AI

MDPI

Article

Swamped with Too Many Articles? GraphRAG Makes Getting Started Easy

Joëd Ngangmeni *,† **and Danda B. Rawat** *,†

Department of Electrical Engineering and Computer Science, Howard University, 2400 6th St NW, Washington, DC 20059, USA

* Correspondence: joed.ngangmeni@bison.howard.edu (J.N.); danda.rawat@howard.edu (D.B.R.)

† These authors contributed equally to this work.

Abstract: Background: Both early researchers, such as new graduate students, and experienced researchers face the challenge of sifting through vast amounts of literature to find their needle in a haystack. This process can be time-consuming, tedious, or frustratingly unproductive. Methods: Using only abstracts and titles of research articles, we compare three retrieval methods—Bibliographic Indexing/Databasing (BI/D), Retrieval-Augmented Generation (RAG), and Graph Retrieval-Augmented Generation (GraphRAG)—which reportedly offer promising solutions to these common challenges. We assess their performance using two sets of Large Language Model (LLM)-generated queries: one set of queries with context and the other set without context. Our study evaluates six sub-models—four from Light Retrieval-Augmented Generation (LightRAG) and two from Microsoft's Graph Retrieval-Augmented Generation (MGRAG). We examine these sub-models across four key criteria—comprehensiveness, diversity, empowerment, and directness—as well as the overall combination of these factors. Results: After three separate experiments, we observe that MGRAG has a slight advantage over LightRAG, naïve RAG, and BI/D for answering queries that require a semantic understanding of our data pool. The results (displayed in grouped bar charts) provide clear and accessible comparisons to help researchers quickly make informed decisions on which method best suits their needs. Conclusions: Supplementing BI/D with RAG or GraphRAG pipelines would positively impact the way both beginners and experienced researchers find and parse through volumes of potentially relevant information.

Keywords: GraphRAG; LightRAG; retrieval-augmented generation; graph; large language model; LLM; RAG

Received: 11 January 2025
Revised: 18 February 2025
Accepted: 25 February 2025
Published: 1 March 2025

Citation: Ngangmeni, J.; Rawat, D.B. Swamped with Too Many Articles? GraphRAG Makes Getting Started Easy. *AI* **2025**, *6*, 47. https://doi.org/10.3390/ai6030047

1. Introduction

Conventional reconnaissance methods (e.g., reading full articles, skimming documents, or relying on summaries) are indirect, uncertain, or time-consuming. They rarely provide quick answers to specific questions readers have. Moreover, the growing volume of available information can quickly become overwhelming and discourage the interest of potential stakeholders.

Contemporary researchers use online platforms—Semantic Scholar, Google Scholar, Arxiv, Connected Papers, CitationGeco, etc.—to source material. Some software no longer gets updated and therefore cannot keep up with current state of the art [1]. Other methods which rely solely on Bibliographic Indexing/Databasing (BI/D) use bibliographic information to proxy article content [2]. While proxies like citation count can help minimize the risk of falsely elevating importance, they introduce the potential for irrelevant citations,

adding an element of uncertainty to BI/D. Though far more efficient than searching through physical libraries for documents, these platforms still retrieve and present full documents in response to use queries, leaving researchers to review the content of entire articles before understanding their true relevance. Making vast amounts of relevant information available, alone, is not enough.

Simply put, these platforms are designed to identify and present relevant sources, but cannot directly answer user queries. As such, they do not fully address researcher concern about time consumption or eased comprehension. We explore methods that enhance the retrieval of key information and present it in ways that directly and clearly answer user queries. Section 2 goes over background information and presents our models of interest. Experimental setup and methodology are described in Section 3. Summarized test results are presented as graphs that quickly allow readers to asses basic strengths and trade-offs of current state-of-the-art models in Section 4. Discussion and future work are covered in Section 5 followed by a brief conclusion in Section 6.

2. Literature Review

2.1. Retrieval Background

Advancements in Artificial Intelligence (AI) and Machine Learning (ML) empower researchers to tackle increasingly challenging problems. The field of Natural Language Processing (NLP), for example, a subset of AI/Machine Learning (ML), is geared toward developing models that process and generate natural (human) language. Approaches to this problem are varied, but have evolutionarily led to improved capabilities [3]. Early statistical models calculated the probability of sentences occurring [4]. As computational capabilities evolved and became more widely available, language models began leveraging the Neural Network (NN) to calculate probabilities of words and groups of words occurring in sentences [5,6]. Eventually, they started being pre-trained to compute more specialized results, paving the way to the modern-day Large Language Model (LLM).

These enhanced capabilities allowed researchers to use Large Language Model (LLM) for retrieving query-specific text from large corpora of information. LLMs sometimes generate false information [7], but since it is becoming increasingly challenging to manually check their output as data pool sizes increase, using LLMs to judge other LLM output is now an open area of research [8]. This, of course, is not without its own concerns; however, prompting LLMs to provide supporting evidence and references alongside generated output improves researcher confidence in their output [9,10]. Figure 1 displays sample output from one such model known as a Retrieval-Augmented Generation (RAG) pipeline.

2.2. Retrieval-Augmented Generation

The idea behind a Retrieval-Augmented Generation (RAG) pipeline is fairly straightforward. Data are divided into chunks, embedded in a certain way [11], and stored in a data container—typically a vector store—that is made accessible to some retrieval engine. When a query is submitted, it is immediately embedded using the same embedding method as the chunks. The retrieval engine then conducts a similarity search on both query and chunk embeddings to identify the most relevant chunks. These chunks are combined with the original query to create a context package, which is then sent to the LLM. With this context package—the query and its matching data chunks—the LLM can generate a more informed/relevant response [12]. RAG eliminates the need for LLMs to be trained or re-trained on domain-specific information, unlocking their potential and making them applicable for contemporary work.

```
User Query:
What are the top 5 themes in the data?

RAG Response:
Based on the analysis of the dataset, the~following are the
   top five themes identified: ### 1. **Integration of AI
   Technologies** The integration of artificial intelligence (
   AI) across various sectors, including healthcare, education
   , and~legal advisory services, is a prominent theme. This
   integration highlights the transformative role of AI in
   enhancing operational efficiency, decision-making, and~user
    interactions. Notably, frameworks like Retrieval-Augmented
    Generation (RAG) and models such as ChatGPT are emphasized
    for their applications in improving service delivery and
   accuracy in various fields [Data: Reports (2496, 2017,
   2827, 3326, 662)]. ### 2. **Advancements in Large Language
   Models (LLMs)** The development and optimization of large
   language models (LLMs) are central to many discussions.
   These models are being integrated into applications ranging
    from healthcare to customer service, showcasing their
   versatility and significance in enhancing communication and
    information retrieval capabilities. The~focus on
   frameworks like RAG and methodologies for fine-tuning LLMs
   indicates a strong interest in improving their performance
   and reliability [Data: Reports (2014, 2529, 2140, 2217,
   3504)]...
```

