OmniRAG - Knowledge Graph Assisted Retrieval Augmented Generation for Medical Diagnosis

Thomas He
University of Michigan
mperform@umich.edu

Abstract

Knowledge Graphs (KG) is a promising Retrievel-Augmented-Generation (RAG) structure that is well-suited for retrieving medical concept relations. It serves as a potential solution to reduce Large Language Model (LLM) hallucination issues in the medical domain. However, existing KG-elicited reasoning frameworks suffer from generalizability issues as well as grounding issues. This paper introduces OmniKG, a novel Hybrid Top-Down Knowledge Graph construction approach for medical domain, aiming to tackles these shortcomings. In addition, OmniRAG, a medical diagnosis framework, is built around OmniKG, referencing existing RAG retrieval methodologies, to maximize the KG retrieval quality. The framework is evaluated on both an In-Domain dataset DDXPlus, and an Out-of-Domain dataset Symptom2Diagnosis to display its diagnosis accuracy and KG retrieval quality. OmniRAG is compared against several baseline approaches and exhibits its superiority in various metrics. To motivate future research, our codebase will be available at `https://github.com/Public-Releases/OmniRAG.git`

1 Introduction

Access high quality medical resources is still limited around the globe. While the rapid development of Large Language Models (LLMs) offer a potential solution for scalable agentic diagnostic systems, their deployment is limited by the reliability issues. Notably, general-purpose LLMs frequently suffer from hallucinations, lack of medical-domain reasoning, and struggle to ground their diagnosis to verifiable medical protocols.

To address such limitations, recent research has focused on Retrieval-Augmented-Generation (RAG), attempting to ground LLM outputs by retrieving relevant medical context and concepts. A sub-category of retrieval called Medical Knowledge Graphs (KGs) has exhibited high potential. KG retrieval contains structured relationships (e.g. Symptom A → Disease B). However, existing methods typically rely on Bottom-Up construction combined with mining statistical correlations from specific datasets. While effective on In-Domain data, they often fail to generalize to real-world, unstructured patient natural language descriptions, leading to inaccurate and ineffective retrieval contents.

In this work, we propose OmniKG, a novel Hybrid Top-Down Knowledge Graph construction approach leveraging powerful State-of-the-Art (SOTA) LLMs as a “Medical Expert” to generate a rigorous clinical ontology. In addition, we supplement this ontology with empirical data mining, to create a hybrid structure fusing both clinical definitions and layperson dialogues. This structure allows our framework to perform semantic retrieval rather than keyword matching. This KG design is then integrated into a RAG and KG retrieval framework, extended from a previous work, MedRAG [17]. We refer to this complete framework as OmniRAG.

We then evaluate OmniRAG on the DDXPlus benchmark and Symtom2Diagnosis dataset. Our results demonstrate that OmniRAG achieves state-of-the-art performance, with 79.0% Exact Match Accuracy on DDXPlus. Most notably, OmniRAG demonstrates superior robustness on unseen out-of-domain

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Example Patient Entry

Do you have pain somewhere, related to your reason for consulting?: **TRUE**

Figure 1: Example of a symptom question and corresponding label.

Node Attribute (has_symptomatology)
"Do you have a burning sensation that starts in your stomach...?"

Patient Input
"Do you have a burning sensation that starts in your stomach...?"

Figure 2: Example alignment between a knowledge-graph symptom node and a patient-provided symptom description.

data with an average KG Retrieval score of 6.57 out of 10, compared to the baseline's 4.69. This confirms the superiority of our framework and KG design, enhancing both the safety and reliability of medical diagnostic agents.

