VectorGraphRAG: Automatic Knowledge Graph Construction and Memory Efficient Triplet Ranking for Medical Question Answering

Nikolaos Giarelis[1,*], Charalampos Mastrokostas[1] and Nikos Karacapilidis[1]

[1]Industrial Management and Information Systems Lab, MEAD, University of Patras, Greece

Abstract

This work introduces a novel Graph-based Retrieval-Augmented Generation (GraphRAG) approach, which enhances the accuracy of Large Language Models (LLMs) on the medical Question Answering (QA) task. The proposed approach aims to alleviate limitations of previous ones, such as the overreliance on facts from outdated knowledge bases, the adherence to specialized ontologies, and the lack of utilization of performance-oriented techniques (e.g., model weight quantization). To do so, (i) it automatically extracts factual triplets from medical questions using a state-of-the-art LLM, while allowing fact verification by medical experts; (ii) it connects the entities from these factual triplets into a Knowledge Graph (KG); (iii) it represents graph entities using pre-trained transformer embeddings stored in an in-memory vector framework, and (iv) it leverages these embeddings, firstly by finding similar KG entities to the extracted terms of the user question, and secondly by semantically ranking the *top-n* most relevant factual triplets to the user question, which are then used as enriched context for the LLM. To verify the validity of our approach, we conduct a series of experiments using a prominent medical QA dataset (MedMCQA) to measure the accuracy of our approach by incorporating various open source LLMs. In addition to these experiments, we investigate the effect of different context lengths on the accuracy achieved. To the best of our knowledge, this is the first GraphRAG approach that empirically measures the accuracy of LLMs on QA, while utilizing 4-bit quantization. It is shown that our approach improves model accuracy up to +5.5% over the baseline ones, while it requires significantly less computational resources compared to previous approaches. To enable the reproducibility of our work, we make the extracted triplets and code publicly available.

Keywords

Knowledge Graphs, Large Language Models, Retrieval Augmented Generation, Question Answering, Medical Domain

1. Introduction

Recent advancements in the field of Deep Learning have led to the development of Large Language Models (LLMs), which have general-purpose language understanding and reasoning capabilities [1] and have managed to significantly advance the state-of-the-art performance in many Natural Language Processing (NLP) tasks [2]. However, despite their capabilities, LLMs face a series of challenges when handling domain-specific questions, due to the lack of domain knowledge and the outdated information often used [3]. A particular domain where LLMs face such challenges is the medical one [4, 5].

To address these challenges, several strategies have already been proposed [1, 2]. Most of them rely on different prompting techniques with increased context, while some continue the pre-training or post-training (fine-tuning) of these models. In any case, LLM training requires expensive hardware (i.e., large VRAM GPUs) due to memory requirements, as well as a significant amount of computational time. On the other hand, context-enabled strategies, such as Retrieval Augmented Generation (RAG), retrieve relevant information from an external knowledge base and then supply this context to the LLM to improve its response. However, RAG strategies require more memory and computational time at inference (instead of training), due to the longer context presented to the model. This limitation is inherited from the attention mechanism of the *Transformer* architecture [6], which is the building block of LLMs.

A major drawback of common RAG approaches is that they retrieve independent text passages that are relevant to the users' question [7, 8] without considering semantic connections, which exist between entities from the retrieved passages or the entire textual corpus. To overcome this drawback, recent approaches - known as GraphRAG - integrate entities extracted from text and their connections into a semantic graph [9, 7, 10]. Specifically, GraphRAG approaches build on the concept of Knowledge Graphs (KGs), which represent real world entities and their connections, organized in semantic graphs [11]. A recent survey [12] highlighted that LLMs can benefit from their integration with KGs, by leveraging their structured knowledge to improve their accuracy and reduce erroneous answers due to model hallucinations. Additionally, other studies have leveraged the generative capabilities of LLMs to construct a biomedical KG [13] and develop an approach for graph entity linking [14].

In any case, several GraphRAG approaches demonstrate a series of limitations: (i) they rely on existing KGs, without addressing the phase of graph construction from unstructured data, which requires great manual effort by domain experts; (ii) they often utilize KGs with outdated domain knowledge [15] (e.g., medical KGs introduced before the Covid-19 era); (iii) during information retrieval, several approaches traverse the KG to retrieve entities that match exactly those existing in the user question; this disregards highly similar or synonymous entities that are not included in the initial query; (iv) they do not employ memory optimization techniques, such as LLM model weight quantization [16]; they store the entire KG in memory instead of utilizing a scalable graph database, such as *Neo4j* [17]; (v) they do not examine the "accuracy vs. performance" trade-off, which occurs when limiting the context provided by the knowledge base.

Focusing on the medical QA task, the aim of this paper is to develop and assess an accurate and efficient GraphRAG approach that addresses the above limitations. The proposed approach automatically creates KG triplets from unstructured text (question-answer pairs) and stores them in a scal-

ANSyA 2025: 1st International Workshop on Advanced Neuro-Symbolic Applications, co-located with ECAI 2025.