Figure 1. Snippet of a RAG model's response.

Unfortunately, the flat data representations used by models like RAG limit their ability to retrieve information dependent on understanding complex entity relationships [10]. Typical pipelines employ vector search methods that compare query embeddings to data chunk embeddings, but lack appreciation of data scope and/or context. Naïve RAG struggles with contextual awareness and is unable to adequately answer questions that require significant logical interpellation to connect query to context. After incorporating the necessary information into a data store, researchers query a naïve RAG pipeline with prompts that require semantical understanding rather than syntactic similarity of query and data [9]. In such cases, naïve RAG failed to provide insightful responses. Other researchers studying the same challenge explain that naïve RAG pipelines may struggle to synthesize information into relevant responses even if the proper information is retrieved [10].

2.3. Other Retrieval Approaches

There are countless variations of recognition/retrieval algorithms and RAG pipeline customizations. Sematic Scholars, for example, is a free AI-powered algorithm for scientific literature. It uses an arsenal of NNs to break sentences into groups of tokens (spans) on which they perform entity and relationship extraction [13]. Then, supplementing extraction with bibliographic metadata and other information (i.e., title, authors, bibliography years, etc.), it generates a knowledge graph. The graph, in response to user queries, is presented as a list of articles. There are two key differences between this and RAG approaches. Firstly, RAGs do not immediately use NNs to perform entity and relationship extraction, they use a combination of LLMs and tactical prompting. Secondly, though it considers a wide range of information, Semantic Scholars' functionality is limited to presenting wholeresearch articles. RAG pipelines provide direct answers to the user. The RoBERTa Multi-scale CNN BiGRU Self-attention CRF (RMBC) model extracts named entities from text sources. This

approach differs from RAG in that it focuses on Chinese medical entities and functions through a combination of NNs, self-attention, and conditional random fields [14].

Some retrieval approaches work by retrieving groups of health-related data and presenting them to users, like [15]. Retrieval-Augmented FineTuning (RAFT), a RAG enhancement that enables them to ignore "distractor" documents when generating answers, also leads to more efficient generation in some cases [16]. The Hypothetical Document Embeddings (HyDE) method translates user queries into LLM-generated documents before embedding and using those generated documents for vector search instead of the original query [17]. Despite this and other customizations, like multi-step question decomposition [18] or step-back reasoning [19], naïve RAGs and models that rely on flat data representations are said to be limited [9,10]. Some queries require a more holistic understanding of the dataset [9].

2.4. Graph Retrieval-Augmented Generation

The benefits of chunking observed in RAG are even more pronounced in Graph Retrieval-Augmented Generation (GraphRAG) where chunking is applied at its most granular level to retrieve entities and relationships. "When handling large token counts and complex queries that require a thorough understanding of the dataset's context, graph-based systems... consistently outperform purely chunk-based retrieval methods such as [RAG] and HyDE" [10]. These models gain an enhanced ability to analyze overall context and content of information since their data representations are no longer flat. They also improve RAG performance on Query Focused Summarization (QFS) tasks [9], outperforming other methods on questions that cannot be made more "retriever-friendly" through customizations. Examples of such questions can be found in Figure 2.

```
1. Provide an overview of the entire dataset.
2. What are the key patterns or trends in the data over time?
3. What are the 5 most mentioned topics in this dataset. How
   many times is each mentioned and where?
4. How do different categories or groups in the data compare
   against each other?
5. Can you summarize the key insights across different
   subgroups in the data?
```

Figure 2. Sample of experiment 3 questions.

We now shift to testing different RAG implementations for a specific use case—analyzing large volumes of research articles and presenting results in a palatable way. Our focus is on two RAG implementations: Light Retrieval-Augmented Generation (LightRAG) and Microsoft's Graph Retrieval-Augmented Generation (MGRAG). While MGRAG augments the retrieval process with varying levels of summaries of entity and relationship information, Light Retrieval-Augmented Generation (LightRAG) concentrates on retrieving entities and relationship information while reducing retrieval overhead from the community-based traversal method used in MGRAG [10].

3. Materials and Methods

The project scope includes five objectives:

1. To apply RAG and GraphRAG to answer subject matter-specific queries using only research article abstracts and titles.
2. To compare RAG and GraphRAG sub-model performance on five criteria (comprehensiveness, diversity, empowerment, directness, overall).

3. To identify best RAG or GraphRAG sub-model for answering subject matter specific queries when context is provided.
4. To identify the best RAG or GraphRAG sub-model for answering subject matter-specific queries when context is notprovided.
5. To evaluate RAG and GraphRAG sub-model performance on queries that require semantical understanding of the entire data pool.

3.1. Criteria Definition

The following criteria definitions were provided to all employed judge LLMs:

- Comprehensiveness: the extent to which the answer addresses all relevant aspects of the question, providing a thorough and complete understanding of the topic.
- Diversity: the range of perspectives, insights, and/or approaches included in the answer, offering a multifaceted view of the subject.
- Empowerment: how effectively the answer equips the reader with the knowledge and/or tools needed to make informed decisions or form well-rounded judgments on the topic.
- Directness: the clarity and conciseness with which the answer communicates the core message, avoiding unnecessary complexity or digression.
- Overall: how well an answer fulfills the requirements of all criteria; its cumulative performance over all four criteria.

