CG-RAG: Research Question Answering by Citation Graph Retrieval-Augmented LLMs

Yuntong Hu
yuntong.hu@emory.edu
Emory University
Atlanta, GA, USA

Zhihan Lei
zhihan.lei@emory.edu
Emory University
Atlanta, GA, USA

Zhongjie Dai
dzj@tongji.edu.cn
Tongji University
Shanghai, China

Allen Zhang
azhang490@gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

Abhinav Angirekula
aa125@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, IL, USA

Zheng Zhang
zheng.zhang@emory.edu
Emory University
Atlanta, GA, USA

Liang Zhao
liang.zhao@emory.edu
Emory University
Atlanta, GA, USA

ABSTRACT

Research question answering requires accurate retrieval and contextual understanding of scientific literature. However, current Retrieval-Augmented Generation (RAG) methods often struggle to balance complex document relationships with precise information retrieval. In this paper, we introduce Contextualized Graph Retrieval-Augmented Generation (CG-RAG), a novel framework that integrates sparse and dense retrieval signals within graph structures to enhance retrieval efficiency and subsequently improve generation quality for research question answering. First, we propose a contextual graph representation for citation graphs, effectively capturing both explicit and implicit connections within and across documents. Next, we introduce Lexical-Semantic Graph Retrieval (LeSeGR), which seamlessly integrates sparse and dense retrieval signals with graph encoding. It bridges the gap between lexical precision and semantic understanding in citation graph retrieval, demonstrating generalizability to existing graph retrieval and hybrid retrieval methods. Finally, we present a context-aware generation strategy that utilizes the retrieved graph-structured information to generate precise and contextually enriched responses using large language models (LLMs). Extensive experiments on research question answering benchmarks across multiple domains demonstrate that our CG-RAG framework significantly outperforms RAG methods combined with various state-of-the-art retrieval approaches, delivering superior retrieval accuracy and generation quality.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**;
• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Graph retrieval-augmented generation, research question answering, hybrid retrieval, graph learning, citation graphs

ACM Reference Format:
Yuntong Hu, Zhihan Lei, Zhongjie Dai, Allen Zhang, Abhinav Angirekula, Zheng Zhang, and Liang Zhao. 2018. CG-RAG: Research Question Answering by Citation Graph Retrieval-Augmented LLMs. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Question answering is a critical domain recently driven by advancements in Large Language Models (LLMs). While LLMs exhibit exceptional capabilities in addressing commonsense questions [2, 13, 20], their pre-trained knowledge inevitably becomes outdated over time, rendering them insufficient for delivering timely, precise, and comprehensive answers, particularly for complex scientific and domain-specific questions. Additionally, the substantial cost of continuously fine-tuning and updating LLMs makes maintaining their relevance impractical.

For open-domain question answering, which requires relevant contexts for precise answers, Retrieval-Augmented Generation (RAG) [21] offers a promising solution. By integrating external knowledge, RAG addresses the limitations of static pre-trained LLMs, enabling access to up-to-date, domain-specific information and thereby enhancing answer accuracy [34, 36, 42]. RAG comprises two components: a retriever, which identifies relevant information from the database based on the query, and a generator, which utilizes the retrieved information to construct responses. The quality of generation heavily depends on the effectiveness of the retrieval

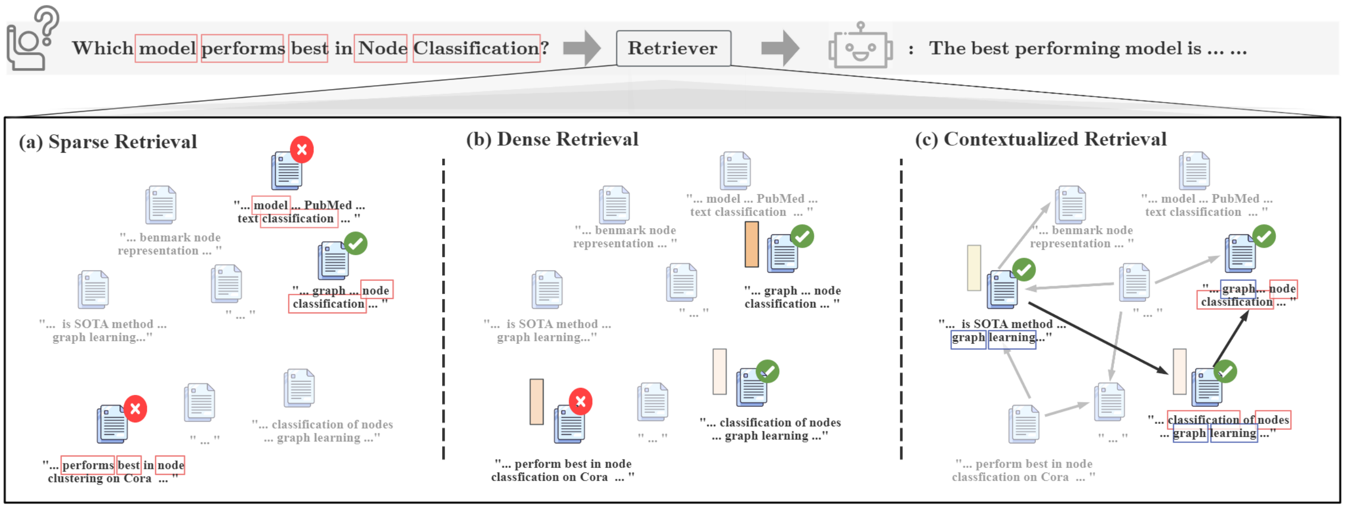
Which model performs best in Node Classification?
Retriever
: The best performing model is … …
(a) Sparse Retrieval
"... model ... PubMed ... text classification ... "
"... benmark node representation ... "
"... graph ... node classification ... "
"... is SOTA method ... graph learning..."
" ... "
"... classification of nodes ... graph learning ..."
"... performs best in node clustering on Cora ... "
(b) Dense Retrieval
"... model ... PubMed ... text classification ... "
"... benmark node representation ... "
"... graph ... node classification ... "
"... is SOTA method ... graph learning..."
" ... "
"... classification of nodes ... graph learning ..."
"... perform best in node classfication on Cora ... "
(c) Contextualized Retrieval
"... model ... PubMed ... text classification ... "
"... benmark node representation ... "
"... graph ... node classification ... "
"... is SOTA method ... graph learning..."
" ... "
"... classification of nodes ... graph learning ..."
"... perform best in node classfication on Cora ... "

Figure 1: Illustration of retrieval-augmented research question answering using: (a) sparse retrieval based on lexical matches, (b) dense retrieval based on semantic relevance, and (c) contextualized retrieval leveraging graph context, i.e., interactions between documents. ▯ represents the dense embedding, where deeper colors indicate a higher semantic relevance to the question. Boxed text in red highlights the matched terms between questions and documents, while Boxed text in blue highlights the matched terms between documents.

process [45, 47]. Traditional retrieval methods primarily focus on lexical and semantic relevance to evaluate the relevance between documents and queries, utilizing sparse and dense retrieval signals, respectively [25, 48]. For graph-based databases, however, these methods fall short, as they fail to account for intricate inter-document relationships. For example, simple retrieval methods often overlook critical citation links in paper citation networks, which are essential for capturing nuanced connections between documents.

