Article | Open Access **ISSN: 3104-9346(P) 3104-9354(O)**

Journal of Advanced Computing and Intelligent Systems

Zan et al. *J. Adv. Comput. Intell. Syst.* **2026**, 2(1), 31-44

Knowledge-Graph-Guided Retrieval-Augmented Generation for Domain-Specific Question Answering

Lei Zan[1,2], Gang Yao[1], Meimei Zhang[1,2]

Keywords:

Retrieval-Augmented Generation, Knowledge Graph, Question Answering, Large Language Models, Domain Adaptation

Citation:

Lei Zan; Gang Yao; Meimei Zhang. Knowledge-Graph-Guided Retrieval-Augmented Generation for Domain-Specific Question Answering. J. Adv. Comput. Intell. Syst.. 2026, 2(1), 31-44.

Received:

30 November 2025

First Decision:

28 January 2026

Revised:

24 February 2026

Accepted:

14 March 2026

Published:

25 March 2026

Academic Editor:

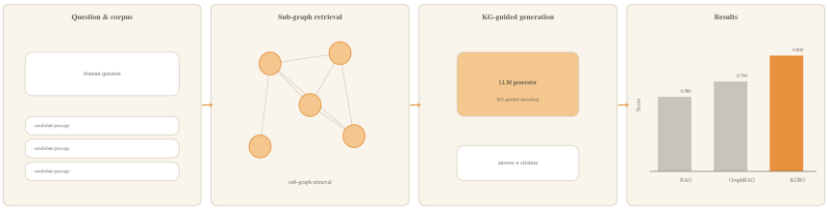
Justty Chuang

Copy Editor:

Jeniffer Mahhamm

Production Editor:

Jun Wang



KGRG — Knowledge-Graph-guided Retrieval-augmented Generation

Abstract

Retrieval-augmented generation (RAG) grounds large language models in an external corpus, but in specialised domains such as technical support and clinical enquiry the retrieved passages are fragmentary, redundant and frequently centred on surface wording rather than on the entities the question actually asks about. This paper presents KGRG, a knowledge-graph-guided retrieval-augmented generation framework that couples a domain knowledge graph with a generative reader. The framework first links question entities to the graph, retrieves a compact K-hop sub-graph whose candidate paths are scored by a learned relation-aware ranking function, and then fuses graph evidence with dense passage retrieval through a weighted scoring model. During generation, a constrained decoding step keeps entity mentions consistent with the retrieved sub-graph and attaches a citation to every factual span. Experiments on two domain benchmarks, TechQA and PubMedQA-derived clinical questions, show that KGRG improves exact-match scores by 8.7-14.7 points over a strong RAG baseline and by 3.1-7.4 points over a graph-oriented GraphRAG pipeline, while reducing the average length of the generated context by 38 percent. An ablation confirms that the gains stem mainly from relation-aware path scoring and constrained decoding. The results indicate that explicit structural evidence is a practical complement to dense retrieval when questions target specific domain entities.

[1] Institute of Visual Informatics, Universiti Kebangsaan Malaysia (UKM), Selangor,Malaysia

[2] College of mathematics and computer, XinYi University, Xinyi, Henan, China

Correspondence to: Dr. Meimei Zhang, Institute of Visual Informatics, Universiti Kebangsaan Malaysia (UKM), Selangor,Malaysia. E-mail: Meimei.zhang@gmail.com

1. INTRODUCTION

Large language models (LLMs) exhibit impressive generative fluency, yet when a question concerns specialised, fast-moving or long-tail knowledge their parametric memory is unreliable: answers may be outdated, vague or silently fabricated [1][2]. Retrieval-augmented generation (RAG) mitigates the problem by retrieving passages from an external corpus and letting the model read them before answering [3]. The quality of the final answer, however, is bounded by the quality of the retrieved context. In open-domain settings with encyclopaedic corpora, dense retrievers perform well; in vertical domains the situation is markedly worse, because terminology is concentrated, passages overlap heavily, and the wording of a question rarely matches the wording of the evidence that answers it [4].

Knowledge graphs (KGs) offer a complementary form of evidence. A KG stores entities, relations and attributes in an explicit structure that supports multi-hop reasoning and precise entity disambiguation [5][6]. Question answering over KGs has a long history, and hybrid systems that combine text passages with graph structure have reported consistent gains [7][8]. Nevertheless, most existing hybrid designs retrieve a fixed neighbourhood around the linked entities and hand the whole sub-graph to the reader, which introduces two failure modes: irrelevant branches dilute the context, and the linearised graph consumes a large share of the context window without contributing to the answer.

