BG-RAG: Concept-Mediated Bipartite Graphs for Scalable RAG

Hang Gao[1,2]* **Yangmin Ding**[1] **Shaobo Han**[1] **Zhuocheng Jiang**[1]
[1]NEC Laboratories America [2]Rutgers University
h.gao@rutgers.edu {yding,shaobo}@nec-labs.com zhuocheng_jiang@hotmail.com

Abstract

Graph-structured retrieval-augmented generation (GraphRAG) has shown strong potential for improving multi-hop reasoning in large language models. However, existing approaches typically depend on dense entity–entity linking, which is computationally expensive, susceptible to hallucinated relations, and difficult to maintain under continual updates. We present Bipartite Graph Retrieval-Augmented Generation (BG-RAG), a lightweight bipartite framework that organizes knowledge into three layers, from chunks to entities to concepts, and replaces direct entity links with concept-mediated connections. This design enables more efficient graph construction, reduces reliance on hallucination-prone relation-edge prediction by avoiding explicit entity–entity relation extraction, and naturally supports incremental updates without global reconstruction. Experiments on HotpotQA, 2Wiki, and MuSiQue show that BG-RAG achieves competitive performance against RAG and GraphRAG baselines. Its concept-mediated structure also provides a lightweight alternative to dense entity–entity graph construction. These results suggest that BG-RAG is a lightweight alternative for retrieval-augmented generation, particularly when avoiding dense entity–entity linking is desirable.

1 Introduction

Large language models (LLMs) augmented with retrieval have achieved notable gains in factual accuracy and reasoning (Lewis et al., 2020; Arslan et al., 2024; Gao et al., 2023b). In particular, retrieval-augmented generation (RAG) enhances LLMs by grounding responses in external documents. However, standard RAG pipelines typically rely on vector similarity alone, which often falls short on multi-hop reasoning tasks that require connecting evidence spread across multiple passages. To address this limitation, recent work has proposed graph-based RAG approaches that construct entity-level graphs from text and use graph traversal to assemble richer contexts (Edge et al., 2024; Han et al., 2024; Peng et al., 2025; Zhang et al., 2025).

And existing GraphRAG methods face several challenges. First, constructing dense entity–entity graphs can incur quadratic pairwise-linking cost in the number of extracted entities. Second, dependence on LLM-based relation extraction introduces spurious or hallucinated edges, which propagate noise into retrieval paths. Third, some graph-construction pipelines require rebuilding when new documents are added, making them poorly suited for dynamic or evolving collections. These limitations restrict the scalability and reliability of current GraphRAG approaches in practice.

We introduce BG-RAG, a lightweight alternative that rethinks graph construction as a three-layer bipartite structure: from chunks to entities to concepts. Instead of linking entities directly to each other, BG-RAG groups them under higher-level concepts, which act as semantic hubs. This design offers three advantages: (i) graph construction becomes more efficient by avoiding costly pairwise linking, (ii) reliance on hallucination-prone relation-edge prediction is reduced by mediating connections through shared concepts, and (iii) the bipartite structure naturally supports incremental updates, enabling new documents to be integrated without rebuilding the entire graph. At query time, BG-RAG maps the input to relevant entities and concepts, then expands to retrieve compact, concept-linked neighborhoods of supporting evidence, which are fed into the LLM. Across multi-hop QA benchmarks ((HotpotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), MuSiQue (Trivedi et al., 2022))), BG-RAG achieves competitive answer quality compared to the evaluated RAG and GraphRAG baselines. Its concept-mediated de-

*Work done during an internship at NEC Laboratories America.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 917–922
November 6-10, 2026 ©2026 Association for Computational Linguistics

