II. Methodology

A. RAG System Architecture

The RAG system employs a microservice-based architecture designed for scalability and modular development. The system processes user queries through these key components: When a user submits a query via the frontend, the middleware API coordinates the workflow. First, the pre-processing API generates keywords and embeddings from the query for semantic comparison. These are passed to a vector handling API that performs hybrid retrieval, combining BM25 [19], keyword matching, and embedding-based semantic search through paradeDB, a PostgreSQL extension supporting vector operations.

Retrieved contexts and metadata flow back to the middleware API, which forwards them to the pre-processing API where a cross-encoder re-ranker prioritizes the most semantically relevant documents. Finally, these re-ranked contexts together with an initial prompt are provided to a LLM that generates a comprehensive response based on the available information and returns it to the user via the frontend.

B. System Implementation

The system is deployed on a Kubernetes cluster with the frontend developed in React and backend services in Python. For the knowledge base, 50 technical documents from HORSCH machinery manuals using Azure Document Intelligence are processed to convert PDF content into processable text.

For embedding generation, the Hugging Face multi-qa-MiniLM-L6-cos-v1 Sentence Transformer model [20] was implemented, selected for its balance of English language capabilities and computational efficiency. Documents were chunked to match the model's maximum token length and then stored with machine-specific metadata.

We evaluated two cross-encoder models for re-ranking: msmarco-MiniLM-L6-en-dev1 [21] and ms-marco-MiniLM-L-6-v2 [22], which reorder retrieved contexts based on query relevance. For response generation, we utilized ChatGPT-3.5-Turbo-0125 with crafted prompts to ensure responses were relevant, accurate, and focused on HORSCH machinery documentation.

Performance timing was implemented using Python's time module, capturing execution duration for each component to enable system optimization.

C. plot-RAG (pRAG): A Novel Evaluation Framework

1) Motivation and Design: A key contribution of this work is pRAG, a novel visualization and evaluation framework specifically designed to address the lack of quantitative, interpretable evaluation methods for RAG systems. pRAG provides granular insights into the performance of individual RAG components, particularly the critical retrieval and re-ranking stages. This contrasts with current evaluation approaches, which often focus on end-to-end performance or rely on limited metrics like recall and precision, which are susceptible to outliers [23].

2) Visualization Components: The pRAG visualization (see Figure 1) displays multiple dimensions of system performance simultaneously:

- **Context position tracking:** Visualizes where relevant contexts from ground truth appear in both retrieval and re-ranked results (blue numbers).
- **Retrieval method comparison:** Distinguishes between embedding-based and BM25 keyword-based retrievals (y-axis).
- **Ground truth distribution:** Shows distances between relevant contexts in the document corpus (green numbers).
- **Quantitative metrics overlay:** Presents calculated performance metrics alongside visual representations (top right corner).
- **Relevant context count at position:** The number of relevant contexts from ground truth appearing at this specific retrieval position across the entire evaluated data set (numbers in parentheses).

3) Metrics Integration: pRAG calculates and visualizes several critical metrics:

- **Specialized recall metrics:**
  - Recall Emb: Effectiveness of embedding-based retrieval

  - Recall BM25: Performance of keyword-based retrieval
  - Recall Full Retrieval: Combined unique contexts retrieval rate
  - Recall Re-ranking from Retrieval: Preservation of relevant contexts after re-ranking
- **Ranking quality:** Normalized Discounted Cumulative Gain (NDCG) calculation highlighting the importance of positioning relevant information earlier in results
- **Retrieval optimization:** Recommended retrieval sizes for both embedding and BM25 components

4) Actionable Insights: The pRAG framework provides actionable insights by visually exposing:

- Which retrieval method (BM25 or embeddings) more effectively captures relevant contexts
- How effectively the re-ranker prioritizes relevant contexts
- Optimal retrieval configuration parameters
- Performance bottlenecks in specific components

This visualization approach enables the identification of system weaknesses without requiring extensive manual analysis, making it particularly valuable for ongoing RAG system development and optimization.

5) Knowledge Graph Integration into pRAG: Furthermore, we supplement the paper with a complementary concept for integrating knowledge graphs into pRAG. Each data point in the pRAG visualization represents one or more chunks contained in the ground truth. If each data point is interpreted as a node for one chunk or as nodes for more chunks, it is possible to draw edges between the data points to symbolize their relationships. In addition, their relationship is made interpretable via several technical components, which which must be interpreted relative to the specific technical components of each RAG implementation.

In pRAG, different chunks can be located at one position. The pRAG plot only shows that a chunk from the ground truth for the question-answer pair occurs there – but not which chunk it is. Since these chunks can now have different knowledge graphs to other chunks, exploration in the Z direction is necessary to display the semantic edges for each node. A 3D plot is not provided here because of the complex visualization.