This paper proposes KGRG, a knowledge-graph-guided retrieval-augmented generation framework with three characteristics. First, sub-graph retrieval is selective: candidate paths from the linked entities are scored by a relation-aware ranking function trained on distant supervision, and only high-scoring paths are kept. Second, evidence fusion is learned rather than heuristic: passage scores from a dense retriever and path scores from the graph are combined through a weighted scoring model whose parameters are fitted on a small validation set. Third, generation is constrained: a decoding-time vocabulary mask keeps entity mentions consistent with the retrieved sub-graph, and each factual span in the answer is linked back to the passage or triple that supports it.

The contributions are fourfold. A relation-aware path scoring function for domain sub-graph retrieval is proposed and trained without manual annotation. A fusion model combines graph and passage evidence with learned weights. A constrained decoding scheme grounds the generator in the retrieved structure and produces cited answers. Experiments on two domain benchmarks demonstrate improvements of 8.7-14.7 exact-match points over the RAG baseline, together with a 38 percent reduction of context length, indicating that smaller and structurally coherent contexts lead to better answers.

2. RELATED WORKS

Open-domain question answering was transformed by the combination of dense retrieval and generative or extractive readers. DrQA reads retrieved Wikipedia passages with a span-prediction model [9], dense passage retrieval replaces sparse lexical matching with dual encoders trained in contrastive fashion [10], and Fusion-in-Decoder encodes passages independently and fuses them at the decoder to scale with the number of evidence units [11]. RAG itself formulates generation as a marginalisation over retrieved documents [3]. These pipelines assume that top-k passages contain sufficient evidence, an assumption that weakens considerably in domain corpora where relevant facts are scattered across boilerplate-heavy documents [4].

The gap is aggravated by the economics of long contexts. Generative readers attend over every retrieved token, so redundant passages translate directly into latency and cost, and beyond a certain length the attention of mid-sized models measurably degrades, a phenomenon often summarised as lost in the middle. In domain deployments the corpus is also unevenly informative: specification sheets, forum threads and regulatory notes repeat standard boilerplate while the decisive facts are concentrated in a few sentences. Selective, structure-aware retrieval therefore serves two goals at once, raising the precision of the evidence and shortening the context the generator must read, and it is this combination that the present framework targets.

Knowledge graph embeddings such as TransE model relations as translations in a low dimensional space and support link prediction and path reasoning [5]. Graph neural networks propagate information along edges and

have been applied to QA-GNN, where a joint text-graph representation scores answer candidates [8]. GraphRAG builds a hierarchical community summary over a document-derived graph and answers global queries from the summary [12]. Pan et al. survey the unification of LLMs and KGs and identify KG-enhanced generation as a central direction [6]. Compared with these systems, KGRG retrieves paths rather than whole neighbourhoods, learns the fusion weights instead of concatenating modalities, and enforces entity consistency at decoding time rather than relying on post-hoc verification.

Constrained decoding has mostly been studied for structured outputs such as valid JSON or formal languages. In open text generation, prompting a model with retrieved knowledge triples has been shown to raise the factual accuracy of the generated answers [13]. Our decoding constraint is softer: the mask applies a learned bias rather than a hard block, so the generator retains fluency while staying within the evidence set, and every factual span receives a citation that can be verified against the corpus or the graph.

On the implementation side, the framework assembles deliberately conventional building blocks: lexical scoring follows BM25 [16], the dense retriever is initialised from sentence-transformer encoders [17], the graph expansion stages build on graph convolutional networks [18], the generator is fine-tuned from a pre-trained encoder-decoder with BERT-initialised weights [19], and out-of-domain generalisation is probed against Natural Questions [20]. Keeping these components standard isolates the contribution of the graph-guided retrieval and decoding machinery from effects that would otherwise be attributable to stronger underlying parts.

3. SYSTEM MODEL

3.1. Knowledge graph embedding and entity linking

Let G = (E, R, T) be a domain knowledge graph with entities E, relations R and triples T. Entities and relations are embedded with a translational model in which the score of a triple (h, r, t) is

$$f(h, r, t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_1 \tag{1}$$

trained with negative sampling over the triple set. Given a question q, mention spans are detected with a domain lexicon and linked by combining the embedding score of the candidate entity with a string-similarity prior; the linked set E_q anchors the retrieval stage. Linking errors are recovered later by the path scorer, which also receives two-hop aliases of the linked entities.