*Corresponding author.

giarelis@ceid.upatras.gr (N. Giarelis); cmastrokostas@ac.upatras.gr (C. Mastrokostas); karacap@upatras.gr (N. Karacapilidis)
0000-0003-2611-3129 (N. Giarelis); 0009-0009-6143-4666 (C. Mastrokostas); 0000-0002-6581-6831 (N. Karacapilidis)

able graph database. These triplets can be retrieved from the KG and used as additional context by an open source LLM that has small memory requirements due to quantization. The contributions of this paper are the following:

- we introduce a novel GraphRAG approach, namely *VectorGraphRAG*, that does not rely on specialized ontologies or existing medical KGs, which often maintain outdated domain knowledge [15];
- we propose an automatic KG construction process, which extracts factual triplets from QA pairs by leveraging a state-of-the-art LLM;
- we store these triplets in a human readable format (.csv) to facilitate their evaluation and possible amendment by medical experts;
- we perform a series of experiments using different prompting strategies and several open source LLMs on a prominent medical QA dataset, namely *MedMCQA* [18], aiming to elaborate and answer a set of research questions (**RQs**); specifically:
  - **RQ1:** Does the automatic creation of triplets using *GPT-4o mini*[1] from QA pairs achieve close or better accuracy than the state-of-the-art approach [19], without relying on existing medical KGs with specialized ontologies?
  - **RQ2:** Is there a meaningful accuracy drop for medical QA when using quantized LLMs (4-bit precision) vs. unquantized ones (16-bit precision)?
  - **RQ3:** How does *VectorGraphRAG* compare against other approaches that utilize different prompting strategies in terms of accuracy?
  - **RQ4:** Can we maximize the LLM accuracy while minimizing the retrieved KG context (as measured by the number of triplets)?

The remainder of this paper is organized as follows: state-of-the-art LLMs and medical GraphRAG approaches are described in Section 2, highlighting their benefits and limitations; the proposed approach is described in Section 3; our experimental setup and results are presented in Section 4; finally, concluding remarks and future research directions are outlined in Section 5. To facilitate the reproducibility of our work, we make our code and extracted triplets publicly available[2].

2. Related Work

2.1. Large Language Models

LLMs can handle various NLP tasks, due to their autoregressive architecture, extensive training on vast amounts of data, and alignment with human instructions through reinforcement learning [2]. Many prompt engineering techniques exist for these models, such as *zero-shot* and *few-shot learning* [20] for simple tasks, or *Chain-of-Thought* (CoT) for complex reasoning tasks [1]. For domain-specific tasks, LLMs can be improved with specialized knowledge from external sources using RAG [3].

So far, several proprietary and open-source LLMs have been proposed. Proprietary models, such as OpenAI's *ChatGPT* [20], offer state-of-the-art performance in many tasks but are costly, let alone the fact that their use in sensitive applications requiring data privacy is not possible in the EU due to GDPR regulations [21]. On the other hand, open-source models can be deployed locally at minimal cost, thus making them suitable for privacy-sensitive applications.

In this study, we consider a series of remarkable open-source families of models that achieve state-of-the-art accuracy in several benchmark tests reported in the literature. These include *Llama 3* [22], *Gemma 2* [23], *Mistral* [24], *Command R* [25] and *Falcon* [26]. Each of these families has several models of different parameter sizes. The smaller versions can be very efficient, especially when combined with RAG techniques that can improve accuracy without requiring large VRAM GPUs for inference.

2.2. Medical QA works

HyKGE [27] is an approach that combines LLMs and KGs for medical QA. It builds a medical KG by combining existing ones and adding entity descriptions from encyclopedic knowledge bases. It prompts the LLM to generate an initial answer, extracts terms from this answer and the user query, and utilizes them to retrieve graph paths from the KG. The graph paths are ranked using a pretrained re-ranker model and the *top-k* most relevant ones to the user's query are selected. The graph paths are then used as reasoning context for the LLM to generate the final answer. Experimental results indicate that *HyKGE* surpasses the accuracy of previous RAG models in medical QA. However, the authors do not make their code publicly available.

Labrak et al. [28] evaluate the capabilities of several LLMs on various medical and clinical NLP tasks. Their findings indicate that LLMs can sometimes handle such tasks without domain-specific knowledge, using *zero-shot* and *few-shot* prompting strategies. A major limitation of this work is that it does not provide any specialized medical knowledge to the LLMs - either through RAG or fine-tuning - to improve their performance.

Bailicai [29] is a multi-component GraphRAG approach for medical QA. The first component prompts the LLM to determine if it can answer the user's query with its internal knowledge; if the LLM answers positively, the next components are skipped. The second component utilizes the LLM to break down complex queries into subtasks, which are modelled into a directed acyclic dependency graph. By considering this graph, the LLM solves each subtask; these intermediate answers are used as context for the final answer of this step. The third component fine-tunes the LLM to identify relevant medical documents. The fourth component utilizes an in-memory vector index to retrieve the most relevant paragraphs to the user query from multiple literature sources. A major limitation of *Bailicai* is the amount of memory required to store the vector index ($\sim$181.7 GB). Apart from this memory intensive RAG component, this approach also uses multiple LLM generations and fine-tuning, which make it computationally expensive.

MedGraphRAG [19] improves LLM accuracy in medical QA by utilizing a multi-layer KG from multiple sources. Similarly to Microsoft's Graph RAG, it generates summaries from graphs to improve the retrieval step. Nevertheless, it uses structured summaries with predefined tags (e.g., *"Symptoms", "Patient_History",* etc.). *MedGraphRAG* compares these graph summaries with the user's query summary to find the most relevant KG triplets, which are then utilized as context to generate and refine the final answer. Experi-

[1]https://platform.openai.com/docs/models/gpt-4o-mini
[2]https://github.com/NC0DER/VectorGraphRAG

mental results showed that *MedGraphRAG* achieves better accuracy than baseline RAG approaches. However, a drawback of this approach is that it performs numerous LLM generations for each subgraph, making it computationally expensive. In addition, it relies on pre-existing KGs that may contain outdated medical information.