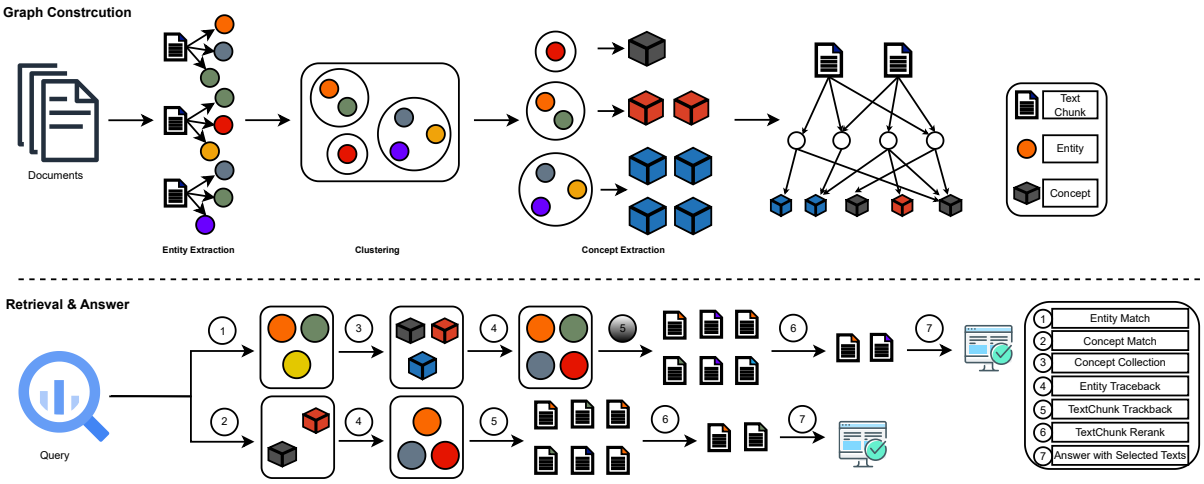


Figure 1: Architecture of the three-layer bipartite framework BG-RAG. Knowledge is organized as chunks → entities → concepts, illustrating concept-mediated connectivity and local insertion updates.

sign avoids explicit entity–entity linking, offering a lightweight structure that is easier to update than dense graph constructions. Our contributions are threefold:

1. We propose a bipartite formulation for graph-based RAG that avoids explicit entity–entity relation prediction.
2. We describe an incremental update mechanism that can incorporate new documents through local updates to chunk–entity and entity–concept incidence layers.
3. We demonstrate empirical effectiveness on three multi-hop QA benchmarks, showing that BG-RAG is a practical alternative to existing GraphRAG methods.

2 Related Work

Retrieval-augmented generation (RAG) grounds LLM outputs in external corpora at inference time (Lewis et al., 2020; Arslan et al., 2024; Fan et al., 2024; Gupta et al., 2024). Techniques such as dense retrieval (Karpukhin et al., 2020; Xiong et al., 2021; Rubin et al., 2022; Gao et al., 2026), reranking (Nogueira and Cho, 2019; Glass et al., 2022; Gao et al., 2024) have been shown to improve the quality of retrieved evidence. Nevertheless, vector-only pipelines remain brittle when addressing multi-hop questions, as they struggle to connect dispersed facts that require explicit reasoning across multiple passages (Yang et al., 2018). Graph-based RAG attempts to overcome this limitation by constructing entity-centric graphs and performing graph traversal to assemble structured evidence (Edge et al., 2024; Han et al., 2024; Sarthi et al., 2024; Gao and Metaxas, 2026). However, building dense entity–entity graphs is computationally costly and highly sensitive to extraction errors and LLM-induced hallucinations (Ji et al., 2023). Earlier information-extraction pipelines such as OpenIE (Angeli et al., 2015) scale tuple extraction but often introduce noise. More recent LLM-driven graph builders (e.g., KGGEN (Mo et al., 2026)) mitigate sparsity through clustering, yet they still depend on predicted links and periodic graph rebuilds, which limits their scalability. Taken together, these lines of work span a continuum from flat passage retrieval to relation-rich graph construction. This continuum motivates intermediate designs that preserve structured evidence expansion without requiring a fully relation-labeled graph. Our proposed approach, BG-RAG, departs from these designs by replacing direct entity–entity connections with a bipartite, concept-mediated structure (from chunks to entities to concepts). This formulation (i) avoids quadratic entity–entity linking and reduces the structural complexity of graph construction, (ii) limits reliance on predicted relations by mediating connectivity through shared concepts, and (iii) supports local updates at both the chunk–entity and entity–concept layers. Recent lightweight and relation-free GraphRAG methods likewise reduce reliance on explicit relation extraction, while concept-oriented methods use concepts for context construction or retrieval. BG-RAG is therefore positioned not as the first relation-free or concept-based approach, but as a compact formulation in which concepts serve as explicit mediation nodes between chunk–entity and entity–concept incidence layers.