3.2. Data Collection

Using the following keywords in the Semantic Scholars API, we collect 3,525 research article titles and abstracts: “RAG LLM”, “Retrieval Augmented Generation”, “graphRAG”, “graph RAG”, “graph retrieval augmented generation”, “graph rag llm”, “Retrieval-Augmented Generation”, “Retrieval Model”, “Large Language Model”, and “Large-Language Model”. Our sample size (3500+ articles) may be considered small for an LLM, but a human would struggle to read through and understand such a large body of work.

3.3. Query Generation

Since the data pool would take too long to manually generate questions from, we give an arbitrary LLM, a gpt-4o-miniinstance, a description of the data for it to generate test questions. This aligns with contemporary best practice [9,10]. The LLM does this by first generating 5 potential users for the dataset. It then generates 5 tasks for each user. Finally, it generates 5 questions per task. The whole process results in the generation of a set of 125 questions. We adapt this set into 2 separate pools, each containing 125 queries. Set 1 contains a user, task, and question, forming our “with context” pool, while Set 2 contains only the question, serving as our “without context" pool. For example, a query from set 1 could be “User is a Data Scientist whose task is to implement data exploration techniques to recommend relevant datasets. Their question is: How does the integration of data analysis and data exploration foster innovation in machine learning applications?”. The complementary—without context—set 2 query would be “How does the integration of data analysis and data exploration foster innovation in machine learning applications?”.

Both sets of queries are necessary when examining RAG pipeline performance because model response is dependent on context. Vector similarity in naïve RAG directly compares query embeddings to document chunk embeddings. Queries with context are embedded differently than those without context. It naturally follows that different document chunks can match each type of query. Once generated, we pass the queries into six different sub-models (four LightRAG sub-models and two MGRAG sub-models) and catalog their responses.

3.4. Hypothesis

We hypothesize that hybrid LightRAG will perform the best in the "overall" criteria due to its innovative implementation of high- and low-level information retrieval [10]. We believe the ability to synthesize high- and low-level information will aid in answering questions that require semantical comprehension.

3.5. Experiment 1

All experiments follow the same general flow (illustrated in Figure 3) with minute changes to address specific objectives. In experiment 1, we simultaneously send all six sub-model responses to a particular query into an arbitrary gpt-4o-mini instance in the following order: l_s_n, l_s_g, l_s_l, l_s_h, m_s_g, m_s_l. The first character of each name in this nomenclature indicates LightRAG or MGRAG and the last character indicates the type of sub-model: naïve, local, global, or hybrid. The middle character is irrelevant for the purposes of this article. The LLM's instructions are to rank answers from best to worst, making sure it provides reasoning for each ranking in the provided JSON format.

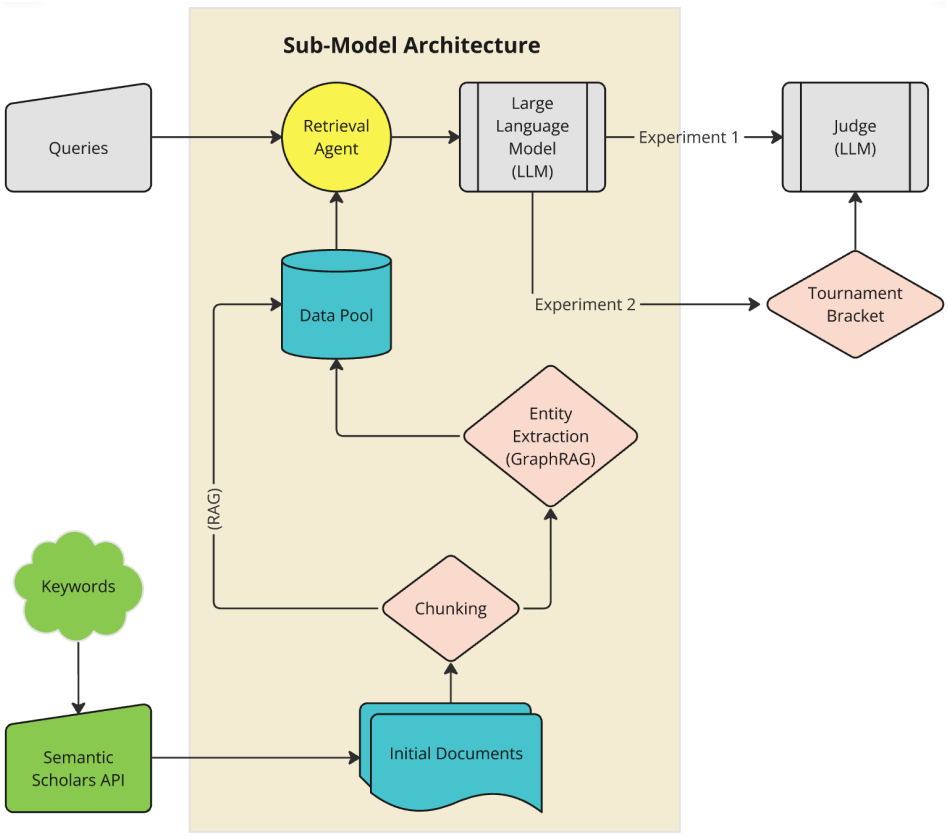

Figure 3. This figure shows a flowchart of our experimental method. Queries are passed to a retrieval agent that attempts to find relevant document chunks before passing everything to the LLM for evaluation. Experiment 3 is not explicitly displayed because though it goes through the tournament bracket, it differs at the creation of queries. Visit Section 3.7 for more information on the implementation of experiment 3.

3.6. Experiment 2

Experiment 2 introduces a randomized tournament (see Figure 4) to diminish bias in evaluation from experiment 1. After answers for a specific query are generated by all sub-models, before being sent to the judge LLM, their order is randomized to obtain an initial seeding. A full run through the bracket indicates the judging of a singular question's six answers. Bracket steps are repeated for all questions, with and without context. A loser's

bracket is included to ensure that all query responses have a chance of being tested for each attribute.