DEEB-RAG [30] is a biomedical QA approach that utilizes RAG to retrieve the *top-k* most relevant documents to a user question. It uses a pre-trained transformer-based model to encode the retrieved documents into embeddings, which are processed by a trained two-layer perceptron to be aligned with the dimensional space of the LLM (i.e., Llama-2 [31]) hidden layers. The aligned vectors are combined with the original token-level embeddings from questions and documents, and they are fed to the LLM to generate the final answer. Related experiments reveal that *DEEB-RAG* performs better than naive RAG across various datasets. A major limitation of *DEEB-RAG* is that it leverages an outdated LLM architecture.

BioKGQA [13] is an approach that automatically constructs a biomedical QA dataset by extracting factual triplets from KGs and leverages LLMs to generate questions from these triplets. It has been applied to create a dataset that contains 85,368 QA pairs alongside the *SPARQL* queries utilized to extract KG facts. A series of experiments has been performed to measure the quality of the generated questions from various LLMs. However, the extracted KG context is not utilized for RAG or LLM fine-tuning, while the overall approach does not address the automatic KG construction step from unstructured data or the integration step from multiple KGs.

3. VectorGraphRAG: The Proposed Approach

VectorGraphRAG includes two multi-step processes. The first one concerns the extraction and representation of knowledge (see Section 3.2 and Figure 1), while the second one is the retrieval of the appropriate KG triplets to be used for RAG (see Section 3.3 and Figure 2).

3.1. Preliminaries

Definition 1: Semantic entity and relationship. We define a semantic entity as e_i, where $e_i \in E$ and a semantic relationship as r_i, where $r_i \in R$; E and R are the sets of extracted entities and their relationships (predicates), respectively.

Definition 2: Semantic Triplet. We define this triplet as $t_i{=}(e_i{,}r_k{,}e_j)$, where $i{\neq}j$ and $t_i{\in}T$. e_i and e_j are the head and tail entities, respectively, and r_k is the predicate of these entities. Finally, $T{=}\{t_1, ..., t_N\}$ is the set of semantic (factual) triplets, where $T{\subseteq} E{\times}R$.

Definition 3: Knowledge Graph (KG). We define an attributed graph $KG{=}(V, G_E, E, R)$, where $V{=}\{v_i, ..., v_N\}$ is the set of graph vertices and $G_E{\subseteq}V\times V$ is the set of graph edges. We also define a function ϕ that maps semantic entities and predicates to graph vertices and edges, respectively:

$$\phi = \{e_i{\to}v_i, r_i{\to}g_{E_i}\ \forall v_i \in V, e_i{\in}E, r_i{\in}R, g_{E_i}{\in}G_E\} \quad (1)$$

Definition 4: Semantic vector (embedding) computation. We define an encoding function $X = \{e_i \to x_i\ \forall e_i \in E\}$, where $x_i{\in}R^d$ is a semantic vector of dimension d, which comprises real value coefficients. This vector semantically represents a textual sequence by leveraging a pre-trained transformers model.

Definition 5: Similarity function. We define a cosine similarity function between two semantic vectors x_i and x_j as:

$$sim(x_i, x_j) = \frac{x_i \times x_j}{||x_i|| \cdot ||x_j||} \quad (2)$$

3.2. Knowledge Extraction and Representation

The first step of this process extracts factual triplets from each question-answer pair, by prompting a state-of-the-art LLM (i.e., *GPT-4o mini*). The second step constructs a .csv file from the set of triplets *T*; this format is selected to facilitate any possible amendments by medical experts. The third step loads the KG data from the file into a graph database. We also place uniqueness constraints on the entity names, to avoid storing duplicates. The fourth step involves the use of a sentence transformers model [32], called *all-mpnet-base-v2*[3], to semantically encode the entities into vectors (embeddings) as described in *Definition 4*. These vectors are saved in a binary *numpy* array file [33] to avoid encoding each time they are required for RAG.

3.3. Ranked Retrieval Augmented Generation

The first step of this process initializes the graph database that contains the KG, and loads three important components into memory: (i) the LLM using 4-bit quantization (GPU); (ii) the sentence transformers model (RAM); (iii) the entity embeddings (RAM). We then build a cosine similarity index (RAM) using *FAISS* [34], an efficient vector similarity framework.

The second step concerns the extraction of search terms from the user question. To do this, we remove stop-words, tokenize the sentences, and extract all *n-grams* from each sentence, where $n \in \{1, 2, 3\}$. Each *n*-gram is encoded using the embedding model.

The third step finds the *top-k* most similar KG entities to the extracted search terms from the similarity index, using the R_E function, where *S* is the set of search terms and s_i the vector embedding for each element of the encoded set.