3 Method

BG-RAG organizes a corpus into a three-level representation (chunks, entities, and concepts) and constructs concept-mediated connections that enable multi-hop retrieval without requiring dense, error-prone entity–entity links. We describe both (i) graph construction and (ii) query-time retrieval. Figure 1 shows the piplines of the BG-RAG.

3.1 Graph Construction

The construction process requires a document set $\mathcal{D}$, a sentence or token splitter, an entity extractor, and a text encoder $f(\cdot)$ for embeddings.

Step G1: Chunking. Each document $d \in \mathcal{D}$ is divided into overlapping chunks $\mathcal{C} = \{c\}$. We store both the chunk text and associated metadata.

Step G2: Entity extraction and normalization. For each chunk c, we apply LLM extraction to obtain entities $\mathcal{E}_c = \{e\}$ with surface forms and types. Entities are normalized through lowercasing, lemmatization, and light canonicalization (e.g., "`U. S.`"→"`United States`"). We record incidence relations $(c, e) \in \mathcal{A}_{CE}$. Each entity is embedded as $\mathbf{e} = f(\text{text}(e))$.

Step G3: Concept induction. Entities are clustered into higher-level *concepts* $\mathcal{K} = \{k\}$ using semantic similarity in embedding space via thresholded nearest-centroid assignment. Let $\mathcal{K}$ denote the current concept set with centroids $\{\boldsymbol{\mu}_k\}_{k\in\mathcal{K}}$. For an entity embedding $\mathbf{e}$, we select

$$k^\star = \arg\max_{k\in\mathcal{K}} \cos\big(\mathbf{e}, \boldsymbol{\mu}_k\big).$$

If $\mathcal{K} \neq \emptyset$ and $\cos\big(\mathbf{e}, \boldsymbol{\mu}_{k^\star}\big) \geq \eta$, we assign the entity to $k^\star$; otherwise, we create a new concept k_{new} with centroid $\boldsymbol{\mu}_{k_{\text{new}}} = \mathbf{e}$. We record the incidence relation $(e, k) \in \mathcal{A}_{EK}$ and update centroids incrementally (EMA update):

$$\boldsymbol{\mu}_k \leftarrow (1-\alpha)\,\boldsymbol{\mu}_k + \alpha\,\mathbf{e}.$$

Here, a concept is an embedding-induced semantic hub rather than a manually named symbolic category or ontology node. The similarity threshold controls concept granularity: a loose threshold may create broad hubs that retrieve distractors, whereas a strict threshold may fragment related entities across concepts. We therefore treat concepts as lightweight retrieval mediators rather than guaranteed clean semantic categories.

Step G4: Storage and indices. We persist only two incidence layers, chunk↔entity ($\mathcal{A}_{CE}$) and entity↔concept ($\mathcal{A}_{EK}$), avoiding explicit entity–entity edges.

By eliminating explicit entity–entity edges, BG-RAG avoids dense pairwise relation prediction. The resulting representation stores only chunk–entity and entity–concept incidence relations, while concepts provide indirect connectivity between related entities.

3.2 Query-Time Retrieval

Given a query q, we encode it with $f(\cdot)$ and apply an evidence budget B (tokens or chunks).

Step Q1: Entity and Concept Matching. The query embedding $\mathbf{q} = f(q)$ is matched to the top-M entities by cosine similarity. These entities activate their associated concepts, which are scored by aggregated entity similarity. The top-K concepts are retained for expansion.

Step Q2: Concept–Entity Expansion. Activated concepts are collected and traced back to their linked entities, recovering semantically related entities that may not appear in the query.

Step Q3: Chunk Gathering. Text chunks connected to either the matched entities or expanded concepts are retrieved. Each chunk is assigned a frequency score reflecting how often it is reached through different entity–concept paths.

Step Q4: Re-ranking and Answer Generation. Candidate chunks are re-ranked using both similarity and frequency, with a light diversity filter to reduce redundancy. The top-ranked chunks fitting budget B are concatenated and fed into the LLM to generate the final answer.