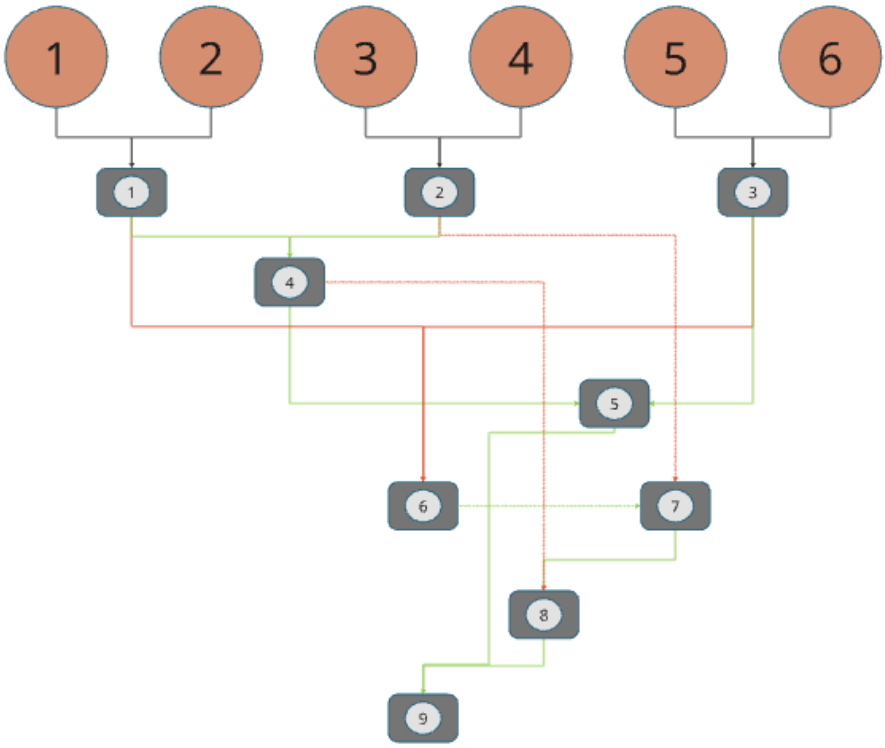

Figure 4. This figure shows the bracket used to test 3 pairs of answers per query. Each answer (from a sub-model) has a randomized seed, 1, 2, 3, 4, 5, or 6. Seeds 1 and 2 face each other in round 1, followed by seeds 3 and 4 in round 2, and seeds 5 and 6 in round 3. The winner of round 1 moves to round 4 where it faces the winner of round 2. The winners from round 4 and 3 face each other in round 5. Round 6 starts the losers bracket, having the loser of round 1 face the loser of round 3. The winner of this matchup moves to round 7 where it faces the loser of round 2. Round 8 features a matchup between the winner of the losers bracket and the loser of round 4. The winner of this round faces the winner of the winners bracket in the final round, 9. The color red denotes a loss while green highlights a win.

Experiment 2 also removes language that could add bias from the judge LLM's prompt. Rather than referring to the answers as "Answer 1" or "Answer 2", we refer to them by the string representation of the answer's origin from experiment 1—l_s_n, l_s_g, l_s_l, l_s_h, m_s_g, or m_s_l. With updated instructions and output format, experiment 2 generates more detailed data files. Each file only contains the comparison result of two sub-model answers to the same question, evaluated on a specific criterion, rather than comparing all six answers at once, as in experiment 1. This method also produces more comparisons per context pool—9 rounds per question $\times$ 125 questions $\times$ 1 file per round = 1136 files comparisons instead of the 126 comparisons of experiment 1.

3.7. Experiment 3

A task like "provide an overview of the entire dataset" is difficult for syntactic retrievers because they search for document chunks whose vector embeddings closely match the phrase "provide an overview of the entire dataset" rather than searching for chunks that respond to that phrase as if it were a question. Enhancements like HyDE, for example, cannot help this use case. HyDE-generated documents for such a query would simply talk about summarizing datasets rather than actually summarizing the dataset. Experiment 3 questions are hand-designed—instead of being LLM-generated, like those of

experiments 1 and 2—to be difficult to enhance with RAG customizations. These support the requirement for an elevated degree of semantical understanding for retrieval. We repeat the same steps from the other experiments to fairly test all six sub-models and rank their responses. Samples of the hand-crafted questions are displayed in Figure 2.

4. Results

We cataloged results from the judge LLM's responses and counted model wins. The bars in Figures 5–7 represent an aggregate count of each model's victories per criteria. Experiment 2 and 3's results are more succinct.