For this evaluation approach we recommend to use uuids for each chunk row in the database to realize working evaluation without the viable ascii errors of text fields. In order to create a generic pRAG plot with the integration of knowledge graphs, the dataset should be structured in JSON format with the following keys:

JSON Keys for generic pRAG evaluation with knowledge graph combination:

- **user_query**: The query of the user asked to the RAG system
  Datatype: `string`
- **rag_answer**: The answer of the RAG system
  Datatype: `string`
- **ground_truth**: The ground truth of the answer. Can also be created in a synthetic way
  Datatype: `list[string]`
- **contexts**: A list of dicts. The keys in a dict represent the different technical retrieval, re-ranking, or other approaches to fetch chunks for the generation step
  Datatype: `list[dict]`
- **pair_of_contexts**: On the same gate, it is possible to use more than one technical component, like the BM25 and Embedding technique, in the retrieval step.
  Datatype: `list[list]`
- **knowledge_graph**: Displays the relation between edges via nodes as UUIDs
  Datatype: `dict[string, list]`

Figure 5 shows the utilization of the augmented pRAG approach in the analysis of retriever performance. The particular results are further discussed in Section III-E.

D. Synthetic Dataset Generation

For evaluation, both manually curated and synthetically generated question-answer pairs based on three technical manuals for products from the HORSCH portfolio: Avatar 12/40 SD, Joker RX, and Tiger MT were created. These documents were selected based on machine sales volume analysis, indicating likely user query subjects.

For each document, we prepared 50 question-answer pairs with relevant contexts as ground truth. From each set, five pairs were randomly selected as examples for synthetic generation. Using these examples iteratively with different ground truth contexts, we generated 45 synthetic question-answer pairs per document using three different language models: GPT-4o-Mini, Gemini-1.5-Flash, and Nemotron-4-340b-Instruct. Also, two comparison methodologies were implemented:

1) **Absolute comparison:** evaluating curated vs. synthetic datasets based on different contexts
2) **Relative comparison:** generating synthetic data using ground truth from the curated dataset

The concept involved comparing the synthetic data based on the same chunks that were used for the processed datasets. This approach is referred to as a relative comparison. The generator models generated synthetic question-answer pairs based on the chunks selected for the processed datasets. Various discriminant models were then applied to make a selection decision between the synthetic and processed pairs. The combinations of generator and discriminant models are shown in Table II.

In addition to the relative comparison, an absolute comparison was also performed. Here, random chunks were retrieved from the database. The synthetic datasets were then generated and an LLM evaluated the new data independently of the manually generated data.

Quality assessment employed a GAN-like approach, using language models (GPT-4o-Mini, Llama-3-Patronus-Lynx-8B-Instruct [24], and Prometheus-7b-v2.0 [25]) as discriminators to evaluate response quality with Pass/Fail determinations and comparative quality judgments.

E. Experimental Setup

Multiple experimental configurations are conceptualized to evaluate different aspects of the RAG system and demonstrate the utility of the pRAG framework. For enhanced retrieval configurations, we used a basic setup and changed specific technical components for enhanced setups:

a) Basic Synthetic Data Evaluation (Setup A):

- Generator: ChatGPT-3.5-Turbo-0125
- Retrieval: paradeDB (BM25+Embeddings)
- Re-ranker: Cross-encoder/msmarco-MiniLM-L6-en-de-v1
- Retrieval size: 8 contexts each for BM25 and embeddings
- Re-ranking size: 4 contexts

b) Enhanced Retrieval Configuration (Setup B):

- Generator: ChatGPT-3.5-Turbo-0125
- Metadata: Machine Name
- Re-ranker: Cross-encoder/msmarco-MiniLM-L6-en-de-v1
- Retrieval size: 60 contexts (BM25+Embeddings)
- Re-ranking size: 20 contexts

c) Enhanced Retrieval Configuration (Setup B-1):

- New Re-ranker: Cross-encoder/ms-marco-MiniLM-L-6-v2

d) Enhanced Retrieval Configuration (Setup B-2):

- New Method: HyDE Integration

For additional quantitative evaluation, we implemented the RAGAS framework to assess context precision, answer credibility, relevance, and accuracy. We compared RAGAS with the pRAG framework to substantiate the validity of the pRAG approach. We supplemented the pRAG approach with timing analysis of each system component, capturing minimum, maximum, and median execution times.

The expert evaluation was conducted with domain specialists who assessed question-answer pair quality in a blinded format, comparing synthesized and manually curated responses without knowledge of their origin to eliminate bias. For this experiment, we used the Basic Setup B with three different datasets.

In this comprehensive methodology using the novel pRAG evaluation framework, we aimed to evaluate not only the overall RAG system performance but also the viability of synthetic data for ongoing system improvement. We aimed to address the issue of available ground truth datasets by generating synthetic datasets automatically based on contexts from the database.

III. Results and Discussion

A. Performance Analysis of RAG Components Using pRAG

1) Retrieval Component Performance: The analysis demonstrates that each component contributes differently to the overall performance and can be individually assessed through visualization with pRAG. Figure 1 illustrates the pRAG visualization, where the positions of contexts in the ground truth collection are mapped against their retrieval positions. It shows that several relevant contexts (positions 6, 7, and 8) were missed by BM25 but captured by embedding-based retrieval. Substantial gaps in the diagram, particularly towards the tail of the retrieval list, can support decision-making on whether increasing the retrieval size at the cost of performance should be implemented to identify only a few additional relevant contexts. The automated evaluation of RAG systems with pRAG does not require an LLM as a judge. This makes the evaluation more resource-efficient, considering the substantial computational power required by LLMs.

Since pRAG visualizes the full set of retrieved contexts, the precision metric can be omitted. However, integrating the recall value into the diagram is beneficial to complement the visualization with a quantitative metric. The average values from setups in Section II-E b), c), and d) are presented in Table I.