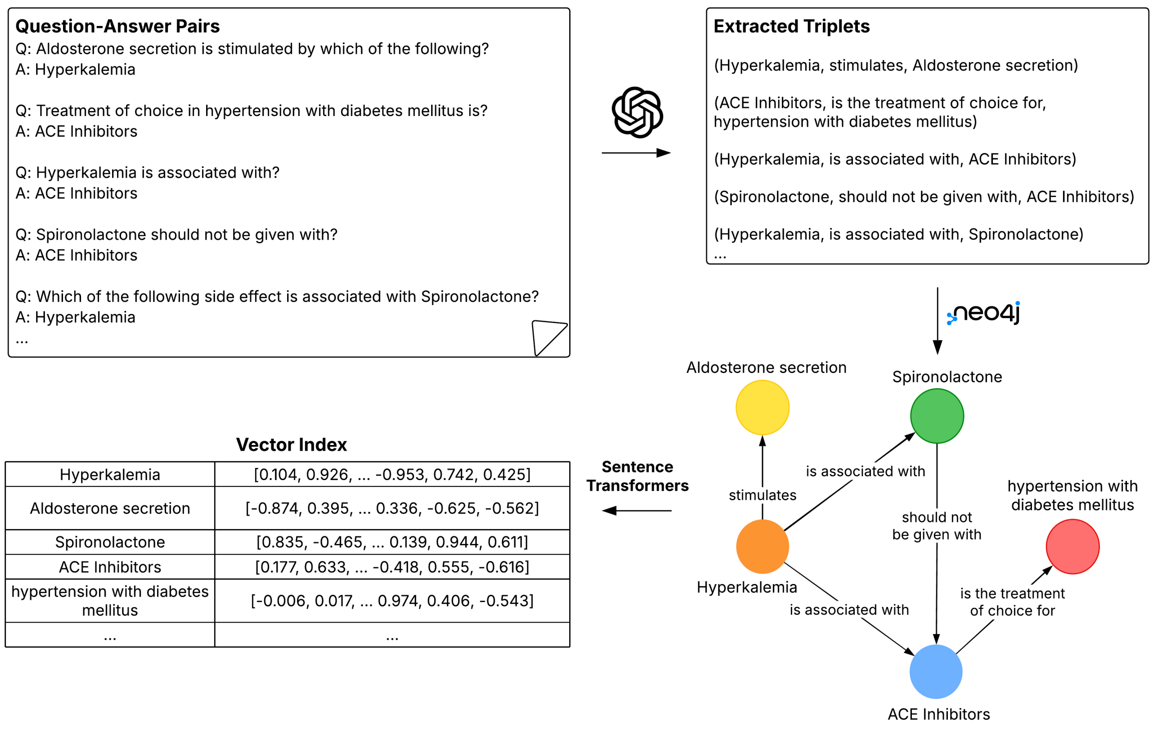
$$R_E = \underset{\substack{x_i \in X(E) \\ \wedge s_i \in X(S)}}{argmax}\,(sim(x_i, s_i) | sim(x_i, s_i) {>} 0.9) \quad (3)$$

The fourth step traverses the graph database to retrieve triplets, whose entities match either the search terms or the *top-k* most similar ones. We then use the sentence transformer model to encode the question *Q* and the triplets *T* to find the *top-n* most similar triplets using the R_T function:

$$R_T = \underset{y_i \in X(T)}{argmax}(sim(y_i{,}X(Q)) | sim(y_i{,}X(Q)){>}0.1) \quad (4)$$

Longer text sequences tend to generate lower similarity scores; thus, the cutoff in equation 4 is much lower in order to avoid dissimilar triplets (i.e., triplets with similarity scores in the range [-1, 0]). The final step utilizes the *top-n* triplets as textual context for the LLM by embedding them in the prompt.

[3] https://sbert.net/


Question-Answer Pairs
Q: Aldosterone secretion is stimulated by which of the following?
A: Hyperkalemia

Q: Treatment of choice in hypertension with diabetes mellitus is?
A: ACE Inhibitors

Q: Hyperkalemia is associated with?
A: ACE Inhibitors

Q: Spironolactone should not be given with?
A: ACE Inhibitors

Q: Which of the following side effect is associated with Spironolactone?
A: Hyperkalemia
...

Extracted Triplets

(Hyperkalemia, stimulates, Aldosterone secretion)

(ACE Inhibitors, is the treatment of choice for, hypertension with diabetes mellitus)

(Hyperkalemia, is associated with, ACE Inhibitors)

(Spironolactone, should not be given with, ACE Inhibitors)

(Hyperkalemia, is associated with, Spironolactone)
...

Vector Index

Hyperkalemia	[0.104, 0.926, ... -0.953, 0.742, 0.425]
Aldosterone secretion	[-0.874, 0.395, ... 0.336, -0.625, -0.562]
Spironolactone	[0.835, -0.465, ... 0.139, 0.944, 0.611]
ACE Inhibitors	[0.177, 0.633, ... -0.418, 0.555, -0.616]
hypertension with diabetes mellitus	[-0.006, 0.017, ... 0.974, 0.406, -0.543]
...	...