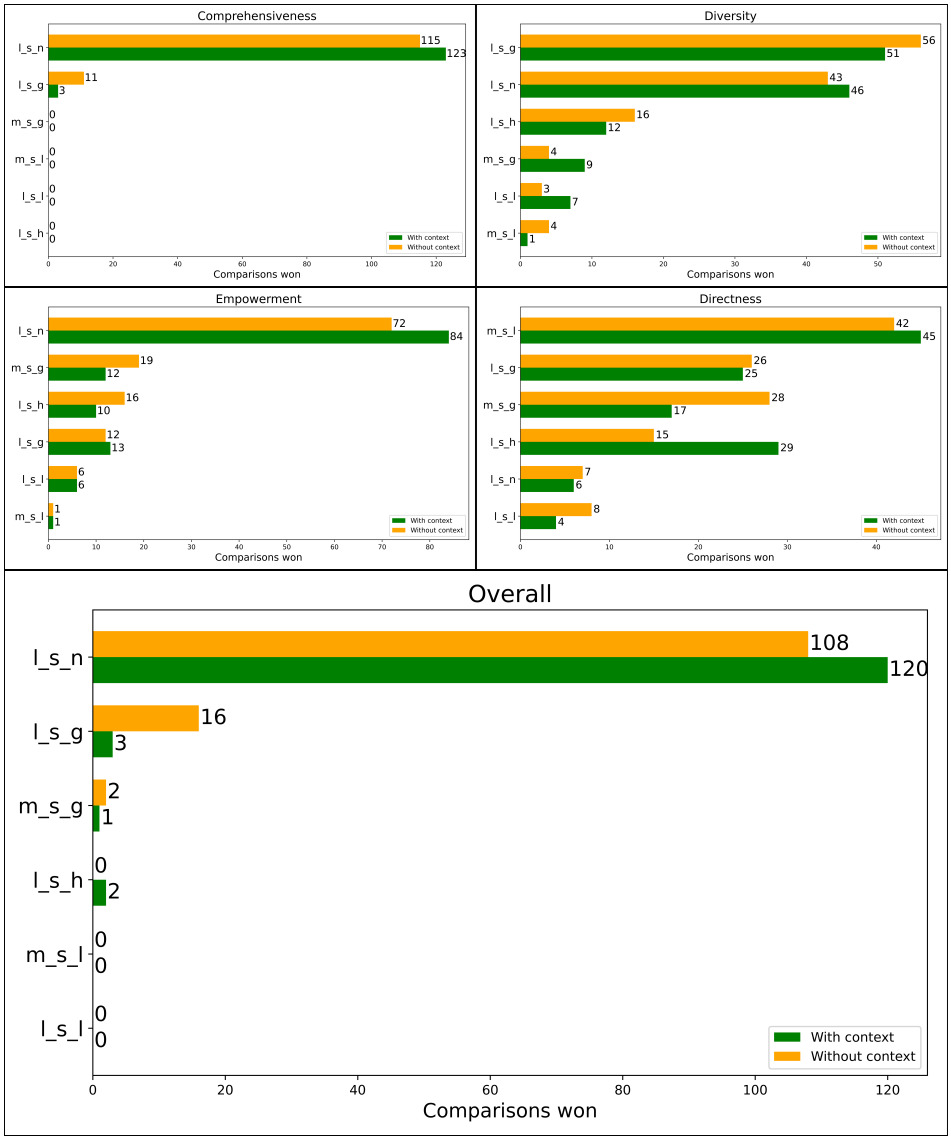

Figure 5. This figure displays experiment 1 results. The judge LLM is given all 6 sub-model responses in the same prompt and asked to rank them by best response according to a criterion. The highest ranking model for each query is cataloged. Each bar in this chart represents the aggregate number of wins that model has per criterion on LLM-generated queries with and without context. The comprehensiveness criteria graph is at the top left, diversity at the top right, empowerment in the middle left, directness in the middle right, and overall at the bottom. The "l_s_n" model, a.k.a. naïve LightRAG or RAG, is clearly dominant across all criteria. Other category placements are discussed in the experiment 1 Results subsection.

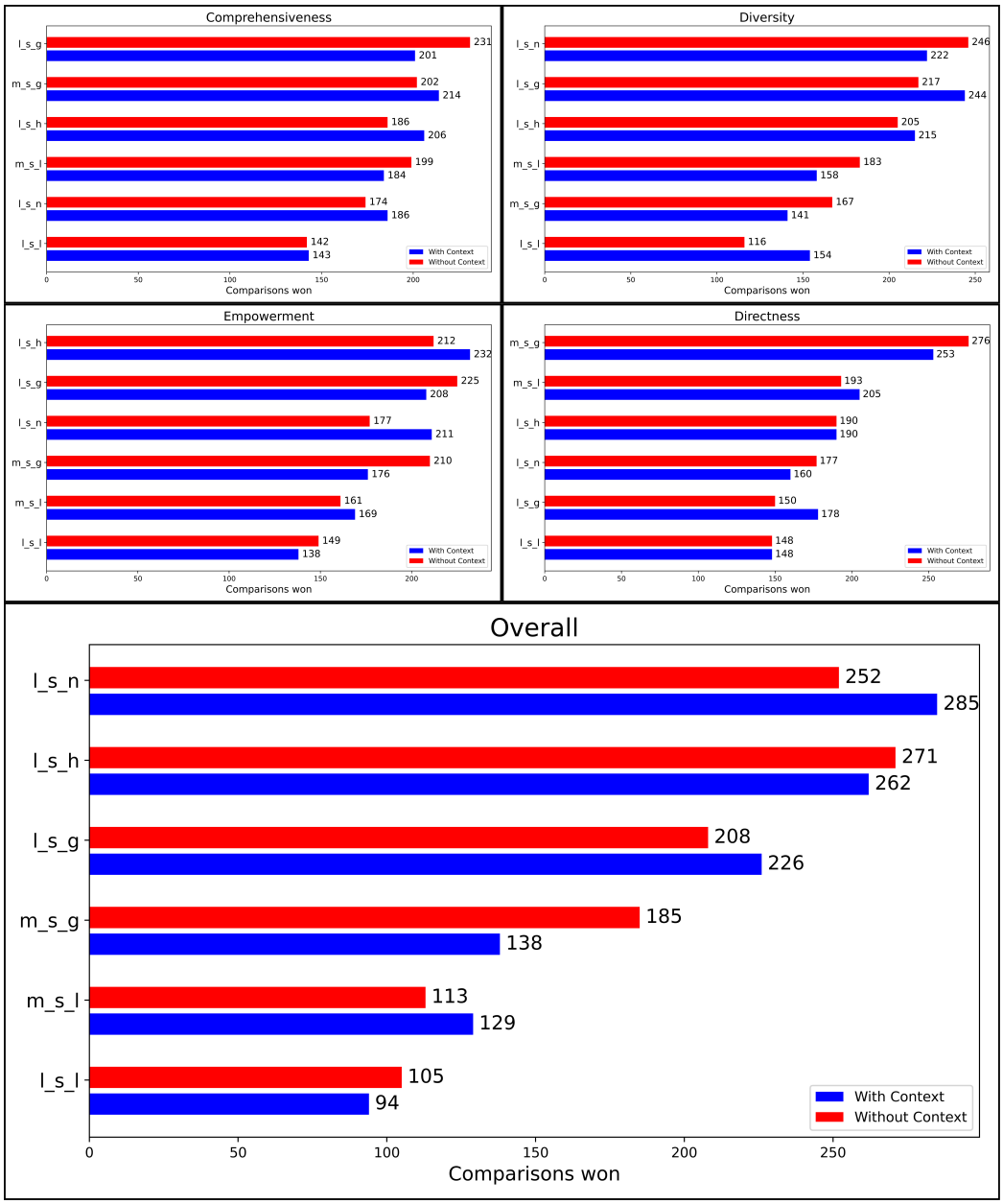

Figure 6. This figure shows experiment 2 Results. The judge LLM is given 2 sub-model responses at a time and asked to rank them by best response according to a criterion. The higher ranking response's model is cataloged. Each bar in this chart represents the aggregate number of wins a model has per criterion on LLM-generated queries with and without context. The comprehensiveness criteria graph is at the top left, diversity at the top right, empowerment in the middle left, directness in the middle right, and overall at the bottom. Winners of each category are discussed in the experiment 2 Results subsection.