TABLE I
Recall metrics across enhanced retrieval configuration

Metric	Setup B	Setup B-1	Setup B-2
Recall BM25	0.4314	0.4615	0.3654
Recall Embeddings	0.6863	0.6154	0.3846
Recall Full Retrieval	0.9412	0.9231	0.6923

The pRAG visualization enabled a dedicated evaluation of retrieval techniques, revealing that relevant contexts were retrieved either through BM25 or embedding-based methods and also in both. This points out the importance of evaluating different retrieval strategies individually, as relevant contexts may be identified in one approach but not in another. Further analysis demonstrated that embedding-based retrieval significantly outperformed lexical methods for datasets containing technical terminology. This underlines the necessity of hybrid retrieval approaches, where the combination of strategies ensures a more comprehensive retrieval process and improves overall performance.

2) Optimal Retrieval Size Determination: The experiments demonstrate a relationship between retrieval size and answer quality. With higher retrieval size RAG systems have to handle more irrelevant contexts. Therefore, the optimal retrieval size can be determined using:

$$\text{Retrieval Size} = \frac{\text{Current Retrieval Size} \times \text{Acceptable Response Time}}{\text{Average Response Time}}$$

This formula provides a practical guideline for balancing response time against completeness. As shown in Figure 1, there is no need to put the retrieval size to 60 because the latest relevant contexts were found in positions 21 by embeddings and position 48 by BM25.

The pRAG analysis revealed diminishing returns beyond certain retrieval sizes. For example, in setup B-2 (cf. Figure 1), increasing BM25 retrieval size from 28 to 48 yielded only one additional relevant context, suggesting a practical cut-off point based on efficiency considerations.

B. Synthetic Dataset Evaluation Results

1) Comparative Quality Assessment: To assess the effectiveness of synthetic versus manually prepared datasets, we evaluated both using specialized discriminator models. The results of this evaluation indicate that synthetically generated data achieves comparable or superior performance; specifically,

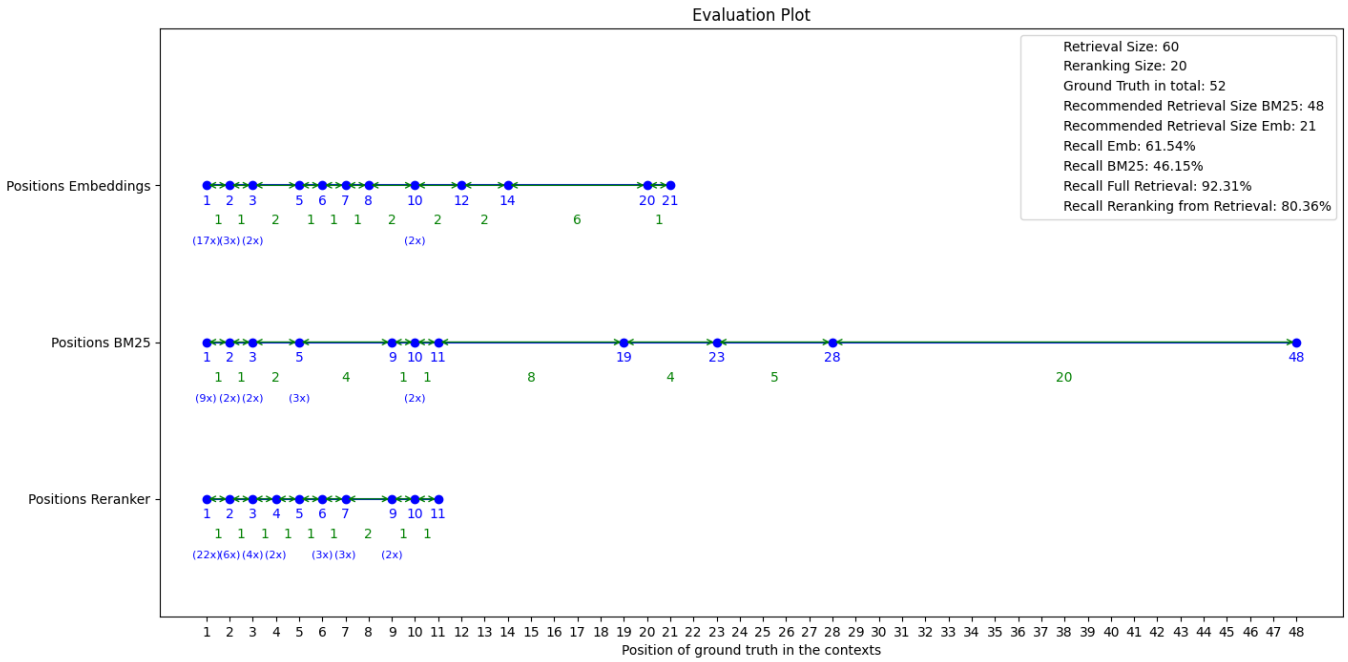

Figure 1. pRAG Visualization Showing Position of Retrieved Contexts Relative to Ground Truth based on Setup B-1.

manually prepared datasets did not offer a notable advantage, and Lynx even performed better on the synthetic data. This confirms that synthetic datasets can provide a similar level of performance to manually prepared ones.