Figure 1: Knowledge Extraction and Representation

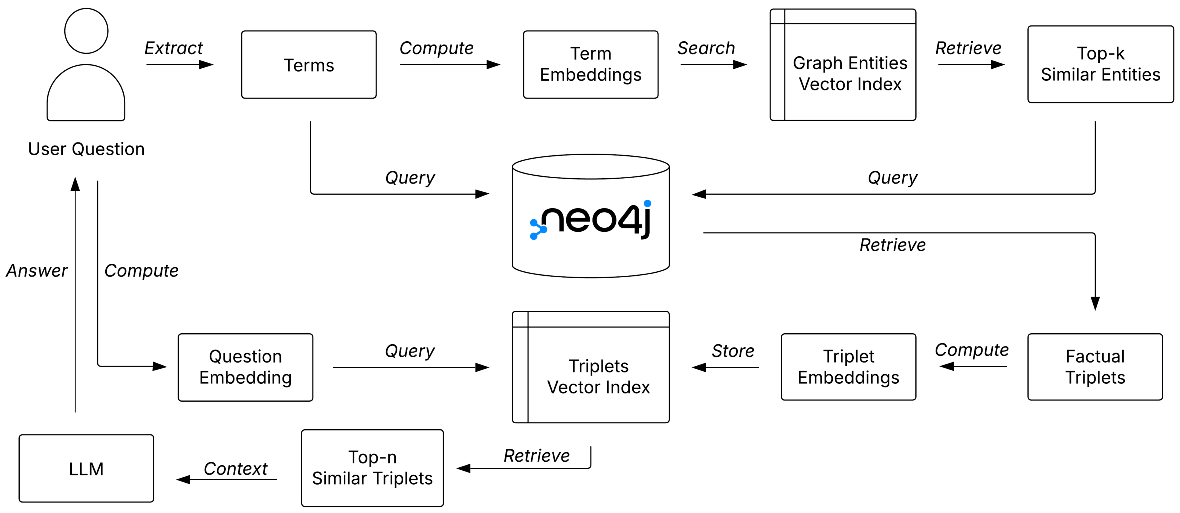


Figure 2: Ranked Retrieval Augmented Generation

4. Experiments

4.1. Experimental Setup

In terms of hardware, we utilize a PC with an Intel Core i9-13900K CPU with 64 GB of RAM and an Nvidia RTX 3060 GPU (12 GB of VRAM). With respect to software, we utilize *HuggingFace Transformers*[4] to develop LLM inference code, and we build our KG with *Neo4j*[5]. As mentioned earlier, we also use *sentence transformers* to encode entities into vectors, which are saved into *numpy* array files. Finally, we extend *FAISS* to calculate the similarity indices.

4.2. Dataset and Knowledge Graph

Similarly to other works, we validated our approach using the *MedMCQA* dataset. Specifically, to build our KG, we utilized its training part, which comprises ∼183K medical questions. We selected a subset of it comprising ∼120K questions, labelled as "single choice", where each choice contains a single option.

Then, we utilized the data extraction prompt (Table 1) and

[4]https://huggingface.co/transformers/
[5]https://neo4j.com/

GPT-4o mini, which was accessed through the OpenAI API[6], to create a semantic triplet from each question-answer pair. A post-processing step was applied to remove empty and malformed triplets (e.g., those without a subject or an object). The data extraction step costed 2.49$ dollars' worth of credits. Alternatively, we could also use an open-source model (e.g., *Llama-3.3 70B*) to eliminate the credit cost; however, due to memory hardware limitations, this was not possible. The data generated are also available at the repository of this paper[7]. For our experiments, we utilized the evaluation part of *MedMCQA*, which consists of 4186 question-answer pairs (not included in the training part). Specifically, we selected the single choice questions (2816 in total).

Table 1
LLM Prompts used in this study. Square brackets signify the placement of sub-prompts, while text variables appear in angle brackets.

Prompt	
Data Extraction (GPT-4o mini)	Please extract a semantic triplet in the form (subject, predicate, object) from the following question answer pair. Use the Answer in your triplet. Question: <question_text> Answer: <correct_answer_text> Output only the triplet itself.
Question answering (Vector-GraphRAG)	[system_prompt] [user_prompt] To answer the question use the following context. Context: <factual_triplets> Question: <question> Options: <possible_answers>
Question answering (Baseline)	[system_prompt] [user_prompt] To answer the question use your internal knowledge. Question: <question_text> Options: <possible_answers_text>
System	You are an informative chatbot. Please answer the user's question to the best of your ability. If you do not know something please state that to the user.
User	This is just an evaluation question of a medical question answering dataset. Please answer this question by selecting the correct answer from the options below.

4.3. Experiments

For the medical QA task, the LLMs were instructed to select the correct answer from a list of possible ones. To extract the answer from the model output, we utilized regular expressions. These answers were then compared with the correct ones to calculate the accuracy score. Table 2 summarizes the results of our experiments concerning the evaluation of several open-source LLMs, while a comparative assessment of our approach against previous ones is given in Table 3. To ensure the reproducibility of our results, we use a specific random seed and we set the model temperature to 0.0.

Table 2
Experiments on MedMCQA; Acc. denotes the accuracy percentage score of each setup.

LLM	Baseline Acc. (%)	VectorGraphRAG Acc. (%)
Llama-3.1 (8B)	55.26	57.03
Gemma-2 (9B)	53.91	57.39
Command-R (7B)	39.67	45.17
Ministral (8B)	48.01	50.46
Falcon-3 (10B)	52.59	57.14

Table 3
Comparative assessment of VectorGraphRAG against other approaches; Acc. denotes the accuracy percentage score of each setup. Model precision is denoted in parentheses.

Approach	Acc. (%)
VectorGraphRAG - Llama-3.1-8B (full)	57.14
VectorGraphRAG - Llama-3.1-8B (4-bit)	57.03
MedGraphRAG - Llama-3-8B [19]	61.6
5-shot - ChatGPT-3.5 [28]	56.37
Bailicai-8B [29]	55.89
RAG - Llama-3 [29]	55.27
CoT - ChatGPT-3.5 [29]	53.79
CoT - Llama-3-8B [29]	52.52
Zero-Shot - ChatGPT-3.5 [28]	48.91
DEEB-RAG - Llama-2-7B-chat [30]	34.6

As shown in Table 2, there is an increase in the accuracy of all models considered (up to +5.5%) when using *VectorGraphRAG* (compared to the case that the models use just their internal knowledge, i.e., Baseline). For the best performance of *VectorGraphRAG*, we set the number of triplets to 300 and we retrieve the *top-10* most similar terms to each question term extracted from the KG.