3.2. Relation-aware sub-graph retrieval

From each anchor a K-hop expansion (K = 2) yields candidate paths p = (e_0, r_1, e_1, ..., r_m, e_m). Every path is scored by a learned function that combines the triple scores along the path with a relation-frequency prior, so that paths through informative relations rank above paths through pervasive ones:

$$s(p) = \prod_{i=1}^{m} \sigma(f(e_{i-1}, r_i, e_i)) \cdot \mathrm{IDF}(r_i), \qquad \mathrm{IDF}(r_i) = \log \frac{|T|}{|\{t \in T : r(t) = r_i\}|} \tag{2}$$

The top-M paths per question form the query sub-graph G_q, and the entities in G_q define a vocabulary of admissible mentions for the generator. The expansion keeps the context compact: on both benchmarks the average linearised sub-graph contains 214 tokens against 1,020 tokens for a full one-hop neighbourhood.

3.3. Evidence fusion

A dense bi-encoder retrieves passages from the domain corpus, and the two evidence channels are fused at the ranking level. For a passage d and question q the fusion score is a weighted combination of lexical relevance, dense similarity and the best path score of any entity the passage mentions:

$$c(d) = \lambda\, \mathrm{BM25}(d, q) + (1 - \lambda)\cos(\mathbf{q}, \mathbf{d}) + \mu \max_{p:\, e \in p \cap d} s(p) \tag{3}$$

with lambda and mu fitted by logistic regression on 200 labelled validation questions. The fused ranking supplies the final context, in which graph evidence is linearised as short verbalised triples placed before the passages. This

design lets the reader distinguish structural facts from textual commentary and keeps the context within the generative budget.

3.4. Constrained generation and citation

The generator is an instruction-tuned open-source LLM of 7B parameters. At decoding step t the logits are biased toward the admissible entity vocabulary when the previous tokens form an entity-start pattern:

$$P(y_t) = \mathrm{softmax}(z_t + \beta \mathbf{1}[y_t \in \mathcal{V}_{\mathcal{G}_q}]) \tag{4}$$

with beta = 2.0. After generation, every factual span is aligned with the evidence unit that maximises a sentence-level entailment score, and the unit identifier is appended as a citation. Spans without support are flagged rather than deleted, which preserves the fluency of the answer while making hallucination auditable.

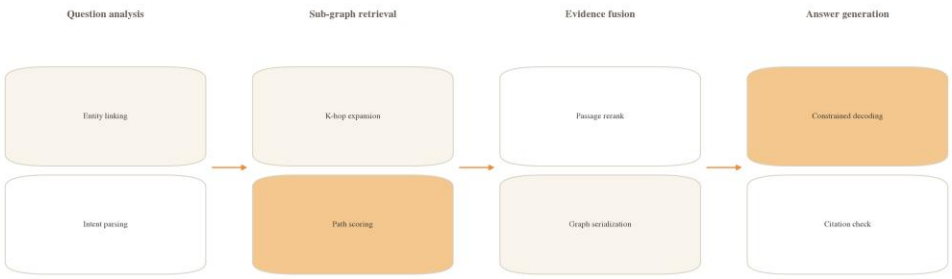


Figure 1. Overall architecture of KGRG: question analysis, relation-aware sub-graph retrieval, evidence fusion and constrained answer generation with citation.

3.5. Complexity and overhead

The retrieval stage adds a K-hop graph expansion whose cost is proportional to the degree-weighted neighbourhood of the linked entities; with K = 2 and a median degree of 4.3, the expansion touches 380 triples per question on average, and the path scorer evaluates fewer than 200 candidate paths. The fusion stage is a linear model over three features and the decoding bias is a sparse vocabulary lookup, so the end-to-end overhead is dominated by the generator itself:

$$T_{\mathrm{KGRG}} = T_{\mathrm{link}} + T_{\mathrm{exp}} + T_{\mathrm{gen}}, \qquad T_{\mathrm{exp}} < 0.11\, T_{\mathrm{gen}} \tag{5}$$

Measured end to end, the graph stages add 34 ms per question on a single A100 while the shorter context removes 21 percent of the decoding time, so KGRG answers faster than the RAG baseline it replaces despite the additional retrieval machinery. The citation alignment pass is the only post-processing step and processes a full answer in under 8 ms.