In a direct comparison, the synthetic datasets were rated notably better by the discriminant model than the processed ones when comparing Figures 2 and 3. Additionally, GPT-4o Mini performed the worst, except in cases where it was evaluated directly by GPT-4o Mini. From this, it can be deduced that AI models evaluate their own synthetically generated datasets better and that dedicated AI models developed for synthetic data generation perform best.

Detailed performance results are presented in Figure 4.

TABLE II
GENERATOR-DISCRIMINATOR COMBINATIONS

No.	Generator - Discriminator Combination
1	GPT-4o Mini - GPT-4o Mini
2	Gemini-1.5-Flash - GPT-4o Mini
3	Nemotron-4-340b-Inst. - GPT-4o Mini
4	GPT-4o Mini - Llama-3-Patronus-Lynx-8B-Inst.
5	Gemini-1.5-Flash - Llama-3-Patronus-Lynx-8B-Inst.
6	Nemotron-4-340b-Inst. - Llama-3-Patronus-Lynx-8B-Inst.
7	GPT-4o Mini - Prometheus-7b-v2.0
8	Gemini-1.5-Flash - Prometheus-7b-v2.0
9	Nemotron-4-340b-Inst. - Prometheus-7b-v2.0

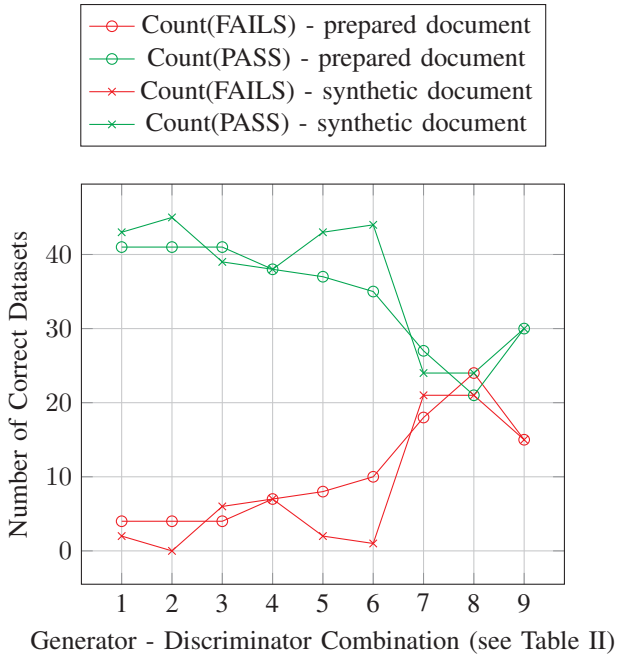

Figure 2. Comparison of Prepared and Synthetic Document for Different Generator-Discriminator Combinations.

2) Human Expert Validation: Human evaluators assessed pairs of question-answer examples from both dataset types. In 68% of comparable cases, experts preferred synthetically generated data, with the remaining 32% showing no clear preference. Table III summarizes these findings. Expert evaluators noted that synthetic datasets showed stronger logical coherence and clearer question formulation. However, they identified a consequential limitation: synthetic datasets generated from tabular data frequently contained factual errors or misinterpretations of numerical relationships, suggesting a specific weakness in

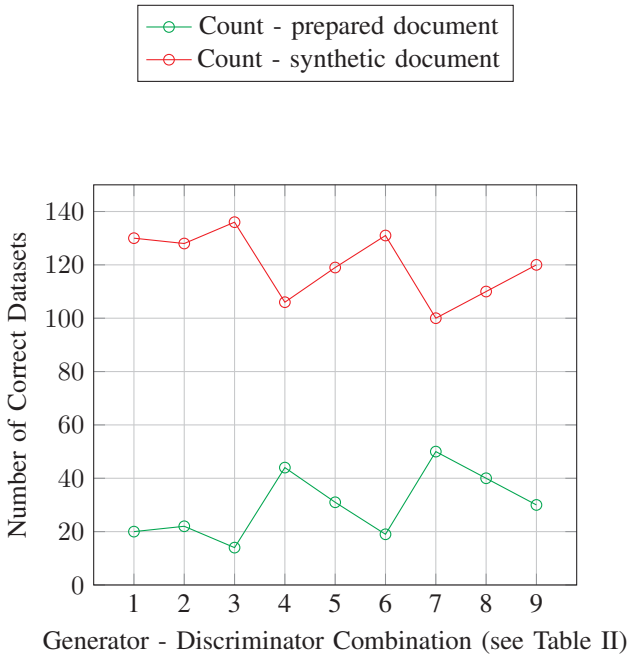

Figure 3. Comparison of Prepared and Synthetic Document for Different Generator-Discriminator Combinations.

current LLM approaches to tabular content. For the evaluation, we used setup B with the three different datasets. The model Nemotron was chosen for its ability to generate better synthetic data [26].

TABLE III
HUMAN EXPERT PREFERENCES IN DATASET EVALUATION

Dataset Pair	Prefer Prepared	Prefer Synthetic	No Preference
Avatar/Nemotron	14	20	16
JokerRX/Nemotron	8	9	31
TigerMT/Nemotron	4	24	22

C. Technical Component Performance Insights

1) Comparative Analysis of Retrieval Enhancements: We evaluated technical enhancements to the base RAG architecture, including re-ranking models and the integration of HyDE (Hypothetical Document Embeddings) [27]. This method decomposes dense retrieval into two distinct tasks: First, it uses an instruction-following language model (like InstructGPT) to generate a hypothetical document in response to a user query. In the second step, an unsupervised contrastively-trained encoder (like Contriever) encodes this hypothetical document into an embedding vector. This vector identifies a neighborhood in the corpus embedding space, from which similar real documents are retrieved based on vector similarity. A detailed component level comparison can be found in Figure 4 while Table IV summarizes these findings.