Unlike typical QA tasks, which range from layman-level to domain-expert-level based on well-curated documents or knowledge graphs, this paper focuses on more challenging QA at the research frontier, which can only be addressed by analyzing the body of research papers, a task referred to as **research question answering**. Research question answering necessitates considering connections between documents through citation links to ensure comprehensiveness [10, 24, 38]. In citation graphs, the relevant papers are connected not only semantically but also via citation links, encapsulating structured contextual information. Leveraging this contextual information is essential for enhancing both retrieval accuracy and the quality of generated responses, as illustrated in Figure 1. Specifically, as shown in Figure 1(c), each paper requires an evaluation of lexical and semantic relevance, as well as its citations to other relevant documents. For instance, a paper focused on "graph representation learning" is theoretically correlated to a query on "node classification," but it cannot be retrieved using sparse or dense retrieval alone. When multi-hop citation links are considered, however, it becomes highly relevant. Notably, not all citation-linked papers are pertinent to the query, requiring an approach that integrates lexical and semantic relevance, structural patterns, and graph-based context to ensure accurate retrieval.

To tackle this problem, we propose ***Contextualized Graph Retrieval-Augmented Generation (CG-RAG)***, a novel framework tailored for research question answering. An important consideration of research question answering is that research paper content is highly heterogeneous: for instance, the information in the related work, methodology, and experimental sections each serve different purposes and answer distinct types of questions. Therefore, effective modeling requires breaking down documents into semantic chunks—coherent sections representing specific aspects of a research paper. To achieve this, we mathematically construct a citation graph at the chunk level, where each chunk represents a semantically meaningful module of the paper. These chunks form a graph structure with intra-paper and inter-paper connections, capturing both internal coherence and external relationships. To represent this, we introduce the ***Contextual Citation Graph***, which decomposes citation graphs into chunk-level granularity, enabling fine-grained relationship discovery.

To synergize lexical and semantic retrieval signals inside and across networked documents in citation graphs, we propose the ***Lexical-Semantic Graph Retrieval (LeSeGR)*** method. This approach convolves over query-relevant subgraphs, where edges and nodes are characterized by lexical and semantic relevance scores. Importantly, we theoretically demonstrate that the existing post-retrieval paradigm is a special case of our approach. When documents are contextually linked, entangling retrieval signals through graph structures enhances retrieval performance. Finally, the subgraph embeddings are used as input to LLMs for generating final answers. Our experiments conducted on citation graphs from diverse scientific domains demonstrate the superior retrieval and generation effectiveness of CG-RAG. Furthermore, our evaluations reveal that retrieval-augmented LLMs equipped with CG-RAG outperform state-of-the-art retrieval strategies, significantly enhancing the quality of LLM-generated responses.

The rest of the paper is organized as follows. Section 2 reviews related work on retrieval-augmented generation. Section 3 highlights key aspects of retrieval-augmented research question answering within the context of citation graphs. Section 4 introduces a novel formulation that extends beyond the current retrieval paradigm and details our proposed retrieval strategy and retrieval-augmented generation method. Section 5 presents experimental results, comparing our approach with RAG frameworks using various state-of-the-art retrieval methods for research question answering. Finally, Section 6 concludes the paper with key insights.

2 RELATED WORK

2.1 Information Retrieval

Information retrieval (IR) focuses on extracting relevant information from large corpora. Two primary retrieval techniques dominate the field: *sparse retrieval* and *dense retrieval*. Sparse retrieval methods, such as TF-IDF [32] and BM25 [31], rely on term-based representations to evaluate lexical matches between queries and documents. These approaches perform well in scenarios where exact term matching is essential, but they struggle with semantic meaning. In contrast, dense retrieval methods leverage pre-trained language models such as BERT [5] to encode queries and documents as continuous, low-dimensional embeddings, capturing semantic similarity through maximum inner product search (MIPS) [16, 30, 43, 44]. Dense retrieval effectively overcomes the lexical gap, retrieving semantically related results even when query terms differ from the document's terminology. Recently, *hybrid retrieval* techniques have also emerged to combine the strengths of sparse and dense methods while addressing their respective limitations [25, 26, 29]. By integrating sparse and dense signals, these approaches provide a robust solution for retrieving relevant information from long and complex documents.

2.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a technique that integrates external retrieval systems to enhance large language models (LLMs) [9, 21, 22]. Unlike traditional LLMs that rely solely on pre-trained knowledge, RAG leverages external sources during inference, enabling more accurate and up-to-date responses. This makes RAG particularly effective for specialized tasks, such as literature-based question answering [27, 46]. Naturally, current literature question-answering systems are predominantly built upon RAG frameworks [1, 10, 35, 39], relying mainly on dense retrieval combined with LLMs. Literature data, however, is inherently graph-structured, where topological information, such as hierarchical and citation relationships, plays a crucial role in the retrieval and generation processes. As such, most existing literature question answering methods fail to incorporate this structural context effectively. Recently, GraphRAG [6, 11, 12] was introduced to extend RAG to graph-related scenarios, offering the potential to capture complex interconnections in literature datasets by leveraging graph-based relationships such as hierarchical and citation structures. In our study, we focus on leveraging graph context to enhance the retrieval and generation processes in literature question answering.

3 RESEARCH QUESTION ANSWERING

Citation Graph. A citation Graph, $\mathcal{G} = (V, E, \{d_v\}_{v \in V})$, consists of a node set V and an edge set E, where each node $v \in V$ is associated with natural language attributes in a paper. These papers, represented by $\mathcal{D} = \{d_v\}_{v \in V}$, include textual information such as abstracts, sections, and other relevant content.

Research Question Answering. Given a query q over the citation graph $\mathcal{G}$, the objective of research question answering is to generate an appropriate answer by leveraging the relevant information retrieved from $\mathcal{G}$. Formally, this objective is defined as:

$$p_\theta(Y|q, \mathcal{G}) = \arg\max_\theta \prod_{i=1}^{n} p_\theta(y_i|y_{<i}, q, \mathcal{G}), \tag{1}$$

where θ represents the parameters of the generative language model, $Y = \{y_i\}_{i=1}^n$ is the generated answer sequence, and $y_{<i}$ denotes the prefix tokens of the sequence up to position $i - 1$.