2 Related Work

With the rise of LLMs in the past few years, significant efforts have been made to design and train domain-specific models. It is undeniable that there is significant application potential with LLMs across the medical field, namely EHR data analysis, patient monitoring, patient diagnosis, etc. [7][16][1]. However, they generally suffer from hallucination and struggle from medical-domain reasoning. Thus, many works have been done to address this with Retrieval-Augmented-Generation (RAG) [10]. However, these approaches struggle to retrieve accurate medical context when encountering patients with similar medical profiles. To this end, recent works have focused on leveraging Knowledge Graph (KG) structures to address such issues [15] [8]. Early medical KGs were predominantly manually curated by experts [9][4]. While accurate, they are labor intensive and not scalable. Thus, additional works have adopted data-driven, bottom-up approaches. KG4Diagnosis [18] constructs a graph by mining the symptom-to-disease occurrences directly from medical datasets and optimizing for collaborative filtering. MedRAG [17] builds a four-tier hierarchical graph based on the statistical distribution of features along with LLM prompting within medical datasets. While these designs achieve high performance on In-Domain datasets, they suffer from Distribution Overfitting, where they essentially created a vocabulary mapping of the dataset. Our experiments show they struggle to generalize to unstructured natural language patient descriptions since they lack the semantic breadth beyond the dataset schema. Thus, another work, Dr. KNOWS [3] leverages LLMs to conduct a Top-Down Knowledge Graph construction. Similarly, medIKAL [6] employs LLMs to extract medical entities and their relationships from textbooks or clinical notes to build such KGs. While these Top-Down approaches excel at capturing medical knowledge relations, they often suffer from grounding issues when applied to real world datasets. This could be due to terminology, data distribution, or vernacular differences.

3 Motivation

While recent RAG frameworks such as MedRAG demonstrate great promise, our reproduction and analysis identified a few limitations that would impact their generalizability to real-world scenarios. Although claiming to be open-source, MedRAG's official codebase is a simplified implementation designed for a private dataset while also omitting the core KG construction algorithm. The codebase follows a simplified KG content retrieval, relying instead on direct symptom-to-disease vector matching. The MedRAG KG, we refer to as Baseline KG, contains majority of verbatim survey questions (e.g. *"Do you have pain...?"*) rather than atomic medical concepts. This results in the inclusion of non-informative nodes, such as the one shown in Figure 1. These non-informative nodes introduce noise without adding diagnostic value. We also hypothesize that baseline KG's high

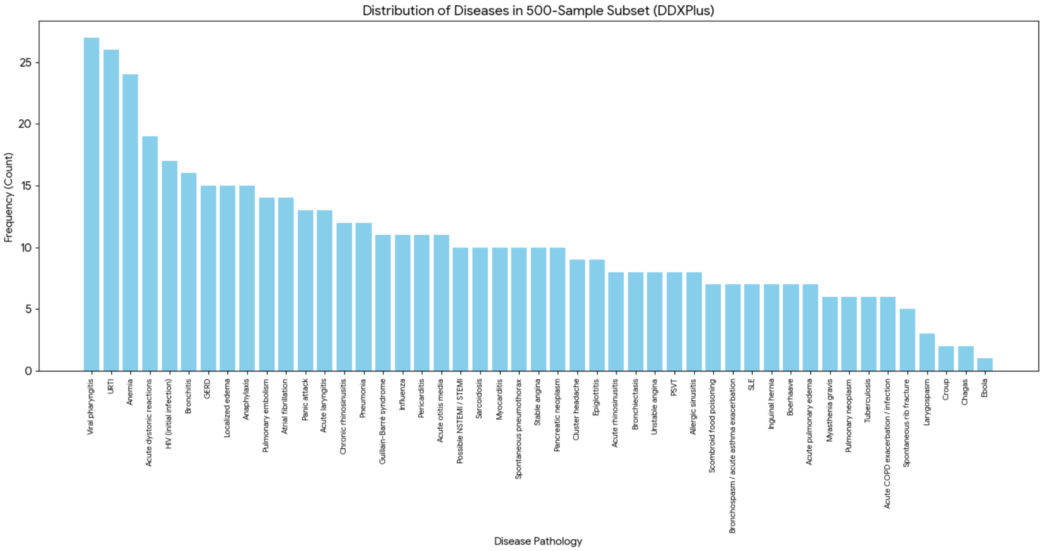


Figure 3: Distribution of diseases in our 500-sample subset of the DDXPlus dataset.

in-domain accuracy stems from keywords pattern matching rather than semantic reasoning. Figure 2 is an example of a correct KG retrieval by baseline KG. We observe that due to the long string node, in-domain matching would function well as the test set contains almost exact matches. However, one symptom can have infinite representations or rephrases, which would no longer match with these long description nodes.