Contrary to [27], the HyDE approach shows reduced performance despite increased processing time. The pRAG analysis reveals that HyDE's theoretical advantage in generating better query representations did not improve the retrieval of relevant contexts in our test datasets.

TABLE IV
PERFORMANCE COMPARISON OF RETRIEVAL ENHANCEMENT TECHNIQUES

Technique	Recall Re-rank from Retrieval	NDCG	Mean Resp. Time (s)
msmarco-MiniLM-L6-en-de-v1	0.7544	0.54	3.41
ms-marco-MiniLM-L-6-v2	0.8036	0.63	4.08
HyDE Integration	0.7179	0.35	5.43

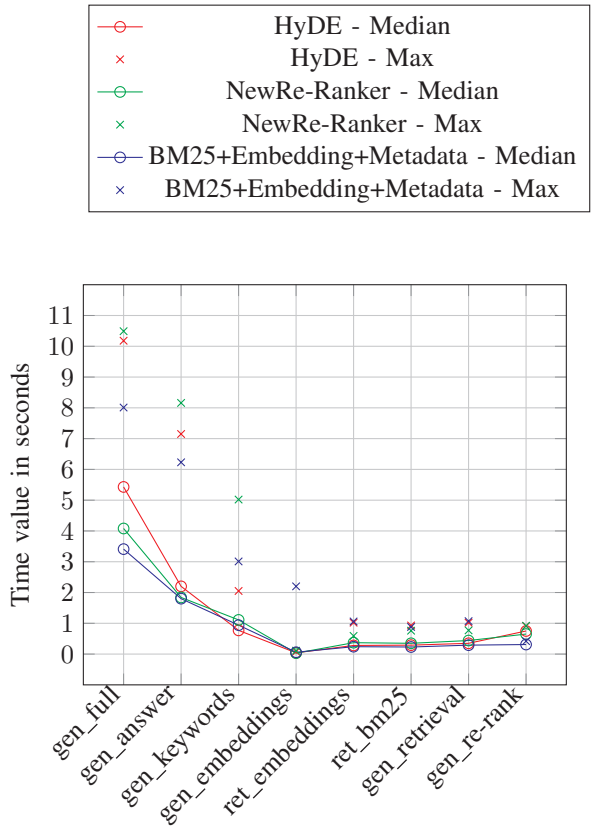

Figure 4. Time Performance across Different Components of the RAG System. Note the Time Advantage of BM25+Embedding compared to HyDE for Comparable Quality Metrics.

Among re-ranking models, ms-marco-MiniLM-L-6-v2 demonstrates the best performance with 80% recall from retrieval but required 20% more processing time than msmarco-MiniLM-L6-en-de-v1. The time-performance analysis shows this tradeoff across various system components.

D. Holistic approach for scaling and evaluating synthetic data

We created data ourselves in our synthetic data generation experiment. This process is not scalable and incurs a lot of human capacity. Rarely does a company invest this huge amount of time to generate prepared datasets. The results of the synthetic data experiments indicate a high usability and intrinsic ability to scale pRAG evaluations. Furthermore, they have the ability to use pRAG at any scale. Further experiments can determine how many synthetic dataset pairs can be generated

from a document, and how long the document should be, or how good it has to be to generate question-answer pairs.

Algorithm 1 Synthetic QA Generation and Evaluation Pipeline

Input:
  Big document D
  AI model for QA generation M_{QA}
  Discriminator model M_{disc}
  Retrieval-Augmented Generation system RAG
Output:
  Evaluation metrics comparing synthetic QA vs. RAG
  pRAG visualization
Step 1: Chunk the document
$chunks \leftarrow$ ChunkDocument(D)
Step 2: Generate synthetic QA pairs
for each c in $chunks$ **do**
    $QA\_pairs_c \leftarrow M_{QA}(c)$
end for
$QA\_pairs \leftarrow$ CombineAll(QA_pairs_c)
Step 3: Filter bad pairs
$Filtered\_QA \leftarrow []$
for each (q, a) in QA_pairs **do**
    if $M_{disc}(q, a) =$ good **then**
        Add (q, a) to $Filtered\_QA$
    end if
end for
Step 4: Generate RAG answers and save contexts
$RAG\_answers \leftarrow []$
for each (q, a) in $Filtered\_QA$ **do**
    $(rag\_answer, context) \leftarrow RAG(q)$
    Save $context$
    Add $(q, a, rag\_answer, context)$ to $RAG\_answers$
end for
Step 5: Evaluate
for each $(q, a, rag\_answer, context)$ in $RAG\_answers$ **do**
    Evaluate rag_answer against a
end for
Return:
  Evaluation metrics e.g NDCG, Recall, ..
  pRAG visualization

Our findings indicate that optimizing the retrieval component provides the greatest improvement in overall system performance. Specifically, the combination of lexical (BM25) and semantic (embedding) retrieval with optimized retrieval size offered the best balance of quality and performance across all test datasets.