The quality of the generated answer is highly dependent on the effectiveness of the retrieval process within the citation graph. Let f_o denote the relevance scoring function for a retrieval system o. Recent advances in graph-based retrieval leverage structural information, formalized as:

$$f_{\text{graph}} : g(f_{\text{dense}}) \to \mathbb{R}, \tag{2}$$

where $g(\cdot)$ encodes topological information. These methods primarily rely on dense representations, which often fail to capture sparse lexical matches—essential in citation graphs for identifying exact cross-references and key terms critical to retrieval accuracy. Thus, integrating sparse and dense retrieval is essential. Existing hybrid retrieval systems combine these signals via post-retrieval fusion: $f_{\text{hybrid}} : f_{\text{sparse}} \bigoplus f_{\text{dense}} \to \mathbb{R}$, where $\bigoplus$ denotes operations such as score fusion. Treating documents as isolated entities, however, limits their effectiveness in structured databases such as citation graphs. Designing a retrieval system that is both lexically and semantically aware in graph retrieval remains an unresolved challenge.

4 METHODOLOGY

4.1 Overview

To overcome these limitations, we propose Contextualized Graph Retrieval-Augmented Generation (CG-RAG), introducing a novel retrieval method called **<u>Le</u>xical-<u>Se</u>mantic <u>G</u>raph <u>R</u>etrieval (LeSeGR)**, which integrates discrete sparse signals and continuous dense signals in a manner that respects the graph topology. Formally, the paradigm is defined as:

$$f_{\text{entangled}} : g(f_{\text{sparse}} \bigotimes f_{\text{dense}}) \to \mathbb{R}, \tag{3}$$

where $g(\cdot)$ is a graph encoder that incorporates structured context during retrieval, and $\bigotimes$ represents the entangled fusion of sparse and dense signals. The transition to $f_{\text{entangled}}$ offers greater generality and capabilities, but requires addressing fundamental challenges:

- Section 4.2 introduces the contextual citation graph, which captures chunk-level cross-relationships by surrounding each chunk with its relevant context.
- Section 4.3 introduces the Lexical-Semantic Graph Retrieval that integrates sparse and dense signals within graph-based scenarios. We prove that it encompasses existing hybrid retrieval methods based on post-retrieval approaches as a special case when graph contextual information is absent and extends dense-signal-only graph retrieval, highlighting its generalizability to current retrieval frameworks.
- Section 4.4 introduces Contextualized Graph Retrieval-Augmented Generation that leverages retrieved contextual subgraphs by LeSeGR to improve the quality of generated responses.

4.2 Contextual Citation Graph

Research paper content is highly diverse, with sections like related work, methodology, and experiments serving distinct purposes and answering different questions, necessitating the segmentation of documents into semantic chunks that capture specific aspects. Given a citation graph $\mathcal{G} = (V, E, \{d_v\}_{v \in V})$, each document d_v is decomposed into a set of chunks C_v, with all chunks collectively forming $C = \bigcup_{v \in V} C_v$. Chunks may reference each other within the same document (*intra-connections*) or across different documents (*inter-connections*). Intra-connections are explicit, such as a section referencing earlier subsections within the same paper, while inter-connections occur when one paper cites another without explicitly linking to specific chunks within it. Intra-document links typically provide highly relevant context, whereas inter-document links offer supplementary but less significant information. To capture these interactions during retrieval, we propose the hierarchical citation graph, modeling relationships both within and across documents. Formally, it is defined as $\bar{\mathcal{G}} = (\bar{V}, \bar{E}, C)$, where $C = \{c_i\}_{i \in \bar{V}}$ represents the set of chunks.

4.2.1 Chunk Node. Each chunk $c_i \in C$ corresponds to a fixed-length segment of text extracted from a document in the citation graph $\mathcal{G}$. Specifically, documents are divided into chunks with a maximum token length of l, such that a document with a total token length of L is divided into $\lceil \frac{L}{l} \rceil$ chunks. These chunks serve as the nodes in the hierarchical citation graph $\bar{\mathcal{G}}$.

4.2.2 Chunk-Chunk Edge. The edges in $\bar{\mathcal{G}}$ represent relationships between chunks, capturing both intra- and inter-document connections. Edge weights reflect the strength and nature of these relationships.

Intra-document Edges connect chunks within the same document, preserving the logical flow and structural hierarchy of the text. When $c_i, c_j \in C_v$, an edge is established if a structural dependency exists between them, such as adjacency or explicit cross-references. Adjacency refers to the logical connection between two consecutive chunks, where c_j precedes c_i, resulting in an edge $c_j \longrightarrow c_i$. Explicit cross-references occur when c_i refers to c_j, establishing an edge $c_j \longrightarrow c_i$.

Inter-document Edges, in contrast, connect chunks across different documents. For a chunk $c_i \in C_u$, the Top-n relevant chunks in C_v are linked to c_i if $(v, u) \in E$. The relevance r_{ij} between c_i and c_j is computed using the relevance scoring functions f_{sparse} and f_{dense}: $f_{\text{sparse}}(\mathbf{c}_i^{\text{sparse}}, \mathbf{c}_j^{\text{sparse}}) + f_{\text{dense}}(\mathbf{c}_i^{\text{dense}}, \mathbf{c}_j^{\text{dense}})$, where $\mathbf{c}_i^{\text{sparse}}$ and $\mathbf{c}_i^{\text{dense}}$ are the sparse and dense representations of chunk c_i, respectively, and similarly for c_j.

4.3 Lexical-Semantic Graph Retrieval (LeSeGR)

Given a hierarchical citation graph $\bar{\mathcal{G}} = (\bar{V}, \bar{E}, C, w)$, the representation of each chunk $c_i \in C$ is designed to combine the advantages of sparse lexical vectors and dense semantic vectors. Additionally, if the contexts around c_i contain relevant information, the representation incorporates contributions from these contextual chunks. This forms the basis of an entangled representation, integrating both sparse-dense fusion and graph contextual information.