3.3 Incremental Updates and Complexity

When new documents arrive, only steps G1–G3 of graph construction are applied. New (c, e) pairs are added, entities are assigned to existing concepts (or start new concepts if similarity falls below η), and centroids are updated locally. This allows updates without global rebuilds or pairwise relinking. The overall construction cost is dominated by chunk-level entity extraction and concept assignment. Without approximate indexing, nearest-centroid assignment costs $O(|\mathcal{E}||\mathcal{K}|)$, while the proposed structure avoids the $O(|\mathcal{E}|^2)$ cost of dense entity–entity linking. Thus, the efficiency

Algorithm 1 IncrementalUpdate for BG-RAG

for all $d \in \mathcal{D}_{\text{new}}$ **do**
  Split into chunks $\mathcal{C}_d$; persist text + metadata
  for all $c \in \mathcal{C}_d$ **do**
    Extract $\mathcal{E}_c$; normalize
    for all $e \in \mathcal{E}_c$ **do**
      Add (c, e) to $\mathcal{A}_{CE}$
      Compute $\mathbf{v}_e \leftarrow f(\text{text}(e))$
    end for
  end for
  for all $e \in \bigcup_c \mathcal{E}_c$ **do**
    $k^\star \leftarrow \arg\max_k \cos(\mathbf{v}_e, \boldsymbol{\mu}_k)$
    if $\cos(\mathbf{v}_e, \boldsymbol{\mu}_{k^\star}) \geq \eta$ **then**
      Add $(e, k^\star)$ to $\mathcal{A}_{EK}$; $\boldsymbol{\mu}_{k^\star} \leftarrow (1-\alpha)\boldsymbol{\mu}_{k^\star} + \alpha\,\mathbf{v}_e$
    else
      Create k_{new} with $\boldsymbol{\mu}_{k_{\text{new}}} \leftarrow \mathbf{v}_e$; add (e, k_{new}) to $\mathcal{A}_{EK}$
    end if
  end for
end for
return Updated $\mathcal{A}_{CE}$, $\mathcal{A}_{EK}$, centroids $\{\boldsymbol{\mu}_k\}$

claim concerns the structural cost of graph construction: BG-RAG avoids dense pairwise entity linking, while end-to-end runtime additionally depends on entity extraction, encoding, and indexing. At query time, BG-RAG performs entity matching, concept aggregation, and scoring over the resulting candidate chunks, while preserving multi-hop connectivity through concept hubs. Algorithm 1 summarizes the incremental update procedure.

4 Experiments

We evaluate BG-RAG on multi-hop question answering across three challenging benchmarks. MuSiQue (Trivedi et al., 2022) is designed to test compositional reasoning by requiring answers that integrate evidence from multiple passages rather than relying on shortcuts. 2Wiki (Ho et al., 2020) contains questions that demand reasoning across two Wikipedia articles, making it well-suited for studying cross-document inference. HotpotQA (Yang et al., 2018) emphasizes explainable multi-hop reasoning, where both supporting evidence and final answers must be derived from diverse sources. To keep experiments computationally tractable yet representative, we randomly sample 500 queries from each dataset as our evaluation pool. Accordingly, we use these results to assess broad performance patterns across datasets rather than to establish a definitive ranking among systems.

For comparison, we include a range of representative baselines. First, *LLM-only* strategies attempt to answer without retrieval, either through direct prompting (Direct). Next, we evaluate *text-based RAG* methods, which retrieve unstructured passages and condition the LLM on them. This category includes the standard Naïve RAG pipeline as well as HyDE (Gao et al., 2023a). Finally, we benchmark against *graph-based RAG* methods that leverage structured representations to facilitate reasoning, including GraphRAG (Edge et al., 2024) and RAPTOR (Sarthi et al., 2024), both of which incorporate graph or cluster structures to aggregate multi-hop evidence.