E. Knowledge Graph Integration into pRAG Visualization

Within organizations, data is typically stored across a diverse range of systems. It is often the case that no single system contains a comprehensive documentation of a given topic [28]. To obtain such a complete record, the evolution of information maturity must be tracked across these multiple systems.

The development of an individual piece of information can be represented as a directed graph, where nodes correspond to information states and edges represent transitions or transformations between them. For this representation, organizations must have an understanding of how information propagates through their internal processes and how its quality changes throughout this trajectory.

When retrieval encounters directed information, reasoning mechanisms can be applied. This enables dynamic model selection in response to encountering interconnected, directed graphs. In contrast, undirected connections can be interpreted as supplementary relationships and can be incorporated post hoc from the database to support query answering.

Additionally, some connections may point to data points, or "chunks",that are not part of the established ground truth (shown as orange edges in Figure 5). We propose two potential strategies for handling such cases. First, the chunk can be included in the reasoning process under the assumption that the ground truth may be incomplete. Alternatively, the chunk can be ignored if it is not part of the ground truth. Since ground truth is often unavailable for novel queries, making a definitive distinction between these approaches can be challenging.

When graphs are integrated into pRAG, it is important to recognize that a single data point, or node, may contain multiple chunks. The number of chunks associated with a node is directly dependent on the number of queries incorporated into pRAG for testing.

If a data point has numerous incoming and outgoing connections, it becomes necessary to explicitly indicate which chunks are responsible for establishing these connections. Consequently, pRAG diagrams can become difficult to interpret when applied to large datasets.

However, this complexity can be managed effectively. By assigning weights to individual edges and considering the number of incoming and outgoing connections, nodes can be reordered to enhance clarity. This approach enables a synergistic combination of the advantages of RAG and Knowledge Graph methodologies, optimizing both information retrieval and reasoning capabilities within pRAG.

The starting point of an outgoing connection should always correspond to the last step immediately preceding the generation of a response by the LLM. If additional connections are incorporated into the context prematurely during retrieval, before they are properly retrieved or ranked, these connections may subsequently be deprioritized by a re-ranking model. As a result, the retrieval effort for these connections becomes effectively wasted, increasing inference cost without contributing meaningfully to the generated answer.

IV. Evaluation and Discussion

This section analyzes the empirical and methodological implications of our results, with a particular focus on evaluation reliability, scalability, and system-level insights for RAG architectures.