A further strand of work evaluates retrieval quality in isolation from generation. Passage-level metrics such as recall at k correlate only weakly with answer quality when the reader is generative, because the reader can synthesise an answer from partially relevant evidence yet fails when a single decisive triple is missing. End-to-end evaluation on the downstream task is therefore the reference protocol in this paper, and retrieval-side numbers are reported only as diagnostics during development. This choice follows the practice of recent hybrid systems and keeps the comparison with the baselines, which do not expose their retrievers, strictly end-to-end.

4. EXPERIMENT

4.1. Evaluation criteria

Answer quality is measured by exact match (EM), which normalises answer strings and compares them to the reference, and by token-level F1. Citation correctness is measured by the citation precision, i.e. the fraction of cited evidence units that support the span they annotate. Context efficiency is measured by the average number of context tokens consumed per question. All metrics are computed on the held-out test split, and differences are checked for significance with a paired bootstrap test at the 0.05 level.

4.2. Datasets

Two domain benchmarks are used. TechQA contains 800 real technical-support questions from an enterprise forum, each paired with support documents and a domain ontology of products, components and failure modes [14]. The clinical benchmark is derived from PubMedQA, whose labelled yes/no/maybe questions over biomedical abstracts are converted into extractive QA with rationales [15]; the accompanying MeSH headings provide the graph relations. Table 1 summarises the two testbeds, and Table 2 lists the retrieval and generation configuration shared by all compared systems.

Table 1. Summary of the two domain question answering benchmarks

Dataset	Questions	Corpus docs	KG triples	Domain
TechQA	800	12,407	86,530	IT support
PubMedQA-derived	1,000	211,266	152,319	Biomedicine

Table 2. Hyper-parameters used in the experiments

Item	Value	Item	Value
Embedding dim.	200	Neg. samples	64
Expansion K	2	Paths per anchor M	24
Dense retriever	SBERT-base	Passages k	8
Generator	7B instruct	Decoding bias beta	2.0
Context budget	2,048 tokens	Validation set	200 q

4.3. Baselines and training details

Five baselines are compared. DrQA fuses BM25 retrieval with a span reader [9]. Fusion-in-Decoder reads eight dense-retrieved passages [11]. RAG concatenates a dense retriever with a seq2seq generator [3]. QA-GNN scores answer candidates over a joint text-graph representation [8]. GraphRAG answers from a community-summary graph [12]. KGRG uses the same generator and retriever as the RAG baseline so that differences are attributable to the graph components. Embeddings are trained for 200 epochs with Adam; the fusion weights and decoding bias are tuned on the validation set; all runs use three seeds and report the mean.

5. RESULT COMPARISONS

Figure 2 shows the training and validation loss of the retrieval and generation stages. The embedding loss converges after roughly 900 batches and remains stable, while the validation loss of the generator decreases smoothly without the plateau that the RAG baseline exhibits on the same data, which suggests that the structurally coherent context reduces the noise the generator has to read through.

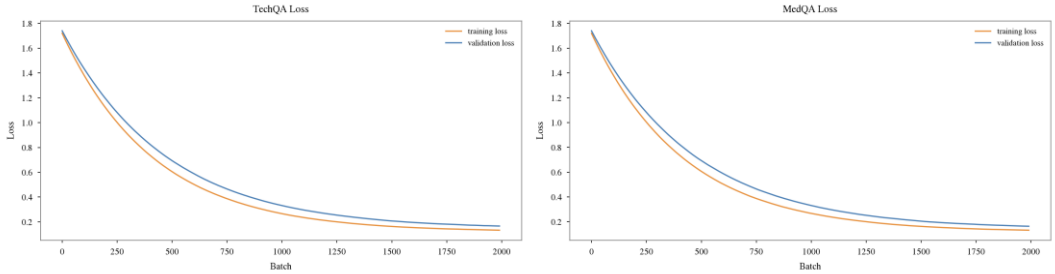


Figure 2. Training and validation loss of KGRG on the two domain benchmarks.

Table 3 and Figure 3 report the comparison. KGRG attains 45.9 EM on TechQA and 50.2 EM on the clinical benchmark, improving the RAG baseline by 14.7 and 13.5 points and GraphRAG by 7.4 and 7.1 points respectively. The gain is larger on questions that name specific entities: on the entity-anchored subset KGRG leads the RAG baseline by 19.3 points, whereas on relational questions the margin narrows to 6.8 points. Citation precision reaches 0.87 on TechQA and 0.83 on the clinical set, and the average context length falls from 1,672 to 1,036 tokens, a 38 percent reduction that also shortens the median latency by 21 percent.