4 Data

4.1 DDXPlus

There are two datasets used in our system. First is the DDXPlus dataset [12], which is a fully synthetic dataset but grounded in real-world medical knowledge. It was generated with a probabilistic framework rather than real clinical data. The medical knowledge base was compiled from over 20,000 medical papers. From the papers, they extracted diseases, prevalence, symptom likelihood ratios across different demographics. Then, they use this knowledge base and public census data through the probabilistic model to generate around 1.3 million synthetic patients. For each patient, they sample a pathology and sample the antecedents and symptoms based on the pathology's probability profile. Then, they generated a ground-truth differential diagnosis, which is a ranked list of likely diseases along with the actual true pathology. This dataset contains a total of 49 distinct diseases, and a combined total of 110 symptoms and 113 antecedents. The training set is around 1 million samples and test set is around 130,000 patients. Figure 4 shows an examplar patient profile. We sample the first 500 examples from the DDXPlus dataset. Since the data is generated with a probabilistic model, we believe that it takes into consideration of the real world probability of each disease occurring. Figure 3 displays the 500-sample distribution of diseases. OmniKG is generated from the first 1000 examples of the DDXPlus training set. OmniRAG Patient manifest retrieval uses the same 1000 subset.

4.2 Symptom2Diagnosis

Additionally, to evaluate our KG's generalizability and effectiveness on unseen and other formatted datasets, we chose the Symptom2Diagnosis [5] dataset available on Hugging Face. This dataset provides 1065 symptom descriptions labeled with 22 corresponding diagnoses. The symptoms are natural English describing the patient's discomforts. An example of a data point is shown in (see Figure 5). In addition, most of the diseases in this dataset are not present in the constructed knowledge graph. Thus, to ensure the effectiveness of the KG, we further filter the dataset to four concepts that exist in our KG: Gastroesophageal Reflux Disease (GERD), Pneumonia, Bronchial Asthma, and

Age	55
Sex	F
Pathology	Anemia
Processed Diagnosis	Anemia
Symptoms	"Do you have pain somewhere, related to your reason for consulting?";... …
Antecedents	"Have you traveled out of the country in the last 4 weeks?: South East Asia";... …

Figure 4: Example synthetic patient record from the DDXPlus dataset.

input_text	I've been having headaches and migraines, and I can't sleep. My whole body shakes and twitches. Sometimes I feel lightheaded.
output_text	drug reaction

Figure 5: Example Symptom2Diagnosis record from the HuggingFace dataset.

Common Cold (URTI – Upper Respiratory Tract Infection). The final generalizability test set is condensed to 195 examples.

5 Approach

5.1 OmniRAG Framework Implementation

We first build OmniRAG framework, Figure 6, from the ground up based on the publicly available DDXPlus dataset referencing the MedRAG design philosophy. We include comprehensive testing using an AI-Driven test suite. First, we implement a preprocessing algorithm that extracts the patient information from DDXPlus dataset into separate patient cases, where each patient case is compiled into a JSON file. For the RAG patient case database search, we use FAISS [2] by Facebook Research, which allows us to efficiently search for similar patient cases from the EHR database. We implement multiple KG context retrieval methods including hierarchical context retrieval and ranking, and text similarity matching. For reproduction of MedRAG, AI-tools such as Cursor was used for portions of the implementation.

5.2 Hybrid Top-Down Knowledge Graph Construction

We leverage a LLM as the medical expert, encoding world knowledge into the graph structure, rather than only relying solely on training data distribution. We also adopt data mining techniques to fuse dataset-specific semantics and patient attributes into the knowledge graph.

Formally, we define our Knowledge Graph as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of medical entities (diseases, symptoms, attributes) and $\mathcal{E}$ represents the semantic relationships between them. Let $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ be the set of target diagnoses derived from the dataset metadata.