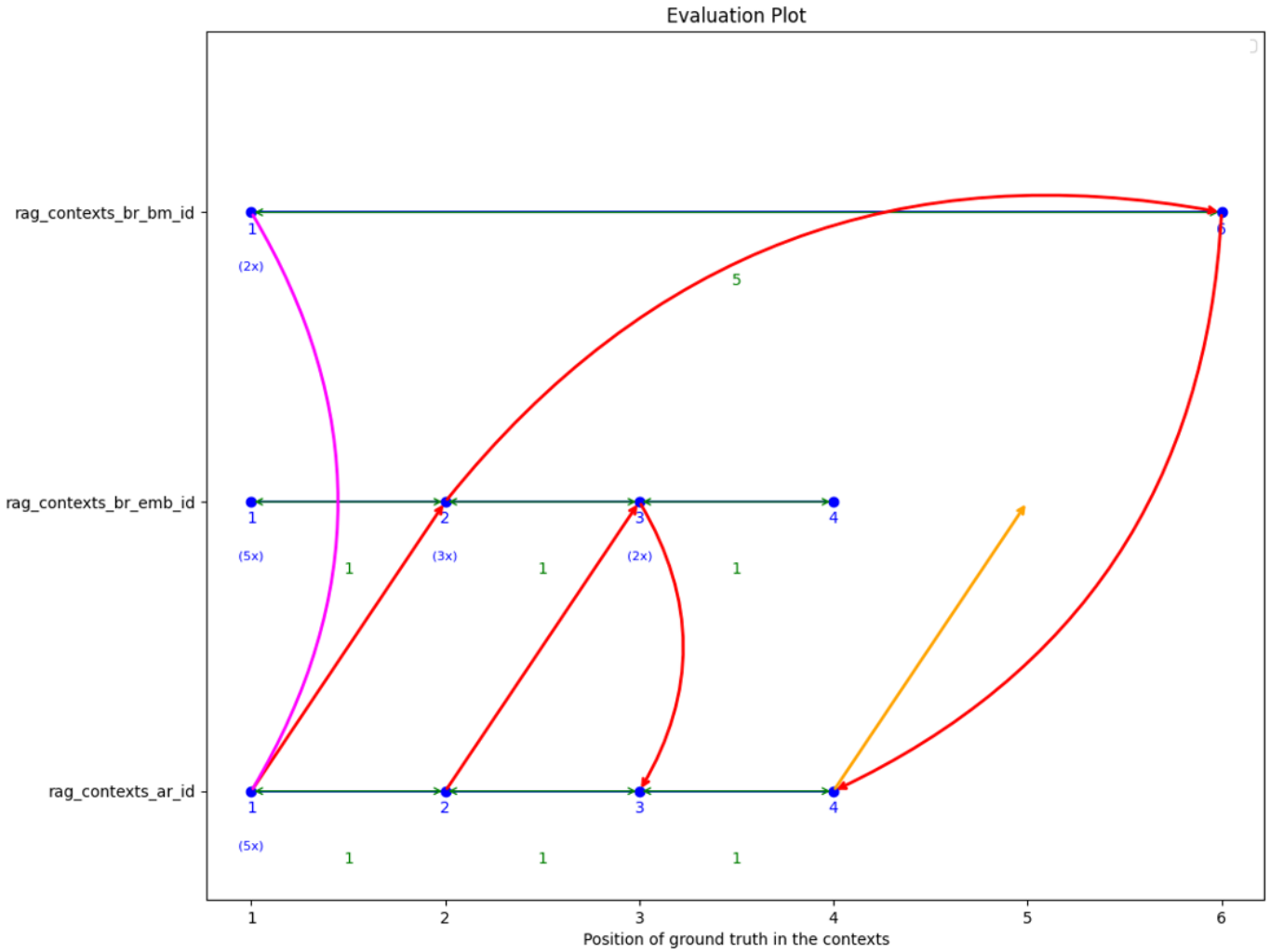

Figure 5. pRAG Visualization Augmented with Knowledge Graph Edges.

A. Role of Ground Truth in RAG Evaluation

A central finding of this work is the decisive role of high-quality ground truth data in evaluating RAG systems. Across all experiments, evaluation outcomes were highly sensitive to the completeness, consistency, and domain alignment of the ground truth. Incomplete or weakly aligned ground truth led to misleading performance signals, particularly in retrieval-heavy configurations.

This highlights a fundamental limitation of many existing evaluation approaches: without reliable reference data, even sophisticated metrics fail to provide meaningful insights. Consequently, evaluation quality is bounded less by metric design and more by the availability of robust ground truth.

B. Effectiveness of pRAG for Component-Level Analysis

The introduction of pRAG enables a shift from aggregated evaluation metrics toward fine-grained, interpretable diagnostics. Our experiments show that pRAG makes it possible to isolate retrieval and re-ranking behaviors, revealing performance bottlenecks that remain hidden in traditional evaluation pipelines.

In particular, pRAG provides:

- Detailed visibility into retrieval depth and ranking quality
- Identification of redundant or low-impact retrieved chunks
- Temporal and structural insights into retrieval dynamics

Compared to LLM-based evaluation methods, pRAG offers greater reproducibility, significantly lower computational cost, and improved transparency. This makes it particularly suitable for iterative system development and benchmarking.

C. Synthetic Data as a Scalable Evaluation Resource

Our results demonstrate that synthetic datasets can serve as a viable substitute for manually curated evaluation data. The generated question-answer pairs achieved high semantic alignment with source contexts and were consistently rated as coherent by both automated discriminators and human evaluators.

This has important implications:

- Evaluation pipelines can be scaled without proportional increases in human annotation effort
- Domain-specific datasets can be generated rapidly, enabling continuous evaluation
- Methodological consistency improves due to standardized generation processes

However, limitations remain in handling structured and numerical data, suggesting that future improvements in generation techniques are required.

D. Retrieval as the Dominant Performance Factor

A consistent observation across all configurations is that retrieval quality dominates overall system performance. Variations in retrieval size, ranking strategy, and corpus structure had a significantly greater impact on results than downstream generation differences.

This finding reinforces the importance of:

- Careful tuning of retrieval size
- Balancing completeness against latency constraints
- Avoiding excessive retrieval that introduces noise

Based on our analysis, we derive practical guidelines for selecting retrieval configurations that optimize the trade-off between response quality and computational efficiency.

E. Towards Graph-Enhanced Evaluation

We further explored the integration of knowledge graph concepts into pRAG. Modeling retrieved chunks as nodes and their relationships as edges enables richer representations of information flow within RAG systems.

This perspective provides:

- Improved interpretability of complex retrieval interactions
- Opportunities for graph-based prioritization and reasoning
- Enhanced handling of incomplete or distributed ground truth

While still conceptual, this extension indicates a promising direction for unifying retrieval evaluation with structured knowledge representations.

F. Implications for RAG System Design

Taken together, our findings suggest that effective RAG system development requires:

- Strong emphasis on retrieval optimization
- Scalable and automated evaluation pipelines
- Interpretable tools for diagnosing component-level behavior

The combination of pRAG and synthetic datasets provides a practical foundation for achieving these goals, enabling systematic and data-driven improvement of RAG architectures.

V. Conclusion and Future Work

This work presents a scalable and interpretable framework for evaluating RAG systems by combining pRAG visualization with synthetic dataset generation. Our results demonstrate that reliable evaluation depends fundamentally on high-quality ground truth, and that synthetic data can effectively address this bottleneck at scale.

The proposed pRAG approach enables fine-grained, component-level analysis of retrieval and re-ranking behavior without relying on LLM-based evaluation, improving both transparency and reproducibility. In combination with synthetic datasets, this establishes a practical and resource-efficient methodology for iterative evaluation and optimization of RAG systems.

Our findings further confirm that retrieval is the dominant factor influencing overall system performance, emphasizing the need for careful configuration of retrieval strategies under latency constraints. The integration of graph-based representations offers additional potential for enhancing interpretability and reasoning.

Future work should focus on dynamic evaluation of evolving systems, extensions to multi-modal data, improved synthetic data generation for structured content, and deeper analysis of retrieval–generation interactions. Additionally, graph-aware optimization and advanced visualization techniques remain promising directions for improving both performance and interpretability.