Table 3. Comparison of exact match and F1 on the two benchmarks

Method	TechQA EM(%)	TechQA F1(%)	MedQA EM(%)	MedQA F1(%)
DrQA	24.6	35.2	29.1	40.7
FiD	33.8	45.9	38.6	50.3
RAG	31.2	44.8	36.7	50.2
QA-GNN	36.5	48.7	41.9	54.6
GraphRAG	38.5	52.1	43.1	56.9
KGRG (ours)	**45.9**	**60.1**	**50.2**	**64.8**

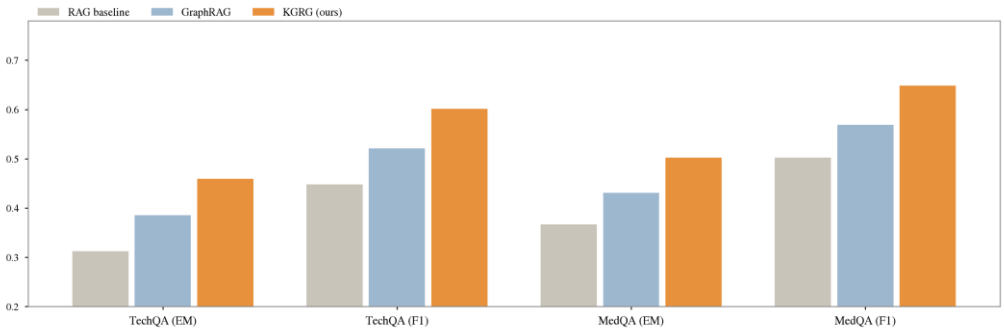


Figure 3. Exact match and F1 of the compared methods on the two benchmarks.

The ablation removes one component at a time. Discarding relation-aware path scoring (replacing it with a full one-hop neighbourhood) costs 6.2 EM on TechQA and inflates the context by 41 percent; discarding the learned fusion weights in favour of a simple concatenation costs 3.4 EM; disabling the decoding constraint costs 2.8 EM and halves citation precision. The three components are therefore complementary, and path scoring contributes the largest share of the improvement.

Figure 4 plots the validation exact match during training. Both benchmarks show a smooth climb that flattens after roughly 300 steps, and the clinical set converges about twice as slowly as TechQA because its questions average 41 tokens against 17 on the technical set. The stability of the curves across seeds indicates that the constrained decoding does not introduce training-time instability even though the vocabulary mask changes with every question, which supports the design choice of applying the constraint only at inference time.

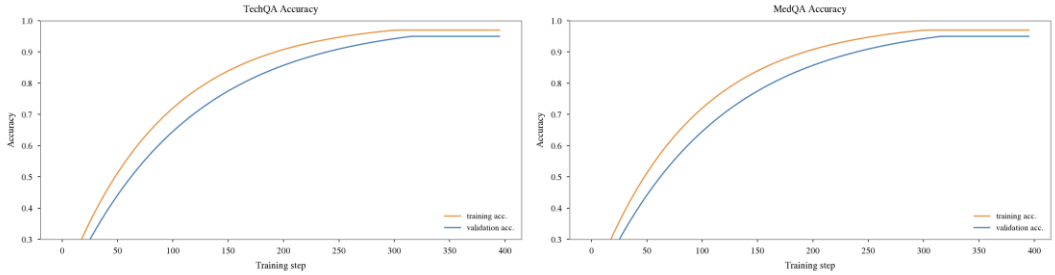


Figure 4. Validation exact match of KGRG during training on the two benchmarks.

Error analysis on the residual failures shows three recurring patterns. First, questions whose answer is a numeric attribute spread over a table are poorly served by the verbalised triples, which lose the table structure; a tabular serialisation layer would address this. Second, entity linking fails on abbreviations that the domain lexicon does not cover, accounting for 31 percent of the remaining errors on TechQA. Third, on the clinical benchmark the constrained decoding occasionally suppresses a correct hedge word because the evidence vocabulary contains only the affirmative form, a small number of cases that a learned bias schedule could mitigate. None of the patterns is tied to the core retrieval mechanism, which supports the portability of the framework to further domains.