5.2.1 Phase 1 – Hierarchical Taxonomy Generation

We first gather the list of dataset diseases $\mathcal{D}$ and organize it into a clinical hierarchy for efficient retrieval. This hierarchy serves as the backbone for the knowledge graph. It will also support the hierarchical voting mechanism in MedRAG.

An LLM, denoted by $\mathcal{M}_{expert}$, constructs a three-tier structure via zero-shot prompting. For each disease $d_i \in \mathcal{D}$, the model maps it to a tuple (l_1, l_2, l_3) where:

- **Level 1:** Organ System (e.g., Cardiovascular System)
- **Level 2:** Disease Subcategory (e.g., Ischemic Heart Disease)
- **Level 3:** Specific Pathology (e.g., Myocardial Infarction)

This process generates a set of edges $\mathcal{E}_{hierarchy}$ representing *is_a* relationships as well as identifying *similar_diseases* relationships between clinically related pathologies.

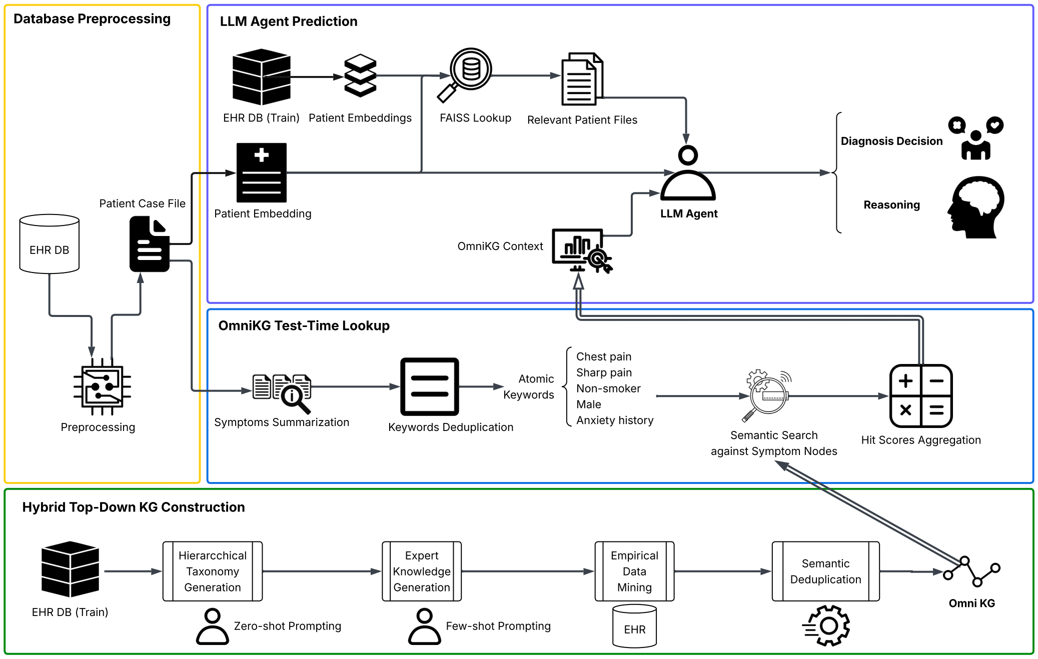


Figure 6: OmniRAG Framework

5.2.2 Phase 2 – Clinical Feature Extraction and Enrichment

Given the backbone graph structure hierarchy, we populate the edge nodes with clinical knowledge through prompt engineering as well as data mining from EHR datasets.

Expert Knowledge Generation To equip the graph with more medical knowledge and relations, we use few-shot prompting to enforce a strict JSON schema when generating symptoms. We prompt the medical expert to output several key categories/attributes for each disease. These include critical symptoms, risk factors, and diagnostic features. To address the generalizability of our KG, we utilize *Layperson Semantic Extraction* to add descriptions alongside medical terminology. We believe this will help match patient profile's symptoms, which are typically layperson descriptions, to the target disease in the graph. We formalize the generation of expert features F_{exp} for a disease d_i as sampling from the model's probability distribution conditioned on the given disease and few-shot examples $\mathcal{I}$:

$$F_{exp}(d_i) \sim P_{\mathcal{M}}(f \mid d_i, \mathcal{I}_{few-shot}) \tag{1}$$

where f denotes the textual output from the $\mathcal{M}_{expert}$ given disease d_i and few-shot examples $\mathcal{I}$.