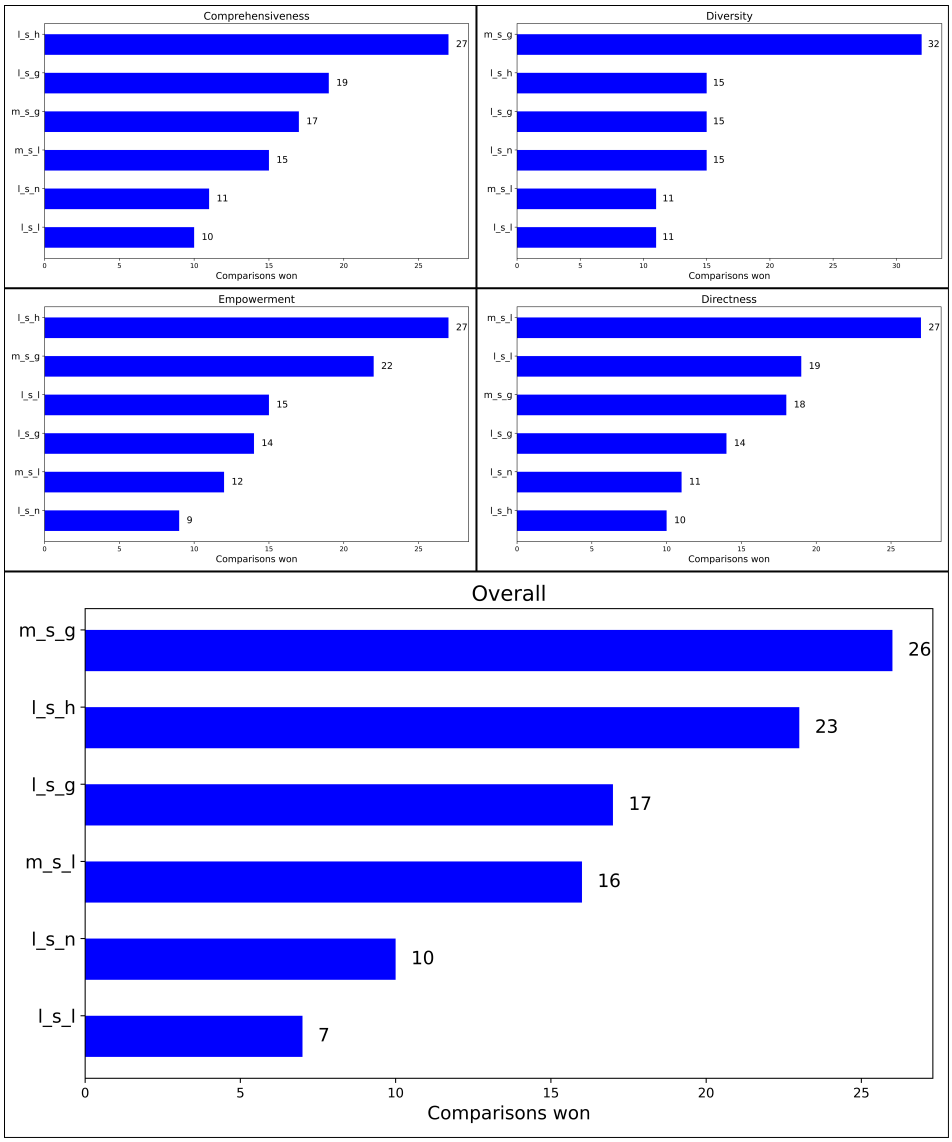

Figure 7. This figure shows experiment 3 Results. The judge LLM is given 2 sub-model responses at a time and asked to rank them by best response according to a criterion. The higher ranking response's model is cataloged. Each bar in this chart represents the aggregate number of wins a model has per criterion on hand-generated queries without context. Comprehensiveness is at the top left, diversity at the top right, empowerment in the middle left, directness in the middle right, and overall at the bottom. Winners of each category are discussed in the experiment 3 Results subsection.

4.1. Experiment 1 Results

Figure 5 displays the results of experiment 1, where the judge LLM is given all six sub-model responses in the same prompt and asked to rank them by best response according to a criterion. For comprehensiveness, naïve and global LightRAG are the only sub-models to win any comparison. Naïve LightRAG far outperformed global LightRAG in this case, winning 123 comparisons with context and 115 without context. Results were more varied for diversity. The global LightRAG model won the most comparisons—51 with context and 56 without context. Naïve LightRAG outshone the rest in empowerment, winning 84 comparisons with context and 72 without context. Local MGRAG won the directness criteria, winning 45 comparisons with context and 42 without context. The overall criteria comparisons were dominated by naïve LightRAG, which won 120 comparisons with context and 108 without context.

4.2. Experiment 2 Results

Results for experiment 2 can be found in Figure 6. While each bar is still an aggregate of that model's wins per criteria, the judge LLM only receives two sub-model responses at a time for comparison. This experiment makes use of the randomized seeds and tournament bracket (see Figure 4). For comprehensiveness, global LightRAG won 201 comparisons with context and 231 without context. Global MGRAG was a close second, winning 214 comparisons with context and 202 without context. For diversity, naïve LightRAG won 222 rounds with context and 246 without context. Global LightRAG came in second, winning 244 rounds with context and 217 without context. In terms of empowerment, hybrid LightRAG won 232 rounds with context and 212 without context. It was followed by global LightRAG which won 208 rounds with context and 225 without context. Global MGRAG won the directness criteria, winning 253 rounds with context and 276 rounds without context. It was followed by local MGRAG, which won 205 rounds with context and 193 without context. Surprisingly, naïve LightRAG again had great performance overall, winning 285 rounds with context and 252 rounds without context. It was closely followed by hybrid LightRAG which won 262 rounds with context and 271 rounds without context.