Overall, this framework provides a robust foundation for the systematic development and evaluation of RAG systems in complex, domain-specific environments.

References

[1] A. Kreß, A. Lawall, and T. Zöller, "Retrieval Performance in RAG Systems: A Component-Level Evaluation Framework," in *The Second International Conference on Generative Pre-trained Transformer Models and Beyond (GPTMB) 2025*, 2025, pp. 3–8, [retrieved: January, 2026]. [Online]. Available: https://www.thinkmind.org/library/GPTMB/GPTMB_2025/gptmb_2025_1_20_30012.html

[2] X. Wang *et al.*, "Searching for best practices in retrieval-augmented generation," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 17 716–17 736.

[3] H. Yu *et al.*, "Evaluation of retrieval-augmented generation: A survey," in *CCF Conference on Big Data*. Springer, 2024, pp. 102–120.

[4] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, "Seven failure points when engineering a retrieval augmented generation system," in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, 2024, pp. 194–199.

[5] A. Salemi and H. Zamani, "Evaluating retrieval quality in retrieval-augmented generation," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 2395–2400.

[6] T. Kenneweg, P. Kenneweg, and B. Hammer, "Retrieval augmented generation systems: Automatic dataset creation, evaluation and boolean agent setup," *arXiv preprint arXiv:2403.00820*, 2024, [retrieved: May, 2025].

[7] Z. Yang *et al.*, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," *arXiv preprint arXiv:1809.09600*, 2018, [retrieved: May, 2025].

[8] D. F. Campos *et al.*, "Ms marco: A human generated machine reading comprehension dataset," *ArXiv*, vol. abs/1611.09268, 2016, [retrieved: May, 2025]. [Online]. Available: https://api.semanticscholar.org/CorpusID:1289517

[9] K. Zhu *et al.*, "Rageval: Scenario specific rag evaluation dataset generation framework," *arXiv preprint arXiv:2408.01262*, 2024, [retrieved: May, 2025].

[10] R. Liu *et al.*, "Best practices and lessons learned on synthetic data," *arXiv preprint arXiv:2404.07503*, 2024, [retrieved: May, 2025].

[11] S. Kim *et al.*, "Evaluating language models as synthetic data generators," *arXiv preprint arXiv:2412.03679*, 2024, [retrieved: May, 2025].

[12] S. Es, J. James, L. E. Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2024, pp. 150–158.

[13] Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, 2023, [retrieved: May, 2025].

[14] I. Mirzadeh *et al.*, "Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models," *arXiv preprint arXiv:2410.05229*, 2024, [retrieved: May, 2025].

[15] Y. Yu *et al.*, "Rankrag: Unifying context ranking with retrieval-augmented generation in llms," *Advances in Neural Information Processing Systems*, vol. 37, pp. 121 156–121 184, 2024.

[16] G. d. S. P. Moreira *et al.*, "Enhancing q&a text retrieval with ranking models: Benchmarking, fine-tuning and deploying rerankers for rag," *arXiv preprint arXiv:2409.07691*, 2024, [retrieved: May, 2025].

[17] P. Mandikal and R. Mooney, "Sparse meets dense: A hybrid approach to enhance scientific document retrieval," *arXiv preprint arXiv:2401.04055*, 2024, [retrieved: May, 2025].

[18] Y. Lu, M. Shen, H. Wang, X. Wang, C. van Rechem, T. Fu, and W. Wei, "Machine learning for synthetic data generation: a review," *arXiv preprint arXiv:2302.04062*, 2023, [retrieved: May, 2025].

[19] S. Robertson, H. Zaragoza *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[20] Hugging Face, "Model card for multi-qa-minilm-l6-cos-v1," [Online]. Available: https://huggingface.co/sentence-transformers/multi-qa-MiniLM-L6-cos-v1, 2021, [retrieved: May, 2025].

[21] ——, "Model card for msmacro-minilm-l6-en-de-v1," [Online]. Available: https://huggingface.co/cross-encoder/msmarco-MiniLM-L6-en-de-v1, 2021, [retrieved: May, 2025].

[22] ——, "Model card for ms-macro-minilm-l-6-v2," [Online]. Available: https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2, 2021, [retrieved: May, 2025].

[23] D. Park and S. Kim, "Probabilistic precision and recall towards reliable evaluation of generative models," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 20 099–20 109.

[24] S. S. Ravi, B. Mielczarek, A. Kannappan, D. Kiela, and R. Qian, "Lynx: An open source hallucination evaluation model," *arXiv preprint arXiv:2407.08488*, 2024, [retrieved: May, 2025].

[25] S. Kim *et al.*, "Prometheus 2: An open source language model specialized in evaluating other language models," *arXiv preprint arXiv:2405.01535*, 2024, [retrieved: May, 2025].

[26] B. Adler *et al.*, "Nemotron-4 340b technical report," *arXiv preprint arXiv:2406.11704*, 2024, [retrieved: May, 2025].

[27] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels, 2022," *URL https://arxiv. org/abs/2212.10496*, 2022, [retrieved: May, 2025].

[28] M. Lenzerini and C. Daraio, *Challenges, Approaches and Solutions in Data Integration for Research and Innovation*. Cham: Springer International Publishing, 2019, pp. 397–420. [Online]. Available: https://doi.org/10.1007/978-3-030-02511-3_15