Empirical Data Mining To capture the "noisy" reality of how patient actually describe symptoms in the dataset, we conduct data mining by scraping the raw features in the EHR training set $\mathcal{T}$. The set of raw observations/features associated with disease d_i in the dataset can be represented by $O_{raw}(d_i)$. These raw features are often verbose questions (e.g. "Do you feel pain somewhere?: lower chest"), which are difficult to generalize to other datasets and also stored as disease attributes. Thus, we deploy a lightweight Language Model serving as a "Clinical Scribe" denoted by $\mathcal{S}_\phi$ to summarize them into atomic keywords:

$$F_{layperson}(d_i) = \{\mathcal{S}_\phi(o) \mid o \in O_{raw}(d_i)\} \tag{2}$$

For example, $\mathcal{S}_\phi$ maps "Do you feel pain somewhere?: lower chest" $\rightarrow$ "lower chest pain".

Semantic Deduplication Since the atomic keywords $F_{layperson}(d_i)$ often result in redundant symptom nodes, we conduct semantic deduplication, which first clusters similar keywords together and naively choose a representative keyword. Formally, we partition $F_{layperson}$ into disjoint clusters $\mathcal{C} = \{C_1, ..., C_K\}$ using Agglomerative Clustering, such that for any two features $u, v \in c_k$:

$$1 - \cos(\mathbf{E}(u), \mathbf{E}(v)) \leq \tau \tag{3}$$

where τ is a hyperparameter controlling the distance threshold. For each cluster, we select the shortest atomic keyword as the representative, favoring concise generalization. The selected keywords are then added along with the generated symptoms from the medical expert to the disease to form a more comprehensive symptoms knowledge base.

5.2.3 Phase 3 – Graph Formalization

Finally, we convert the processed taxonomy and deduplicated features into a set of triplets (h, r, t) to form the edge set $\mathcal{E}$. The relation types r include:

- `is_a`: Hierarchical parent-child links.
- `similar_to`: Differential diagnosis links.
- `has_symptom`: Connects diseases to deduplicated symptom nodes.
- `risk_factor` / `distinguished_by`: Connects diseases to antecedents and diagnostic nuances.

The final knowledge graph $\mathcal{G}$ thus encodes a dual-layer representation: rigorous medical ontology from the LLM and empirical patient terminology from the data.

5.3 Knowledge Graph Test-time Integration

Given the new Knowledge Graph construction method, we can no longer adopt the MedRAG retrieval process as the content are different. Thus, we employ a three-phase KG retrieval process. First, we conduct patient symptoms summarization and Deduplication. Second, we perform a semantic search against the KG symptoms. Last, we aggregate the symptoms hit scores and find the disease with the highest relevancy to the symptoms.

5.3.1 Phase 1 – Patient symptoms summarization and Deduplication

Since OmniKG contains symptoms that are atomic keywords and deduplicated, to maximize the semantic similarity and medical similarity, we employ the same symptoms cleansing method. We denote the raw input patient symptoms as a set $S_{raw} = \{s_1, s_2, ..., s_m\}$. We leverage the same "Clinical Scribe" lightweight model, $\mathcal{M}_{scribe}$, to extract a set of atomic keywords, S_{atomic}, from raw patient symptoms during test-time via:

$$S_{atomic} = \{\mathcal{M}_{scribe}(s) \mid s \in S_{raw}\} \tag{4}$$

Then, we conduct Symptoms Deduplication by mapping the atomic keywords into vector space using embedding function $\mathbf{E}(\cdot)$. We then partition the vectorized S_{atomic} into clusters $\mathbf{C}$ using Agglomerative Clustering. We partition such that for any two keywords k_i, k_j in a cluster $\mathbf{C}$:

$$1 - cos(\mathbf{E}(k_i), \mathbf{E}(\mathbf{k_j})) \leq \tau \tag{5}$$

where τ is the distance threshold. The final clean query consists of the shortest string length atomic keyword for each cluster.