4.3. Experiment 3 Results

Figure 7 displays the results of experiment 3. Experiment 3 also uses randomized seeds and the tournament bracket (see Figure 4), but on hand-generated queries. Hybrid LightRAG won the comprehensiveness category, coming out on top in 27 comparisons. Global MGRAG dominated the diversity criteria, winning 32 comparisons. In comparison, hybrid, local, and naïve LightRAG each secured 15 wins. Though empowerment wins were a little more distributed, hybrid LightRAG won 27 comparisons. Local MGRAG won the directness criteria with 27 comparison wins. In the final overall assessment, global MGRAG had 26 comparison wins and was closely followed by hybrid LightRAG which won 23 comparisons.

5. Discussion

Experiment 1 results would normally indicate that naïve RAG is superior in most categories to variations of GraphRAG. This defies researcher expectations because current literature suggests that GraphRAG outperforms naïve RAG in similar experiments [9,10]. There are two reasons we believe explain this surprising outcome. First, all six sub-model responses were sent to the judge LLM at the same time. Secondly, the order in which they were sent never changed for this experiment. Naïve RAG being the first response passed in—the order was always l_s_n, l_s_g, l_s_l, l_s_h, m_s_g, m_s_l—could influence the judge LLM rankings (Section 3.5 explains the nomenclature). Contemporary research supports this assessment, stating that the order in which samples are passed to the judging LLM has a strong influence on the results of any ranking task [8]. Because of this, we revised experiment 1 to diminish bias in evaluation, resulting in experiment 2.

Two observations can be drawn from experiment 2 results. First, the combination of a randomized tournament bracket (see Figure 4) and ranking two responses at a time lead to a more evenly distributed outcome (see Figures 6 and 7). The more evenly distributed outcome suggests that the tournament bracket actually reduces some judge LLM bias. Additionally, the use of randomization gives more confidence in these rankings. Unlike results from experiment 1, in which each criterion was completely dominated by naïve RAG—some GraphRAG models never even won a single round—none of the sub-models in experiment 2 win less than 94 comparisons. Naïve RAG still outperformed other sub-models in the "overall" criteria comparison. This second surprising outcome led to our second observation. Questions for experiments 1 and 2, in conformity with the current state of the art for this type

of research [9,10], were all LLM-generated. Naïve RAG's repeated superiority in the aggregate overall calculation suggests that the LLM-generated questions are directly (syntactically) similar to the embedded chunks [20]. It is conceivable that questions from this test pool did not require in-depth semantical understanding of the dataset.

However, we observe an interesting change in pipeline behavior in experiment 3. Naïve RAG no longer dominated the criteria competitions. In fact, naïve RAG now ranked fourth or worse in everycriteria comparison. This leads us to two conclusions. First, hand-crafted queries are more desirable for testing RAG performance than LLM-generated queries. Second, MGRAG appears to be better for users who are interested in maximizing all tested criteria—comprehensiveness, diversity, empowerment, directness.

Our hypothesis proved incorrect. Though it won some individual criteria and came close to first in others, hybrid LightRAG did not win first place in the overall criteria for any experiment. More research needs to be conducted to determine the direct cause, but we suspect MGRAG won experiment 3 due to its use of community-level summaries. One of LightRAG's objectives is to cut down on the cost GraphRAG incurs in its elaborate extraction and summarization methodology [10]. It may have achieved that goal while slightly diminishing the computational capability of the hybrid LightRAG model. Though our hypothesis proved incorrect, it is important to note that the performance of any of the tested models only validates the necessity for these experiments. The retrieval platforms mentioned earlier (e.g., Google Scholar, Semantic Scholar, Arxiv) and other such platforms are currently unable to win against the tested models because they do not directly answer user queries.

Future Work

An approach that combines RAG/GraphRAG pipelines with BI/D would yield a hybrid model that supplements BI/D's network of supporting metadata with RAG and GraphRAG's ability to isolate and palatably present relevant context as responses to queries. Since our goal was to highlight key differences between RAG and BI/D methods, less challenging questions were sufficient. However, more challenging hand-crafted questions can be created if the focus is on testing the ability of individual models. We leave these as projects for future work. Exploring how GraphRAG can be leveraged for chronicling the development of a concept/theory (e.g., idea h was developed in year i. This made it possible for j to k. Following this, l did m to allow h h to n...) is another possibility for future work. The data used in our three experiments are heterogeneous—they span different (possibly disjoint) articles. Not all articles agree with each other and they are certainly not all written by the same entity. Investigating the applicability of this framework on fully homogeneous data may yield interesting results. Exploring how RAG pipelines might be used for LLM-based security could also be fruitful. For example, one could augment bio-security approaches like [21] with LLM-generated secret keys, questions, or passphrases. Of course this would rely on the development of methods that diminish LLM propensity to generate semantically similar content [20]. Finally, using full articles instead of just abstracts could lead to interesting discoveries. Questions could be generated from the "just abstracts" pool and tested in the "full text" context or vice versa to reveal latent facts about the articles and/or retrieval models.

6. Conclusions

Freely making information available to researchers is one of the advantages of BI/D-dependent platforms (Arxiv, CitationGeco, etc.) when compared to RAG and GraphRAG. Other research cataloging platforms (Google Scholar, Semantic Scholars, etc.) join them in presenting researchers with documents, but they all remain unable to directly answer user queries. To address this research gap, we explore methods that not only enhance

the retrieval of information, but also present that information in palatable ways. RAG introduces the ability to directly answer user queries (e.g., Figures 1 and 2), but struggles when they require significant logical interpellation. GraphRAG improves on RAG by using entity and relationship extraction along with text chunking to respond to queries that cannot be edited to become more retriever-friendly.