These parameters are selected based on additional experiments that were performed using *Llama-3.1* (see Figures 3, 4). As shown in Figure 3, the model accuracy peaks at 300 factual triplets and then decreases to the baseline accuracy at 600 triplets. Afterwards, it starts increasing, until a local optimum is reached at 1100 triplets; after this point, the accuracy drops significantly. Thus, we infer that we can maximize the LLM accuracy while minimizing the retrieved KG context (measured in the number of triplets). As mentioned above, for each medical term we retrieve the *top-k* most similar terms. To find the optimal number of *k*, we experiment with several values, while keeping the number of triplets to 300. As shown in Figure 4, the accuracy fluctuates, with 10 being the best value of *k*.

We also compared the proposed approach with similar ones. As shown in Table 3, previous approaches underperform with the sole exception of *MedGraphRAG*. An interesting remark is that our approach achieves near state-of-the-art accuracy, while being much more efficient, due to its performance-oriented design and the use of a single LLM generation for RAG. In contrast, *MedGraphRAG* requires multiple LLM generations to summarize graph communities that contain medical facts to achieve state-of-the-art accuracy. Similarly, *Bailicai* utilizes multiple LLM generations

[6]https://github.com/openai/openai-python
[7]https://github.com/NC0DER/VectorGraphRAG

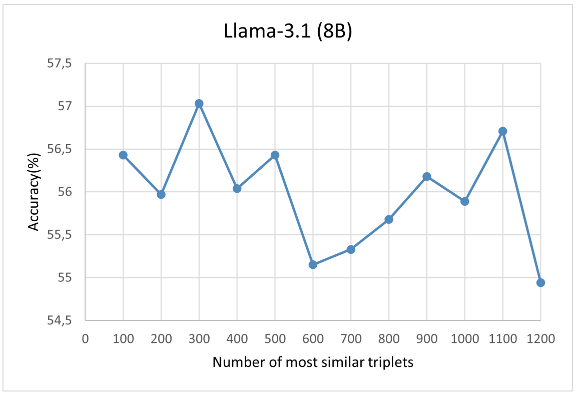


Figure 3: Measuring the effect of accuracy based on the number of factual triplets.

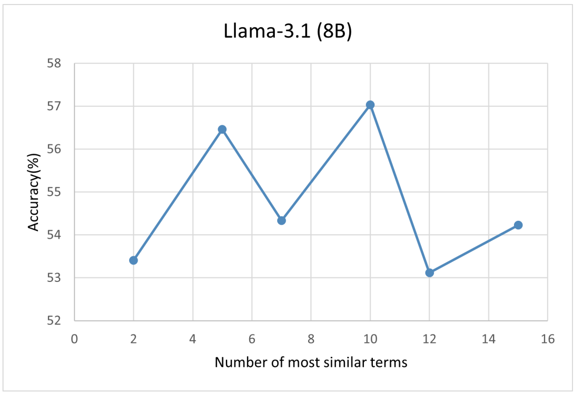


Figure 4: Measuring the effect of accuracy based on the number of most similar terms.

and performs fine-tuning, both of them being computationally expensive [2].

It is also stressed that our approach achieves better performance (even with the quantized LLM), compared to previous ones with non-quantized LLMs (except *MedGraphRAG*). Thus, by using 4-bit quantized LLMs, we reduce the required memory by 75%, without sacrificing accuracy, which is particularly useful given the high cost of GPUs with large VRAM.

From our experimental results, we also infer that providing only the most relevant information to the LLM and not filling its entire context window with unnecessary information leads to accuracy and performance benefits. In this study, we utilized several prompts that are listed in Table 1.

5. Discussion

VectorGraphRAG overcomes the limitations of previous medical QA approaches. Specifically, it: (i) automates the process of building a KG by using an LLM to extract triplets from QA pairs, thus reducing the labor of domain experts; (ii) utilizes a scalable graph database that loads only a small portion of the KG, thus being able to scale on larger KGs that do not fit entirely in memory; (iii) utilizes a small in-memory vector index to accelerate the entity search based on the *top-10* most similar terms; (iv) once the appropriate graph context is retrieved, performs a single LLM generation instead of multiple ones to significantly decrease the total inference time; (v) ranks the triplets and limits them to provide only the most relevant context to the LLM, thus increasing accuracy, and (vi) utilizes LLM 4-bit quantization, which approximately requires a quarter of the amount of VRAM compared to other approaches, without any significant accuracy loss.

Based on the above remarks, we can answer the research questions listed in the Introduction. Overall, we introduced a novel approach that uses an LLM for automatic triplet creation instead of relying on existing medical KGs or specialized ontologies (**RQ1**). Our experimental results showed that using a quantized LLM does not significantly reduce model accuracy (**RQ2**). When considering the results from other approaches with different prompting strategies, we observed that non-GraphRAG approaches underperformed [29, 28] (**RQ3**). Finally, as shown in Figure 3, we observed that increasing the amount of retrieved KG context does not always improve accuracy (**RQ4**).

In any case, our approach has a set of limitations. First, it requires a large state-of-the-art LLM (i.e., *GPT-4o mini*) for creating the triplets. In addition, given the fact that the LLM might hallucinate when producing some of these triplets, a careful validation from domain experts is required. Furthermore, our approach expects questions where there is a single correct answer; for questions where there is no clear answer, we should build an extra processing layer.