Algorithm 1 Lexical-Semantic Graph Retrieval.

```
Require: Citation graph $\mathcal{G} = (V, E, \{d_v\}_{v \in V})$, Query $q$
Ensure: A list of contextual subgraphs $\{\bar{\mathcal{G}}\}$
▷Initialize graph structures and representations.
if cached contextual graph $\bar{\mathcal{G}}$ exists then
    Load $\bar{\mathcal{G}} = (\bar{V}, \bar{E}, C)$
else
    Generate $\bar{\mathcal{G}} = (\bar{V}, \bar{E}, C)$ from $\mathcal{G}$
    Cache $\bar{\mathcal{G}}$ for future use
end if
▷Initialize sparse and dense representations for the query and all chunks.
$\mathbf{c}_i^{\text{sparse}}, \mathbf{c}_i^{\text{dense}} \longleftarrow$ sparse and dense encoders for $\forall c_i \in C$
$\mathbf{q}^{\text{sparse}}, \mathbf{q}^{\text{dense}} \longleftarrow$ sparse and dense encoders for $q$
▷Compute initial query relevance for all chunks.
for each chunk $c_i \in C$ do
    $\delta_{qi} \longleftarrow f_{\text{sparse}}\mathbf{q}^{\text{sparse}}, \mathbf{c}_i^{\text{sparse}})$          ▷ Sparse Signal
    $\alpha_{ij} \longleftarrow \mathrm{MLP}_\phi(\mathbf{c}_i^{\text{dense}} \ominus \mathbf{c}_j^{\text{dense}})$          ▷ Dense Signal
end for
▷Perform message passing through the graph.
for each layer $k = 1, \ldots, K$ do
    for each chunk $c_i \in C$ do
        ▷Compute messages from neighbors and itself.
        $\mathbf{m}_j^{(k)} \longleftarrow \mathrm{MSG}^{(k)}\big(\delta_{qj} \cdot \alpha_{ij} \cdot \mathbf{h}_j^{(k)}\big)\forall j \in \{i\} \cup \mathcal{N}(i)$
        ▷Aggregate messages.
        $\mathbf{h}_i^{(k+1)} \longleftarrow \mathrm{AGG}^{(k)}\big(\{\mathbf{m}_j^{(k)} \mid j \in \{i\} \cup \mathcal{N}(i)\}\big)$
    end for
end for
▷Compute relevance scores with entangled representations.
for each chunk $c_i \in C$ do
    $s(q, c_i) \longleftarrow f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{h}_i^{(K)})$
end for
▷Select top-relevant chunks and construct subgraphs.
$S(\bar{\mathcal{G}}; q) \longleftarrow$ Top-$N$ contextual subgraphs based on $s(q, c_i)$
▷Return the retrieved subgraphs.
return $S(\bar{\mathcal{G}}; q)$
```

The entangled representation of c_i is obtained through a graph encoder, such as a GNN, which aggregates information based on the contextual graph:

$$\mathbf{H}_i = g_\Phi(\{\mathbf{h}_j^{(0)}, \delta_{qj}, \alpha_{ij} \mid j \in \{i\} \cup \mathcal{N}(i)\}), \tag{4}$$

where Φ denotes the parameters of the graph encoder, and $\mathbf{h}_j^{(0)} = \mathbf{c}_j^{\text{dense}}$ is the initial dense representation of c_j. The terms δ_{qj} and α_{ij} control the message passing, ensuring that only relevant information is propagated. Specifically, the message contribution from a neighboring chunk c_j to c_i and the subsequent update are defined as:

$$\mathbf{m}_j^{(k)} = \text{MSG}^{(k)}\left(\delta_{qj} \cdot \alpha_{ij} \cdot \mathbf{h}_j^{(k)}\right), \tag{5}$$

$$\mathbf{h}_i^{(k+1)} = \text{AGG}^{(k)}\left(\{\mathbf{m}_j^{(k)} \mid j \in \{i\} \cup \mathcal{N}(i)\}\right), \tag{6}$$

where $\mathbf{h}_j^{(k)}$ is the representation of c_j at layer k. The term δ_{qj} evaluates the relevance between the query and the context, while α_{ij} measures the relevance between the central chunk c_i and its neighboring chunks c_j. Specifically, δ_{qi}, the sparse relevance between a query q and a chunk c_i, is computed as:

$$\delta_{qi} = f_{\text{sparse}}(\mathbf{q}_i^{\text{sparse}}, \mathbf{c}_i^{\text{sparse}}). \tag{7}$$

where f_{sparse} indicates a relevance scoring function used in sparse retrieval such as cosine similarity and dot product. This incorporates sparse relevance into the dense embedding during message passing. To model the dense interaction between c_i and its neighboring chunks $c_j \in \mathcal{N}(i)$, α_{ij} is calculated as:

$$\alpha_{ij} = \text{MLP}_\phi(\mathbf{c}_i^{\text{dense}} \ominus \mathbf{c}_j^{\text{dense}}), \tag{8}$$

where α_{ii} is defined as 1, $\ominus$ represents element-wise subtraction to compute the feature difference, and MLP_ϕ parameterized by ϕ, adaptively assesses the relevance between chunks.

This entangled representation framework integrates sparse and dense features while leveraging structural information from the graph context, ensuring that each chunk's representation reflects both its intrinsic relevance and its contextual relationships.

Given a query q over the chunk set C of a citation graph $\mathcal{G}$, the relevance scoring function $f_{\text{entangled}}(q, c_i) : Q \times C \to \mathbb{R}$ is:

$$s(q, c_i) = f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{H}_i), \tag{9}$$

where f_{dense} indicates a relevance scoring function used in dense retrieval, $\mathbf{q}^{\text{dense}}$ represents the dense vector of the query, and $\mathbf{H}_i$ denotes the entangled representation of the chunk c_i. The overall algorithm is depicted in Algorithm 1.

Proposition 4.1 (LeSeGR Generality). *Post-retrieval methods with the metric $f_{hybrid} : f_{sparse} \bigoplus f_{dense} \to \mathbb{R}$, represent a special case of the proposed Lexical-Semantic Graph Retrieval (LeSeGR). This holds when no additional relevant contextual information exists for any chunk in the citation graph, reducing LeSeGR to existing hybrid retrieval used post-retrieval fusion.*

Proof. The entangled representation of a chunk c_i is given by:

$$\mathbf{H}_i = \text{AGG}(\{\mathbf{h}_j^{(k)} \mid j \in \{i\} \cup \mathcal{N}(i)\}) = \text{AGG}(\lambda_i \mathbf{m}_i^{(k)}, \lambda_\mathcal{N} \mathbf{m}_\mathcal{N}^{(k)}), \tag{10}$$

where $\mathbf{m}_i^{(k)}$ and $\mathbf{m}_\mathcal{N}^{(k)}$ denote messages from the central chunk c_i and its neighboring chunks at layer k, respectively. The weights λ_i and $\lambda_\mathcal{N}$ are typically equal and defined as $\frac{1}{|\mathcal{N}(i)|+1}$.