The framework we used to examine research article abstracts, not full documents, provided answers to data pool spanning questions without the need to train or retrain an LLM—addressing objective 1. The graphs included in this article address objectives 2, 3, and 4, displaying the ranked results of RAG and GraphRAG sub-model performance on all five criteria with and without context. Finally, experiment 3 demonstrates GraphRAG's ability to tackle data pool spanning questions that demand deep semantic understanding of data (objective 5). Results suggest that Microsoft's Graph Retrieval-Augmented Generation (MGRAG) model is superior when taking into account a response's comprehensiveness, diversity, empowerment, and directness. Supplementing BI/D with RAG or GraphRAG pipelines would positively impact the way both beginners, such as new graduate students, and experienced researchers find and parse through volumes of potentially relevant information.

Author Contributions: Conceptualization, J.N. and D.B.R.; methodology, J.N.; software, J.N.; validation, J.N. and D.B.R.; formal analysis, J.N.; investigation, J.N.; resources, J.N.; data curation, J.N.; writing—original draft preparation, J.N.; writing—review and editing, J.N.; visualization, J.N.; supervision, D.B.R.; project administration, D.B.R.; funding acquisition, D.B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the DoD Center of Excellence in AI/ML at Howard University under Contract W911NF-20-2-0277 with the US ARL. However, any opinion, finding, and conclusions or recommendations expressed in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agency.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: This work would not have been possible without the guidance and support of Danda B. Rawat.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
BI/D	Bibliographic Indexing/Databasing
GraphRAG	Graph Retrieval-Augmented Generation
HyDE	Hypothetical Document Embeddings
LightRAG	Light Retrieval-Augmented Generation
LLM	Large Language Model
MGRAG	Microsoft's Graph Retrieval-Augmented Generation
ML	Machine Learning
NLP	Natural Language Processing
NN	Neural Network
QFS	Query Focused Summarization

RAFT	Retrieval-Augmented FineTuning
RAG	Retrieval-Augmented Generation
RMBC	RoBERTa Multi-scale CNN BiGRU Self-attention CRF

References

1. CitationGecko. Gecko-React. 2025. Available online: https://github.com/CitationGecko/gecko-react (accessed on 11 February 2025).
2. About Connected Papers. 2025. Available online: https://www.connectedpapers.com/about (accessed on 11 February 2025).
3. Wang, Z.; Chu, Z.; Doan, T.V.; Ni, S.; Yang, M.; Zhang, W. History, Development, and Principles of Large Language Models-An Introductory Survey. *arXiv* **2024**, arXiv:2402.06853. [CrossRef]
4. Jelinek, F. *Statistical Methods for Speech Recognition*; Language, Speech, and Communication; MIT Press: Cambridge, MA, USA, 1998.
5. Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychol. Rev.* **1958**, *65*, 386–408. [CrossRef] [PubMed]
6. Bengio, Y. Learning Deep Architectures for AI. *J. Mach. Learn. Res.* **2003**, *3*, 1337–1365.
7. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–55. [CrossRef]
8. Ye, J.; Wang, Y.; Huang, Y.; Chen, D.; Zhang, Q.; Moniz, N.; Gao, T.; Geyer, W.; Huang, C.; Chen, P.Y.; et al. Justice or Prejudice? Quantifying Biases in LLM-as-a-Judge. *arXiv* **2024**, arXiv:2410.02736.
9. Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Larson, J. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv* **2024**, arXiv:2404.16130.
10. Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; Huang, C. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv* **2024**, arXiv:2410.05779.
11. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *arXiv* **2013**, arXiv:1310.4546.
12. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv* **2021**, arXiv:2005.11401.
13. Ammar, B.W.; Groeneveld, D.; Bhagavatula, C.; Beltagy, I.; Crawford, M.; Downey, D.; Dunkelberger, J.; Elgohary, A.; Feldman, S.; Ha, V.A.; et al. Construction of the Literature Graph in Semantic Scholar. In Proceedings of the North American Chapter of the Association for Computational Linguistics, New Orleans, LA, USA, 1–6 June 2018.
14. Shi, L.; Zou, X.; Dai, C.; Ji, Z. Uniting Multi-Scale Local Feature Awareness and the Self-Attention Mechanism for Named Entity Recognition. *Mathematics* **2023**, *11*, 2412. [CrossRef]
15. Zeng, X.; Huang, M.; Zhang, H.; Ji, Z.; Ganchev, I. Novel Human Activity Recognition and Recommendation Models for Maintaining Good Health of Mobile Users. *WSEAS Trans. Inf. Sci. Appl.* **2024**, *21*, 33–46. [CrossRef]
16. Zhang, T.; Patil, S.G.; Jain, N.; Shen, S.; Zaharia, M.; Stoica, I.; Gonzalez, J.E. RAFT: Adapting Language Model to Domain Specific RAG. In Proceedings of the First Conference on Language Modeling, Montreal, QC, Canada, 7–10 October 2025 .
17. Gao, L.; Ma, X.; Lin, J.; Callan, J. Precise Zero-Shot Dense Retrieval without Relevance Labels. *arXiv* **2022**, arXiv:2212.10496.
18. Trivedi, H.; Balasubramanian, N.; Khot, T.; Sabharwal, A. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. *arXiv* **2023**, arXiv:2212.10509.
19. Zheng, H.S.; Mishra, S.; Chen, X.; Cheng, H.T.; Chi, E.H.; Le, Q.V.; Zhou, D. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. *arXiv* **2024**, arXiv:2310.06117.
20. Meincke, L.; Girotra, K.; Nave, G.; Terwiesch, C.; Ulrich, K.T. Using Large Language Models for Idea Generation in Innovation. In *The Wharton School Research Paper Forthcoming*; The Wharton School, University of Pennsylvania: Philadelphia, PA, USA, 2024. [CrossRef]
21. Bhuva, D.R.; Kumar, S. A novel continuous authentication method using biometrics for IOT devices. *Internet Things* **2023**, *24*, 100927. [CrossRef]