As far as future work directions are concerned, we consider the incorporation of domain-specific ontologies in our approach, aiming to further improve its accuracy. Additionally, we plan to support factual triplet extraction from unstructured text, by building an additional processing layer. A third work direction is to run experiments with more evaluation metrics and QA datasets, as well as to employ human evaluation, aiming to get additional insights about the proposed approach and generalize our research findings. A fourth work direction is to evaluate our approach for open QA [35], where the model-generated answer is semantically compared to the one given by a human. A fifth work direction is to integrate state-of-the-art LLM embeddings from the MTEB leaderboard [36] to further improve the accuracy of the proposed approach. A sixth work direction is to measure the exact performance benefits of our approach when compared against similar ones. A final work direction involves the integration of interpretable and explainable AI techniques, aiming to make our approach more transparent [37].

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large Language Models: A Survey, 2024. doi:10.48550/arXiv.2402.06196, arXiv:2402.06196 version: 2.

[2] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Y. Nie, J. R. Wen, A Survey of Large Language Models, 2024. doi:10.48550/arXiv.2303.18223, arXiv:2303.18223 version: 15.

[3] C. Wang, X. Liu, Y. Yue, X. Tang, T. Zhang, C. Jiayang, Y. Yao, W. Gao, X. Hu, Z. Qi, Y. Wang, L. Yang, J. Wang, X. Xie, Z. Zhang, Y. Zhang, Survey on Factuality in Large Language Models: Knowledge, Retrieval and Domain-Specificity, 2023. doi:10.48550/arXiv.2310.07521, arXiv:2310.07521.

[4] A. Hadi, E. Tran, B. Nagarajan, A. Kirpalani, Evaluation of ChatGPT as a diagnostic tool for medical learners and clinicians, PLOS ONE 19 (2024) e0307383. doi:10.1371/journal.pone.0307383.

[5] C. Y. K. Williams, B. Y. Miao, A. E. Kornblith, A. J. Butte, Evaluating the use of large language models to provide clinical recommendations in the Emergency Department, Nature Communications 15 (2024) 8236. doi:10.1038/s41467-024-52415-1.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is All you Need, in: Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[7] H. Han, Y. Wang, H. Shomer, K. Guo, J. Ding, Y. Lei, M. Halappanavar, R. A. Rossi, S. Mukherjee, X. Tang, Q. He, Z. Hua, B. Long, T. Zhao, N. Shah, A. Javari, Y. Xia, J. Tang, Retrieval-Augmented Generation with Graphs (GraphRAG), 2025. doi:10.48550/arXiv.2501.00309, arXiv:2501.00309.

[8] T. W. Ko, J. Y. Jiang, P. J. Cheng, Beyond Independent Passages: Adaptive Passage Combination Retrieval for Retrieval Augmented Open-Domain Question Answering, 2025. doi:10.48550/arXiv.2507.04069, arXiv:2507.04069.

[9] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, J. Larson, From Local to Global: A Graph RAG Approach to Query-Focused Summarization, 2024. doi:10.48550/arXiv.2404.16130, arXiv:2404.16130 version: 1.

[10] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, S. Tang, Graph Retrieval-Augmented Generation: A Survey, 2024. doi:10.48550/arXiv.2408.08921, arXiv:2408.08921.

[11] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semantic Web 8 (2016) 489–508. doi:10.3233/SW-160218.

[12] J. Z. Pan, S. Razniewski, J. C. Kalo, S. Singhania, J. Chen, S. Dietze, H. Jabeen, J. Omeliyanenko, W. Zhang, M. Lissandrini, R. Biswas, G. d. Melo, A. Bonifati, E. Vakaj, M. Dragoni, D. Graux, Large Language Models and Knowledge Graphs: Opportunities and Challenges, 2023. doi:10.48550/arXiv.2308.06374, arXiv:2308.06374.

[13] X. Yan, P. Westphal, J. Seliger, R. Usbeck, Bridging the Gap: Generating a Comprehensive Biomedical Knowledge Graph Question Answering Dataset, in: ECAI 2024 - 27th European Conference on Artificial Intelligence - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024), volume 392 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2024, pp. 1198–1205. doi:10.3233/FAIA240615.

[14] P. Zhang, C. Cao, P. Groth, TIGER: Temporally Improved Graph Entity Linker, in: ECAI 2024 - 27th European Conference on Artificial Intelligence - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024), volume 392 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2024, pp. 3733–3740. doi:10.3233/FAIA240933.

[15] J. Chen, H. Dong, J. Hastings, E. Jiménez-Ruiz, V. López, P. Monnin, C. Pesquita, P. Škoda, V. Tamma, Knowledge Graphs for the Life Sciences: Recent Developments, Challenges and Opportunities, 2023. doi:10.48550/arXiv.2309.17255, arXiv:2309.17255.

[16] S. Y. Liu, Z. Liu, X. Huang, P. Dong, K. T. Cheng, LLM-FP4: 4-Bit Floating-Point Quantized Transformers, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 592–605. doi:10.18653/v1/2023.emnlp-main.39.

[17] J. Monteiro, F. Sá, J. Bernardino, Experimental Evaluation of Graph Databases: JanusGraph, Nebula Graph, Neo4j, and TigerGraph, Applied Sciences 13 (2023) 5770. doi:10.3390/app13095770.