When the AGG function is defined as either mean or sum, both of which satisfy the distributive law, the relevance score $s(q, c_i)$, as defined in Equation 9, can be expanded as:

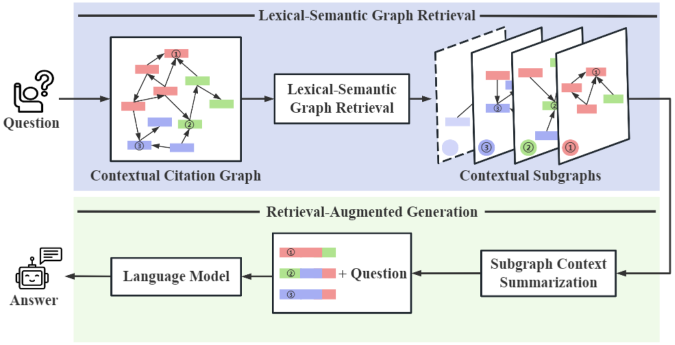
Lexical-Semantic Graph Retrieval
Question
Contextual Citation Graph
Lexical-Semantic Graph Retrieval
Contextual Subgraphs
Retrieval-Augmented Generation
Answer
Language Model
+ Question
Subgraph Context Summarization

Figure 2: Overview of Contextualized Graph Retrieval-Augmented Generation.

$$s(q, c_i) = f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \text{AGG}(\lambda_i \mathbf{m}_i^{(k)}, \lambda_\mathcal{N} \mathbf{m}_\mathcal{N}^{(k)})) \tag{11}$$

$$\propto \lambda_i f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{m}_i^{(k)}) + \lambda_\mathcal{N} f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{m}_\mathcal{N}^{(k)}). \tag{12}$$

When no relevant neighbors exist (i.e., $\mathcal{N}(i) = \emptyset$), the second term vanishes, leaving:

$$s(q, c_i) \propto \lambda_i f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{m}_i^{(k)}) = \lambda_i \delta_{qi} f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{h}_i^{(k)}). \tag{13}$$

Substituting $\mathbf{h}_i^{(k)} = \mathbf{c}_i^{\text{dense}}$, the relevance score simplifies to:

$$s(q, c_i) \propto \delta_{qi} f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{c}_i^{\text{dense}}). \tag{14}$$

Taking the logarithm for further analysis:

$$\log(s(q, c_i)) \propto \log(\delta_{qi}) + \log(f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{c}_i^{\text{dense}})). \tag{15}$$

Since $\delta_{qi} = f_{\text{sparse}}(\mathbf{q}^{\text{sparse}}, \mathbf{c}_i^{\text{sparse}})$, this becomes:

$$\log(s(q, c_i)) \propto \log(f_{\text{sparse}}(\mathbf{q}^{\text{sparse}}, \mathbf{c}_i^{\text{sparse}})) \tag{16}$$

$$+ \log(f_{\text{dense}}(\mathbf{q}^{\text{dense}}, \mathbf{c}_i^{\text{dense}})), \tag{17}$$

where the use of $\log(\cdot)$ requires f_{sparse} and f_{dense} to be mapped to $\mathbb{R}^+$. This requirement can be fulfilled by applying appropriate activation functions or transformations to transform them from $\mathbb{R}$ into the non-negative domain. In this case, the relevance score is equivalent to the additive fusion of sparse and dense relevance, as used in post-retrieval hybrid methods. Hence, when no contextual information exists ($\mathcal{N}(i) = \emptyset$), our graph-contextualized retrieval reduces to the post-retrieval fusion paradigm, demonstrating that the latter is a specific instance of the former. □

When relevant graph contexts are present, the entangled framework dynamically propagates and aggregates sparse and dense signals through structural relationships among neighboring chunks. This enables the model to capture relational dependencies and multi-hop connections in graphs, enhancing retrieval accuracy and effectively utilizing sparse and dense signals from neighbors.

4.4 Contextualized Graph Retrieval-Augmented Generation (CG-RAG)

Given a query q on a citation graph $\mathcal{G}$, we use LeSeGR to retrieve the top N chunks that are most relevant to the question. These retrieved chunks, together with their contextual subgraph, are then used to generate the answer, as illustrated in Figure 2. Specifically, for each selected chunk, its corresponding contextual subgraph $\bar{\mathcal{G}}_i$ is retained for the generation phase, where $\bar{\mathcal{G}}_i = \bar{\mathcal{G}}[\{i\} \cup \mathcal{N}(i)]$ represents the induced subgraph consisting of chunk c_i and its direct neighbors. Formally, the set of contextual subgraphs for the Top-N chunks is defined as:

$$S(\bar{\mathcal{G}}; q) = \bigcup_{c_i \in \text{Top-}N(q,C)} \bar{\mathcal{G}}_i, \qquad (18)$$

where Top-$N(q, C)$ represents the Top-N chunks ranked by $s(q, c_i)$. This ensures the retrieval retains both the most relevant chunks and their graph context for downstream tasks.

To effectively utilize the contextual information of each retrieved chunk and adapt to various LLMs, including open-source models such as LLaMA and closed-source models such as ChatGPT, we first summarize the graph context and then concatenate this summarized context with the central chunks to enhance generation. Specifically, for each contextual subgraph $\bar{\mathcal{G}}_i \in S(\bar{\mathcal{G}}; q)$, we prompt the LLM to summarize the contextual information surrounding c_i. The summarized context is concatenated with the query to form the final input for generating the answer. The generation process over the citation graph $\mathcal{G}$ is formally defined as:

$$p_\theta(Y|q, \mathcal{G}) = \arg\max_{\theta} \prod_{i=1}^{n} p_\theta(y_i | y_{<i}, X_q, X_C), \qquad (19)$$

where θ represents the LLM parameters, $X_q = \text{TextEmbedder}(q)$ is the query embedding, and X_C is the context embedding:

$$X_C = \text{TextEmbedder}\left(\left[\text{Summarize}(q, \bar{\mathcal{G}}_i)\right]_{\bar{\mathcal{G}}_i \in S(\bar{\mathcal{G}}; q)}\right), \qquad (20)$$

representing the embeddings of the concatenated summarized contexts from the retrieved contextual subgraphs. The summarization is performed by the LLM itself, extracting relevant information from the contextual subgraph to aid in generating the final answer.

5 EXPERIMENT

We conduct experiments to evaluate the effectiveness (Section 5.2) and efficiency (Section 5.3) of Contextualized Graph RAG, along with an analysis of the individual contributions of our technical designs (Section 5.4).

5.1 Settings

Datasets. Our experiments utilize two datasets: PubMedQA-1k and PapersWithCodeQA. PubMedQA-1k is a publicly available dataset, introduced by Jin et al., and comprises 1,000 question-answer pairs designed for PubMed literature [1], with human-labeled gold-standard retrieval and answer annotations. The original dataset, however, lacks citation information between papers, which we addressed by extracting the references for each paper and constructing a citation graph database with a total of 7,849 papers.