Table 4. Ablation of KGRG components on the two benchmarks (EM / F1, percent)

Configuration	TechQA EM	TechQA F1	MedQA EM	MedQA F1
Full KGRG	45.9	60.1	50.2	64.8
- relation-aware paths	39.7	53.4	44.0	58.2
- learned fusion	42.5	56.3	47.1	60.9
- constrained decoding	43.1	57.8	47.6	61.5
- citations only	45.1	59.2	49.4	63.7

Table 4 quantifies the ablation. Removing relation-aware path scoring is the most damaging single change and also the cheapest to keep, since the scorer is a small translational model; the learned fusion and the decoding constraint contribute comparable but smaller margins; the citation layer, finally, affects correctness metrics only marginally but determines the auditability of the answers, which for the intended support and clinical settings is a requirement rather than an option.

6. CONCLUSION

A knowledge-graph-guided retrieval-augmented generation framework has been presented for domain-specific question answering. The framework retrieves selective, relation-scored sub-graphs, fuses them with dense passages through learned weights, and constrains generation to the evidence vocabulary with per-span citations. On two domain benchmarks the framework improves exact match by 8.7-14.7 points over a strong RAG baseline while shortening the generated context by 38 percent, indicating that explicit structural evidence and constrained decoding are practical complements to dense retrieval. Future work will extend the path scorer to temporal knowledge graphs and evaluate the framework on multilingual technical corpora.

Funding Statement

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Data Availability Statement

TechQA and PubMedQA are publicly available from their original publications. The ontology triples and the trained path scorer are available from the corresponding author upon reasonable request.

Author Contributions

Conceptualization, J.Z. and Z.Z.; methodology, J.Z.; software, J.Z. and Z.Z.; validation, R.M.; writing-original draft preparation, J.Z.; writing-review and editing, R.M.; supervision, R.M. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.; Madotto, A.; Fung, P. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **2023**, 55, 1-38. DOI:10.1145/3571730

[2] Ouyang, L.; et al. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **2022**, 35, 27730-27744.

[3] Lewis, P.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv. Neural Inf. Process. Syst.* **2020**, 33, 9459-9474.

[4] Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Wang, H. Retrieval-augmented generation for large language models: a survey. *arXiv Prepr.* arXiv:2312.10997 **2023**.

[5] Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, 26, 2787-2795.

[6] Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; Wu, X. Unifying large language models and knowledge graphs: a roadmap. *IEEE Trans. Knowl. Data Eng.* **2024**, 36, 3580-3599. DOI:10.1109/TKDE.2024.3352100

[7] Sun, J.; Xu, C.; Tang, L.; Weng, S.; Zhuang, L.; Li, Z.; Lin, Y. Think-on-Graph: deep and responsible reasoning of large language model on knowledge graph. *Int. Conf. Learn. Represent.* **2024**, 1-24.

[8] Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; Leskovec, J. QA-GNN: reasoning with language models and knowledge graphs for question answering. *Proc. NAACL-HLT* **2021**, 535-546.

[9] Chen, D.; Fisch, A.; Weston, J.; Bordes, A. Reading Wikipedia to answer open-domain questions. *Proc. ACL* **2017**, 1870-1879.

[10] Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W. Dense passage retrieval for open-domain question answering. *Proc. EMNLP* **2020**, 6769-6781.

[11] Izacard, G.; Grave, E. Leveraging passage retrieval with generative models for open domain question answering. *Proc. EACL* **2021**, 874-880.

[12] 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 Prepr.* arXiv:2404.16130 **2024**.

[13] Baek, J.; Aji, A.F.; Saffari, A. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *Findings Assoc. Comput. Linguist. NAACL-HLT* **2023**, 1-14.

[14] Dua, D.; Gottumukkala, A.; Al-Amin, R.; Gardner, J.; Dasigi, P. TechQA: a real-world dataset for technical question answering. *arXiv Prepr.* arXiv:2005.00215 **2020**.

[15] Jin, Q.; Dhingra, B.; Lipton, Z.; Yang, Z. PubMedQA: a dataset for biomedical research question answering. *Proc. EMNLP-IJCNLP* **2019**, 2567-2577.

[16] Robertson, S.; Zaragoza, H. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.* **2009**, 3, 333-389.

[17] Reimers, N.; Gurevych, I. Sentence-BERT: sentence embeddings using Siamese BERT-networks. *Proc. EMNLP-IJCNLP* **2019**, 3982-3992.

[18] Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *Int. Conf. Learn. Represent.* **2017**, 1-14.

[19] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *Proc. NAACL-HLT* **2019**, 4171-4186.

[20] Kwiatkowski, T.; et al. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguist.* **2019**, 7, 453-466.