5.3.2 Phase 2 – Semantic search against KG symptom nodes

When the KG Retrieval framework is initialized, it will pre-compute all symptoms embeddings using SentenceTransformer embedding model. Thus, during test-time, the framework conduct a semantic search between the atomic keywords embeddings and symptoms embeddings, which return a list of matching (symptoms ID, hit score) relationships.

Formally, let V_{KG} represent the set of all symptom nodes in the Knowledge Graph. For each patient case's clean symptom query $q \in S_{clean}$, we compute its semantic similarity against all graph nodes $n \in V_{KG}$:

$$sim(q, n) = cos(\mathbf{E}(q), \mathbf{E}(n)) \tag{6}$$

We filter the matching nodes n^* subject to a relevance threshold δ by:

$$\text{match}(q) = \begin{cases} (n^*, \text{sim}(q, n^*)) & \text{if } \text{sim}(q, n^*) > \delta \\ \emptyset & \text{otherwise} \end{cases} \tag{7}$$

where $n^* = argmax_{n \in V_{KG}} sim(q, n)$.

5.3.3 Phase 3 – Hit Scores Aggregation

Finally, to rank the potential diagnoses for a patient, we aggregate the similarity scores of all matching symptoms connected to a disease $d \in \mathcal{D}$ that has the *has_symptom* relation. The cumulative relevance score $R(d)$ can be written as:

$$R(d) = \sum_{q \in S_{clean}} \mathbb{I}(d \in \text{neighbors}(\text{match}(q))) \cdot \text{sim}(q, \text{match}(q)) \tag{8}$$

where $\mathbb{I}$ is the indicator function where it equates to 1 if disease is linked to the symptom node q, 0 otherwise. Lastly, we rank the top-N ranked diagnoses following:

$$D_{result} = \underset{d \in \mathcal{D}}{\text{top-}N}\, R(d) \tag{9}$$

6 Experiments

6.0.1 Accuracy

We evaluate OmniRAG on exact match diagnosis and flexible accuracy. The exact match measures strict string equality between the generated diagnosis and ground truth. The flexible accuracy incorporates a robust matching logic that accounts for substring overlaps and a predefined dictionary of disease aliases.

6.0.2 LLM-as-a-Judge

We also adopt an "LLM-as-a-Judge" framework to assess the quality of the generated reports and KG retrieved context. We employ an efficient and knowledge-rich model – Gemini 2.0 Flash as the Medical Judge. We score based on two rubrics: Retrieval Content Score and Generated Diagnosis Score. Retrieval Content Score rates the quality of the KG retrieved content. Generated Diagnosis Score rates the quality of the diagnosis content generated by the Backbone LLM. Both rubrics contain five key criteria: medical coherence, logical consistency, completeness, clinical relevance, and accuracy against ground truth. The two scores are graded on a scale of 1-10.

6.1 Experiment Setup

6.1.1 Baselines

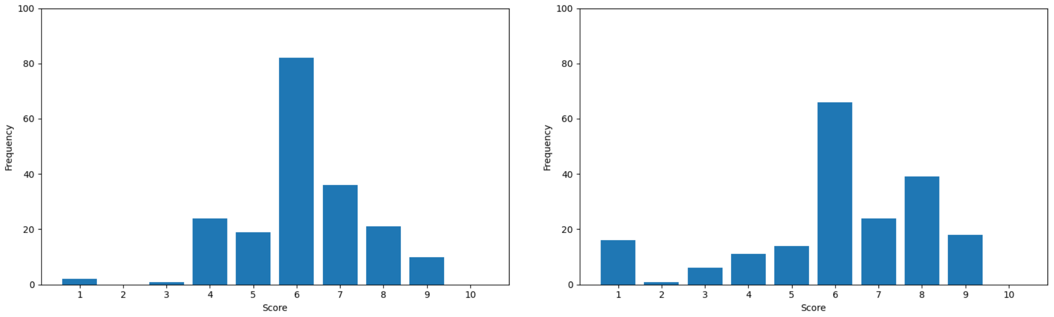
We compare OmniRAG to two baseline methods. First, we compare to a vanilla LLama-3.1-8B without any aid from RAG or KG retrieval. Second, we compare our results to the MedRAG configuration combined with their Baseline KG.