The PapersWithCodeQA dataset was collected from the PapersWithCode website[2], which tracks research papers across various computer science fields. We used 84 leaderboards[3] spanning diverse domains, including *Computer Vision*, *Natural Language Processing*, *Medical*, and *Graphs*. For each leaderboard, we extracted the top 20 papers' contents and references from arXiv[4] to construct a graph database. The LaTeX content of each paper was preserved as its textual attributes. The dataset comprises 12,171 papers, from which we also crafted 924 questions centered on leaderboard analysis, with ground truth answers derived directly from the leaderboards. These include 420 True/False questions, 420 multiple-choice questions, and 84 generative questions. For generative questions, we first provide the LLMs with the most relevant contexts labeled by humans and allow the LLMs to generate answers. We then evaluate the quality of retrieval-augmented generation by replacing the human-selected contexts with those retrieved by different retrieval methods.

Table 1: Example question-answering pairs and corresponding evaluation metrics.

True or False:	Do mitochondria play a role in remodelling lace plant leaves during programmed cell death? Yes, No or Maybe.
Answer:	Yes
Metrics:	Accuracy (Acc), F_1 Score.
Multiple Choice:	Which model achieves state-of-the-art* performance on the ADE20K dataset for semantic segmentation? (a) BEiT-3 (b) DINOv2 (c) ONE-PEACE (d) EVA
Answer:	(c)
Metrics:	Mean Reciprocal Rank (MRR), Hit@k.
Essay:	Could you provide an overview of the model development for semantic segmentation?
Answer:	... Early models like FCN (Fully Convolutional Networks) laid the foundation by adapting classification networks for pixel-level predictions ... Recently, ONE-PEACE emerged as a state-of-the-art model ...
Metrics:	Coherence, Consistency, Relevance

*Within the citation graph we collected.

Metrics. To comprehensively evaluate the performance of RAG systems in retrieving relevant information and generating accurate, contextually appropriate answers, we employ distinct metrics for retrieval and question answering. For retrieval, we use Hit@1 and Hit@3, which measure the proportion of queries where the correct chunk is ranked within the top-1 and top-3 retrieved results, respectively.

For research question answering, example questions and used evaluation metrics are presented in Table 1. For multiple-choice questions, we use Mean Reciprocal Rank (MRR) and Hit@k to assess ranking quality. For True/False questions, Accuracy (Acc) and F_1 score evaluate classification performance. For generative tasks, we leverage the UniEval model [49] to assess Coherence, Consistency, and Relevance, which evaluate logical flow, factual accuracy, and topical alignment, respectively. All metrics adhere to the principle that higher values indicate better model performance.

Implementations. Experiments are conducted using two NVIDIA A10 GPUs, with Graph Transformer [33] serving as the graph encoder. The configuration includes two layers, each featuring four

[1] https://pubmed.ncbi.nlm.nih.gov/

[2] https://paperswithcode.com/

[3] https://paperswithcode.com/sota

[4] https://arxiv.org/

Table 2: Evaluation of the retrieval-augmented research question answering. The best performance is highlighted in BOLD, while the second-best performance is underlined. Performance of our methods is highlighted .

		PapersWithCodeQA							PubMedQA	
Category	Method	Acc	F_1	MRR	Hit@1	Coherence	Consistency	Relevance	Acc	F_1
Sparse	BM25	0.689	0.617	0.765	0.736	0.905	0.858	0.859	0.662	0.604
	Doc2Query	0.705	0.629	0.748	0.731	0.914	0.833	0.852	0.684	0.614
	BGE-M3	0.751	0.648	0.810	0.787	0.934	0.876	0.863	0.722	0.644
Dense	MiniLM	0.730	0.644	0.782	0.758	0.919	0.872	0.828	0.712	0.641
	LaBSE	0.591	0.552	0.677	0.643	0.875	0.545	0.616	0.403	0.396
	mContriever	0.523	0.531	0.647	0.613	0.863	0.469	0.438	0.288	0.271
	E5	0.579	0.560	0.659	0.628	0.872	0.521	0.544	0.363	0.360
	SPAR	0.611	0.583	0.643	0.609	0.886	0.549	0.537	0.392	0.384
Hybrid	Score Fusion	0.739	0.656	0.774	0.749	0.908	0.891	0.887	0.674	0.613
	ColBERT	0.769	0.661	0.827	0.778	0.927	0.884	0.874	0.724	0.642
	CLEAR	0.618	0.575	0.667	0.643	0.894	0.623	0.685	0.468	0.456
	LeSeGR (Ours)	**0.835**	**0.703**	**0.884**	**0.827**	**0.956**	**0.921**	**0.914**	**0.778**	**0.685**

attention heads and a hidden dimension size of 1024. The maximum chunk length is set to 8,192 tokens. Training is conducted using CrossEntropy, with 10% of the samples labeled with gold-standard retrieval, and optimized using the AdamW optimizer [23]. The relevance scoring function for both dense and sparse representations is dot product. For generation, GPT-4 is employed through the OpenAI API, specifically leveraging the `gpt-4o-2024-05-13` model version.

Baseline methods. To evaluate the proposed graph-contextualized retrieval method, we benchmark it against several state-of-the-art baselines renowned for their effectiveness in the retrieval phase across various RAG systems. These retrieval techniques are categorized into three groups: sparse retrieval, dense retrieval, and hybrid retrieval.

- **Sparse Retrieval**: *BM25* [31], an advanced refinement of TF-IDF [32], enhances relevance scoring by incorporating probabilistic modeling, term saturation, and document length normalization, providing robust performance in keyword-based retrieval tasks. *Doc2Query* [28] improves sparse retrieval by generating synthetic queries for documents using a pre-trained language model (PLM). *BGE-M3* [3] utilizes a multi-vector architecture to create robust representations, integrating contrastive learning and knowledge distillation techniques.
- **Dense Retrieval**: *MiniLM* [41] is a lightweight transformer model that uses deep self-attention distillation to create dense embeddings. *LaBSE* [7] is a bilingual embedding model, leveraging dual encoders and a large-scale parallel corpus to ensure semantic alignment across languages. *mContriever* [14] employs unsupervised contrastive learning to train dense retrievers, focusing on encoding diverse and nuanced contextual information. *E5* [40] optimizes embeddings for text retrieval by integrating explicit supervision from retrieval datasets and task-specific fine-tuning. *SPAR* [4] employs salient phrase representation learning to bridge dense and sparse retrieval, utilizing a dual encoder architecture that explicitly models both phrase-level and document-level semantics.
- **Hybrid Retrieval**: *ScoreFusion* [19] combines the output scores of sparse and dense retrieval models to produce a unified ranking. *ColBERT* [17] introduces late interaction to compute pairwise term similarities between query and document embeddings, enabling efficient and fine-grained integration of sparse and dense signals. *CLEAR* [8] employs a residual learning framework to combine sparse and dense representations, ensuring complementary signals are utilized for improved retrieval performance.