Baseline methods	MuSiQue		2Wiki		HotpotQA	
	EM	LJ	EM	LJ	EM	LJ
gpt-4o-mini (Direct)	0.17	0.20	0.34	0.39	0.42	0.47
Naïve RAG	0.28	0.26	0.36	0.40	0.59	0.66
HyDE	0.17	0.29	0.42	0.44	0.54	0.61
GraphRAG	0.25	0.26	0.40	0.37	0.47	0.65
RAPTOR	0.30	0.33	0.49	0.46	0.65	0.67
BG-RAG	**0.36**	**0.39**	**0.46**	**0.51**	**0.63**	**0.71**

Table 1: Performance comparison among baselines and BG-RAG on three benchmarks. The backbone is gpt-4o-mini.

Evaluation is carried out using two complementary measures. Exact Match (EM) requires the predicted answer string to exactly match the ground truth. While EM provides a strict correctness check, it can underestimate performance when semantically valid answers differ in surface form. To address this, we additionally adopt an LLM to Judge protocol (LJ), in which a large language model compares the model prediction with the reference answer and determines whether they are semantically equivalent. LJ is intended to capture semantic equivalence when correct answers differ in surface form.

Results in Table 1 reveal consistent trends across datasets. Direct prompting is generally among the weakest settings, which is consistent with the value of retrieval for these multi-hop QA tasks. Text-based RAG methods yield notable improvements, with Naïve RAG benefiting from grounding in retrieved passages and HyDE better capturing semantically aligned evidence. Graph-structured methods, like RAPTOR and GraphRAG, provide additional gains in several settings, suggesting that structured evidence organization can benefit multi-hop reasoning when the retrieved structure is reliable. BG-RAG is competitive across the evaluated settings, obtaining the highest LJ score on all three benchmarks and the highest EM score on MuSiQue among the representative baselines. The

higher observed LJ scores indicate that BG-RAG's final answers are more often judged semantically equivalent to the references.

5 Conclusion

We introduced BG-RAG, a bipartite graph framework for retrieval-augmented generation that organizes knowledge into chunks, entities, and concepts. The design avoids explicit entity–entity relation prediction and permits local incidence-layer updates. On three sampled multi-hop QA benchmarks, BG-RAG achieves competitive observed performance among the evaluated baselines. These results motivate further study of concept-mediated retrieval, including its efficiency, update behavior, and applicability beyond the evaluated setting.

Limitations

While BG-RAG offers a lightweight and practical alternative to existing GraphRAG approaches, there are several aspects that warrant further investigation. First, our current implementation relies on relatively simple clustering methods for concept induction. Although effective, more sophisticated or domain-aware clustering strategies could further improve concept granularity and retrieval precision. Second, our evaluation is limited to multi-hop QA datasets. Applying BG-RAG to broader domains (e.g., scientific literature, legal texts, or multilingual collections) would help assess its generality. Finally, while our design supports incremental updates, we have not explored long-term dynamics such as continual concept drift or the impact of large-scale streaming data. Because concept membership captures semantic association rather than explicit directionality, roles, comparisons, or temporal order, BG-RAG is intended for associative evidence discovery rather than as a replacement for relation-aware graphs when such constraints are essential. Over long update streams, loose thresholds may produce overly broad hubs, whereas strict thresholds may fragment related entities or cause concepts to proliferate. We leave these directions for future work.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. 2024. A survey on rag with llms. *Procedia computer science*, 246:3781–3790.

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

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.

Hang Gao, Dong Deng, and Yongfeng Zhang. 2024. Vector retrieval with similarity and diversity: How hard is it? *arXiv preprint arXiv:2407.04573*.

Hang Gao and Dimitris N. Metaxas. 2026. Beyond explicit edges: Robust reasoning over noisy and sparse knowledge graphs. In *Forty-third International Conference on Machine Learning*.

Hang Gao, Wujiang Xu, Kai Mei, and Dimitris N Metaxas. 2026. Pooling and semantic shift: The fundamental challenges in long text embedding and retrieval. *arXiv preprint arXiv:2603.21437*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023a. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

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

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao,

Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

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)*, pages 6769–6781.

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

Belinda Mo, Kyssen Yu, Joshua Kazdan, Proud Mpala, Lisa Yu, Charilaos Kanatsoulis, and Sanmi Koyejo. 2026. Kggen: Extracting knowledge graphs from plain text with language models. *Advances in Neural Information Processing Systems*, 38:30092–30115.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2025. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems*, 44(2):1–52.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2655–2671.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pages 32628–32649.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

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