6.1.2 Implementation Details

For efficiency and cost effectiveness, we used Llama 3.1-8B [14] as the Backbone LLM. For KG generation, the taxonomy and clinical features are generated using Gemini-2.5-Flash. For extracting atomic keywords from raw symptoms features, we employ Qwen-3-VL-2B-Instruct [13] as it is lightweight and instruction-following capable. For the embedding generation, we opt for the all-MiniLM-L6-v2 [11] for both graph nodes and patient symptoms embeddings.

6.2 Results and Analysis

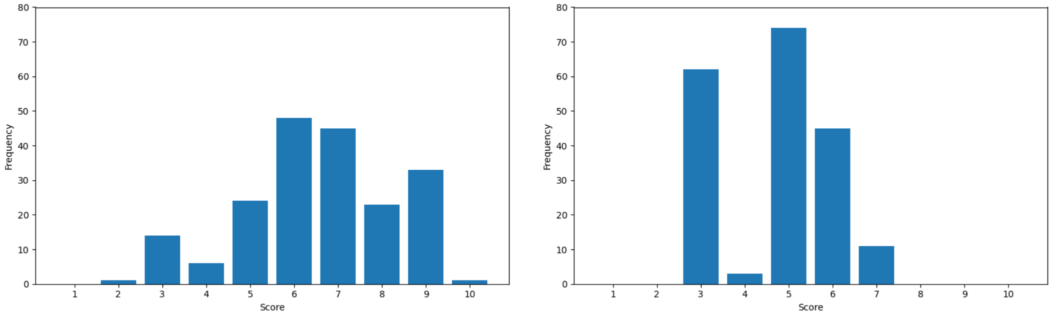
To validate OmniKG construction and retrieval, we structured our analysis into four parts: In-Domain Performance (DDXPlus), Out-of-Domain Generalizability Analysis (Symptom2Diagnosis), Ablation Study, and Qualitative Error Analysis.



(a) OmniRAG (Ours)

(b) MedRAG

Figure 7: Comparison of Generated Diagnosis Score distributions between OmniRAG and Baseline MedRAG.



(a) OmniKG (Ours)

(b) Baseline KG

Figure 8: Comparison of Retrieval Content Score distributions between our OmniKG and the Baseline KG.

6.2.1 In-Domain Performance (DDXPlus)

Table 1: **Diagnostic Performance on DDXPlus Test Set.**

Method	**KG Source**	**Retrieval**	**Top-1 Acc (%)**	**Flexible Acc (%)**	**Δ Flex**
Baseline (Llama-3.1-8B)	None	None	15.30	34.42	+19.12
Baseline KG	Baseline KG	None	47.80	50.80	+3.0
Full MedRAG	Baseline KG	Patient EHR RAG	39.96	52.33	+12.37
Full OmniRAG (Ours)	OmniKG	Patient EHR RAG	38.76	51.34	+12.58
OmniKG (Ours)	**OmniKG**	**None**	**79.00**	**85.00**	**+6.0**

We first compared our framework against baselines on the DDXPlus test set ($N = 500$). As shown in Table 1, OmniKG (OmniRAG with patient EHR retrieval disabled) achieves a Top-1 Exact Match accuracy of **79.00%**, significantly outperforming the baseline and any OmniRAG configurations. Crucially, we observed a distinct gap between Exact Match and Flexible Accuracy across all methods. This indicates that a significant portion of the errors were actually valid diagnosis but just synonym diagnosis names.