5.2 Main Results

Our proposed Contextualized Graph Retrieval-Augmented Generation with LeSeGR achieves state-of-the-art performance across all tasks and datasets, as shown in Table 2. For true/false questions, LeSeGR significantly surpasses sparse, dense, and hybrid baselines in both accuracy (Acc) and F_1 scores, demonstrating its capability to effectively capture domain-specific terms and semantic nuances. Hybrid methods such as ScoreFusion combine sparse and dense signals but fail to achieve the deeper integration of retrieval signals offered by LeSeGR. Through its graph-structured integration, LeSeGR dynamically propagates and entangles retrieval signals from contextual information, fully leveraging the relationships embedded in the graph structure. This advanced integration translates to superior performance, particularly in metrics such as MRR and Hit@1.

In generative tasks, our method demonstrates significant improvement in Coherence, Consistency, and Relevance by leveraging its entangled sparse-dense representation and graph-based contextualization. Unlike hybrid baselines such as ColBERT, which emphasizes token-level interactions but overlooks graph-level relationships, LeSeGR's graph encoder dynamically aggregates signals across interconnected chunks. This enhances contextual understanding, enabling high-quality and contextually rich text generation. For instance, LeSeGR achieves a Coherence score of 0.956 on PapersWithCodeQA, outperforming ColBERT's 0.927. These results underscore the unique strengths of LeSeGR in effectively

Table 3: Evaluation of the retrieval effectiveness on the citation graph of PubMed (PubMedQA). The best performance is highlighted in BOLD, while the second-best performance is underlined.

Category	Method	Hit@1	Hit@3
Sparse	BM25	0.835	0.912
	Doc2Query	0.832	0.930
	BGE-M3	0.915	0.960
Dense	MiniLM	0.887	0.945
	LaBSE	0.305	0.471
	mContriever	0.472	0.496
	E5	0.231	0.355
	SPAR	0.256	0.385
Hybrid	Score Fusion	0.829	0.925
	ColBERT	0.913	0.968
	CLEAR	0.470	0.612
	LeSeGR (Ours)	**0.961**	**0.987**

bridging sparse and dense retrieval with graph-based contextualization, thereby advancing the retrieval-augmented generation process to new levels of effectiveness.

Table 3 demonstrates that LeSeGR significantly outperforms all baselines in the retrieval phase, including sparse, dense, and hybrid approaches. Our method achieves superior retrieval accuracy by effectively entangling sparse and dense signals within the graph structure, allowing contextual information to enhance relevance scoring. Unlike post-retrieval fusion methods, such as ScoreFusion, which combine sparse and dense signals after separate retrieval processes, our approach dynamically integrates these signals during retrieval, leading to more coherent and effective utilization of both lexical and semantic information. Furthermore, while ColBERT performs token-level interactions for fine-grained relevance, it operates at the query-document level without fully leveraging the structural relationships present in citation graphs. In contrast, our method extends relevance computation to the graph structure, propagating and aggregating signals across related chunks to capture multi-hop and relational dependencies. This deeper integration of graph context and entangled sparse-dense signals enables our method to outperform ColBERT, achieving the highest Hit@1 and Hit@3 scores.

5.3 Efficiency Analysis

As shown in Table 4, LeSeGR demonstrates competitive retrieval efficiency on the citation graph of arXiv (PapersWithCodeQA). It strikes a balance between memory usage and latency, leveraging GPU computation effectively. LeSeGR achieves faster query latency (403.94 ms) compared to ColBERT (561.91 ms) while maintaining a moderate GPU memory footprint (1,921 MB). In contrast, ScoreFusion exhibits high CPU memory usage (5,655 MB) and slower query speeds, whereas LeSeGR optimizes GPU utilization by integrating both sparse and dense retrieval signals into the message passing process of the graph encoder. Additionally, LeSeGR outperforms CLEAR in query speed while maintaining similar memory usage. These results highlight LeSeGR's efficiency and scalability for large-scale graph-based retrieval tasks without compromising effectiveness.

Table 4: Evaluation of the retrieval efficiency on the citation graph of arXiv (PapersWithCodeQA).

Method	CPU & GPU Memory		Indexing & Query Latency	
	(MB)	(MB)	(ms)	(ms)
Score Fusion	5,655	770	43.94	1,580.14
ColBERT	0	12,674	12.40	561.91
CLEAR	0	1,538	205.36	16.07
LeSeGR	0	1,921	19.22	403.94

5.4 Ablation Studies

The ablation studies performed for each influential factor in LeSeGR is shown in Table 5. In this experiment, we evaluate LeSeGR on the citation graph of PubMed, which contains 7,849 papers. Our main observations are as follows:

Table 5: Ablation studies on PubMedQA. The default settings of LeSeGR are marked with *.

Factor	Setting	Hit@1	Hit@3
Graph Encoder	GAT	0.939	0.968
	GCN	0.955	0.976
	Graph Transformer*	**0.961**	**0.987**
Top-*n* Context	2	0.931	0.949
	8	0.950	0.984
	4*	**0.961**	**0.987**
Sparse Signal	TF-IDF	0.903	0.962
	BM25	0.919	0.962
	Doc2Query	0.926	0.964
	BGE-M3*	**0.961**	**0.987**
Dense Signal	E5	0.676	0.765
	mContriever	0.838	0.883
	MiniLM*	**0.961**	**0.987**

- **Graph Encoder.** Among Graph Attention Networks (GAT) [37], Graph Convolutional Networks (GCN) [18], and Graph Transformer [33], Graph Transformer achieves the highest Hit@1 and Hit@3 scores of 0.961 and 0.987, respectively. Notably, regardless of which graph encoder is employed, our LeSeGR method still consistently achieves the best retrieval performance when compared to the baselines in Table 3. This underscores LeSeGR's superior ability to model complex relationships and effectively aggregate contextual information for retrieval.
- **Top-*n* Context.** We further assess three configurations for the number of contexts connected via inter-document edges, i.e., Top-*n*. The $n = 4$ setting achieves the best results, striking a balance between sufficient contextual inclusion and noise reduction. Smaller values, such as $n = 2$, restrict the scope of context, while larger values, such as $n = 8$, may introduce irrelevant information, diluting the positive impact of relevant context and reducing retrieval effectiveness. It is anticipated that if the chunk length

is sufficiently large, retaining only the top-1 relevant chunk will suffice.