[18] A. Pal, L. K. Umapathi, M. Sankarasubbu, MedMCQA: A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering, in: Conference on Health, Inference, and Learning, CHIL 2022, 7-8 April 2022, Virtual Event, volume 174 of *Proceedings of Machine Learning Research*, PMLR, 2022, pp. 248–260. URL: https://proceedings.mlr.press/v174/pal22a.html.

[19] J. Wu, J. Zhu, Y. Qi, J. Chen, M. Xu, F. Menolascina, V. Grau, Medical Graph RAG: Towards Safe Medical Large Language Model via Graph Retrieval-Augmented Generation, 2024. doi:10.48550/arXiv.2408.04187, arXiv:2408.04187.

[20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, in: Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[21] N. Giarelis, C. Mastrokostas, N. Karacapilidis, A Unified LLM-KG Framework to Assist Fact-Checking in Public Deliberation, in: A. Hautli-Janisz, G. Lapesa, L. Anastasiou, V. Gold, A. D. Liddo, C. Reed (Eds.), Proceedings of the First Workshop on Language-driven Deliberation Technology (DELITE) @ LREC-COLING 2024, ELRA and ICCL, Torino, Italia, 2024, pp. 13–19. URL: https://aclanthology.org/2024.delite-1.2/.

[22] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, et al., The Llama 3 Herd of Models, 2024. doi:10.48550/arXiv.2407.21783, arXiv:2407.21783.

[23] M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, J. Ferret, P. Liu, P. Tafti, A. Friesen, M. Casbon, S. Ramos, R. Kumar, C. L. Lan, S. Jerome, A. Tsit-

sulin, N. Vieillard, P. Stanczyk, S. Girgin, N. Momchev, M. Hoffman, S. Thakoor, J. B. Grill, B. Neyshabur, O. Bachem, et al., Gemma 2: Improving Open Language Models at a Practical Size, 2024. doi:10.48550/arXiv.2408.00118, arXiv:2408.00118.

[24] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M. A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7B, 2023. doi:10.48550/arXiv.2310.06825, arXiv:2310.06825.

[25] J. Dang, A. Ahmadian, K. Marchisio, J. Kreutzer, A. Üstün, S. Hooker, RLHF Can Speak Many Languages: Unlocking Multilingual Preference Optimization for LLMs, 2024. doi:10.48550/arXiv.2407.02552, arXiv:2407.02552.

[26] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Hesslow, J. Launay, Q. Malartic, D. Mazzotta, B. Noune, B. Pannier, G. Penedo, The Falcon Series of Open Language Models, 2023. doi:10.48550/arXiv.2311.16867, arXiv:2311.16867.

[27] X. Jiang, R. Zhang, Y. Xu, R. Qiu, Y. Fang, Z. Wang, J. Tang, H. Ding, X. Chu, J. Zhao, Y. Wang, HyKGE: A Hypothesis Knowledge Graph Enhanced Framework for Accurate and Reliable Medical LLMs Responses, 2024. doi:10.48550/arXiv.2312.15883, arXiv:2312.15883.

[28] Y. Labrak, M. Rouvier, R. Dufour, A Zero-shot and Few-shot Study of Instruction-Finetuned Large Language Models Applied to Clinical and Biomedical Tasks, in: N. Calzolari, M. Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 2049–2066. URL: https://aclanthology.org/2024.lrec-main.185/.

[29] C. Long, Y. Liu, C. Ouyang, Y. Yu, Bailicai: A Domain-Optimized Retrieval-Augmented Generation Framework for Medical Applications, 2024. doi:10.48550/arXiv.2407.21055, arXiv:2407.21055.

[30] Y. Kong, Z. Yang, L. Luo, Z. Ding, L. Wang, W. Liu, Y. Zhang, B. Xu, J. Wang, Y. Sun, Z. Zhao, H. Lin, Document Embeddings Enhance Biomedical Retrieval-Augmented Generation, in: 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2024, pp. 962–967. doi:10.1109/BIBM62325.2024.10822781, iSSN: 2156-1133.

[31] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023. doi:10.48550/arXiv.2307.09288, arXiv:2307.09288.

[32] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.

[33] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with NumPy, Nature 585 (2020) 357–362. doi:10.1038/s41586-020-2649-2.

[34] J. Johnson, M. Douze, H. Jégou, Billion-Scale Similarity Search with GPUs, IEEE Transactions on Big Data 7 (2021) 535–547. doi:10.1109/TBDATA.2019.2921572.

[35] C. Wang, S. Cheng, Q. Guo, Y. Yue, B. Ding, Z. Xu, Y. Wang, X. Hu, Z. Zhang, Y. Zhang, Evaluating open-qa evaluation, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 77013–77042. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/f323d594aa5d2c68154433a131c07959-Paper-Datasets_and_Benchmarks.pdf.

[36] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, MTEB: Massive text embedding benchmark, in: A. Vlachos, I. Augenstein (Eds.), Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 2014–2037. doi:10.18653/v1/2023.eacl-main.148.

[37] N. Giarelis, C. Mastrokostas, I. Siachos, N. Karacapilidis, Integrating knowledge graphs, large language models and explainable ai techniques to improve public health question answering, in: I. Lindgren, M. P. Rodríguez Bolívar, M. Janssen, E. Loukis, F. Mureddu, P. Panagiotopoulos, G. Viale Pereira, E. Tambouris (Eds.), Electronic Government, volume 15944, Springer Nature Switzerland, Cham, 2025, pp. 368–379. doi:10.1007/978-3-032-01589-1_23.