- **Retrieval Signals.** We compare four sparse representation methods. BGE-M3 outperforms other sparse encoders, achieving the highest scores, as its ability to integrate lexical and semantic features is critical for domain-specific term matching. TF-IDF and BM25, while strong in lexical precision, lack semantic adaptability. In addition, we also compare three dense representation methods, with MiniLM delivering the best performance. MiniLM's compact representation effectively captures semantic nuances, making it better suited for diverse queries and documents. As a whole, however, combining two expressive retrieval models with our LeSeGR framework results in stronger retrieval performance. Notably, the performance of LeSeGR appears to be primarily constrained by the quality of the dense retrieval signal.

6 CONCLUSION

In this work, we introduce Lexical-Semantic Graph Retrieval (LeSeGR), a novel framework that integrates sparse, dense, and graph-structured retrieval signals for complex and structured database. Based on LeSeGR, we present Contextualized Graph Retrieval-Augmented Generation (CG-RAG) for research question answering. By leveraging a contextual citation graph, our approach effectively captures intra- and inter-document relationships, enabling a dynamic propagation of contextual information through an entangled hybrid retrieval paradigm. This paradigm bridges lexical precision and semantic understanding while generalizing to existing retrieval methods. Furthermore, CG-RAG incorporates a graph-aware generation strategy, enhancing the contextual richness of generated responses. Extensive experiments across multiple citation networks demonstrate the superior performance of CG-RAG based on LeSeGR, achieving state-of-the-art results in retrieval metrics such as Hit@1 and generation metrics such as Coherence and Relevance. Our findings underscore the effectiveness of graph-contextualized representations in advancing the capabilities of retrieval-augmented generation for citation graphs, setting a new benchmark for retrieval-augmented research question answering.

REFERENCES

[1] Sören Auer, Allard Oelen, Muhammad Haris, Markus Stocker, Jennifer D'Souza, Kheir Eddine Farfar, Lars Vogt, Manuel Prinz, Vitalis Wiens, and Mohamad Yaser Jaradeh. 2020. Improving access to scientific literature with knowledge graphs. *Bibliothek Forschung und Praxis* 44, 3 (2020), 516–529.

[2] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2024), 1–45.

[3] 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. *arXiv preprint arXiv:2402.03216* (2024).

[4] Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2021. Salient phrase aware dense retrieval: can a dense retriever imitate a sparse one? *arXiv preprint arXiv:2110.06918* (2021).

[5] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130* (2024).

[7] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *arXiv preprint arXiv:2007.01852* (2020).

[8] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complement lexical retrieval model with semantic residual embeddings. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*. Springer, 146–160.

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

[10] Hamed Babaei Giglou, Tilahun Abedissa Taffa, Rana Abdullah, Aida Usmanova, Ricardo Usbeck, Jennifer D'Souza, and Sören Auer. 2024. Scholarly Question Answering using Large Language Models in the NFDI4DataScience Gateway. *arXiv preprint arXiv:2406.07257* (2024).

[11] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630* (2024).

[12] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. GRAG: Graph Retrieval-Augmented Generation. *arXiv preprint arXiv:2405.16506* (2024).

[13] Yuntong Hu, Zheng Zhang, and Liang Zhao. 2023. Beyond Text: A Deep Dive into Large Language Models' Ability on Understanding Graph Data. *arXiv preprint arXiv:2310.04944* (2023).

[14] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).

[15] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146* (2019).

[16] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6769–6781.

[17] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[18] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[19] Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Leveraging semantic and lexical matching to improve the recall of document retrieval systems: A hybrid approach. *arXiv preprint arXiv:2010.01195* (2020).

[20] Jens Lehmann, Antonello Meloni, Enrico Motta, Francesco Osborne, Diego Reforgiato Recupero, Angelo Antonio Salatino, and Sahar Vahdati. 2024. Large Language Models for Scientific Question Answering: An Extensive Analysis of the SciQA Benchmark. In *European Semantic Web Conference*. Springer, 199–217.

[21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[22] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110* (2022).

[23] I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[24] Alejandro Lozano, Scott L Fleming, Chia-Chun Chiang, and Nigam Shah. 2023. Clinfo. ai: An open-source retrieval-augmented large language model system for answering medical questions using scientific literature. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2024*. World Scientific, 8–23.

[25] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345.

[26] Priyanka Mandikal and Raymond Mooney. 2024. Sparse Meets Dense: A Hybrid Approach to Enhance Scientific Document Retrieval. *arXiv preprint arXiv:2401.04055* (2024).

[27] Kurnia Muludi, Kaira Milani Fitria, Joko Triloka, et al. 2024. Retrieval-Augmented Generation Approach: Document Question Answering using Large Language Model. *International Journal of Advanced Computer Science & Applications* 15, 3 (2024).

[28] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[29] Vít Novotnỳ and Michal Stefánik. 2022. Combining Sparse and Dense Information Retrieval.. In *CLEF (Working Notes)*. 104–118.

[30] N Reimers. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084* (2019).

[31] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[32] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[33] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509* (2020).

[34] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics* 11 (2023), 1–17.

[35] Markus Stocker, Allard Oelen, Mohamad Yaser Jaradeh, Muhammad Haris, Omar Arab Oghli, Golsa Heidari, Hassan Hussein, Anna-Lena Lorenz, Salomon Kabenamualu, Kheir Eddine Farfar, et al. 2023. FAIR scientific information with the open research knowledge graph. *FAIR Connect* 1, 1 (2023), 19–21.

[36] Tilahun Abedissa Taffa and Ricardo Usbeck. 2023. Leveraging LLMs in Scholarly Knowledge Graph Question Answering.. In *QALD/SemREC@ ISWC*.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[38] Chengrui Wang, Qingqing Long, Xiao Meng, Xunxin Cai, Chengjun Wu, Zhen Meng, Xuezhi Wang, and Yuanchun Zhou. 2024. BioRAG: A RAG-LLM Framework for Biological Question Reasoning. *arXiv preprint arXiv:2408.01107* (2024).

[39] Haiwen Wang, Le Zhou, Weinan Zhang, and Xinbing Wang. 2021. LiteratureQA: A Qestion Answering Corpus with Graph Knowledge on Academic Literature. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4623–4632.

[40] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[41] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems* 33 (2020), 5776–5788.

[42] Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. 2024. REAR: A Relevance-Aware Retrieval-Augmented Framework for Open-Domain Question Answering. *arXiv preprint arXiv:2402.17497* (2024).

[43] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 641–649.

[44] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[45] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of Retrieval-Augmented Generation: A Survey. *arXiv preprint arXiv:2405.07437* (2024).

[46] Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R Dalal, Jennifer L Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, et al. 2024. Almanac—retrieval-augmented language models for clinical medicine. *NEJM AI* 1, 2 (2024), AIoa2300068.

[47] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473* (2024).

[48] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems* 42, 4 (2024), 1–60.

[49] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197* (2022).

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009