6.2.2 Out-of-Domain Generalization (Symptom2Diagnosis)

As mentioned previously, the primary limitation of the MedRAG approach is the generalizability limitation and dataset-overfitting. To validate this, we evaluated OmniRAG versus MedRAG framework on the Symptom2Diagnosis dataset. Since Patient EHR Retrieval negatively impacted overall score as shown in Table 1, both systems have this module disabled. As shown in Figure 2, OmniRAG achieved an average Retrieval Content Score of 6.57, a 2.57 score improvement over Baseline KG. From Figure 8, OmniKG exhibits high retrieval quality as compared to the Baseline KG. From Figure

Table 2: **KG Quality Comparison on Symptom2Diagnosis Dataset**

Metric	Baseline KG	OmniKG (Ours)	Improvement
Retrieval Content Score	4.69	**6.57**	+2.57
Generated Diagnosis Score	6.09	**6.14**	+0.05

7, we can see that even though both systems have similar distribution, Baseline KG has numerous low-score diagnoses. The low-score diagnoses indicates that the reliability of Baseline KG is low.

Table 3: **Impact of Patient EHR Retrieval Module Over 100 Samples**

Method	KG Source	Retrieval	Top-1 Acc (%)	Flexible Acc (%)	Δ Flex
Baseline (Llama-3.1-8B)	None	None	17.05	31.82	+14.77
Patient EHR Retrieval Only	None	Patient EHR RAG	17.98	31.46	+13.48
Patient EHR Retrieval + OmniKG	Top-Down	Patient EHR RAG	36.08	48.45	+12.37
OmniKG	Top-Down	None	79.00	85.00	+6.00

6.2.3 Ablation Study: Impacts of Patient EHR Retrieval

Since we observe that the system performance is being "dragged down" by the Patient EHR Retrieval, we investigate this issue by isolating it and comparing to various configurations. Specifically, we test the sole performance of this module by disabling the OmniKG module and comparing it to other configurations. We sampled 100 examples from DDXPlus due to computing limits. From Figure 3, we can observe that the Patient EHR module is not helpful at boosting diagnosis accuracy. From textual analysis, we observe that the Backbone LLM often gets overwhelmed or confused by the retrieved relevant patient cases, thus diminishing its accuracy as a system.

Table 4: **Common Misclassifications**

Ground Truth	Predicted Diagnosis	Count
Chronic rhinosinusitis	Acute rhinosinusitis	11
Stable angina	Unstable angina	10
Bronchitis	Acute bronchitis	8
Panic attack	Acute pulmonary edema	7
Panic attack	PSVT	5
Bronchospasm / acute asthma exacerbation	Acute asthma exacerbation	4
Boerhaave	Unstable angina	4
SLE	Anaphylaxis	3

6.2.4 Error Analysis

We also conduct misclassification analysis on the DDXPlus dataset results performed by OmniKG. From Figure 4, we can notice that the common misclassifications are valid clinical differentials or semantic overlaps rather than pure hallucinations. This indicates that our KG retrieval is retrieving valid and medically relevant context, even for the misdiagnosed cases.

In Table 5, we compare the per-disease accuracy (top-10 frequency) between the Baseline KG and OmniKG. We observe strong gains across all disease categories. We want to note that Localized edema and Panic Attack failed in OmniKG due to engineering issues during our KG construction. Thus, these nodes were not included in the KG.

7 Conclusion

In conclusion, we introduced OmniRAG, a medical diagnosis framework leveraging a novel Hybrid Top-Down Knowledge Graph, OmniKG, built into a three-phase KG retrieval module, while also supporting Patient EHR database retrieval. We addressed the fundamental vocabulary gap that

Table 5: **Per-Disease Accuracy (Top-10 by Frequency)**

Disease	**Count**	**Baseline KG**	**OmniKG (Ours)**	**Diff**
Viral pharyngitis	27	77.8%	**88.9%**	+11.1%
URTI	26	42.3%	**80.8%**	+38.5%
Anemia	24	87.5%	**100.0%**	+12.5%
Acute dystonic reactions	19	5.3%	**100.0%**	+94.7%
HIV (initial infection)	19	0.0%	**100.0%**	+100.0%
Anaphylaxis	16	62.5%	**100.0%**	+37.5%
Bronchitis	16	18.8%	**43.8%**	+25.0%
GERD	15	86.7%	**100.0%**	+13.3%
Localized edema	15	**26.7%**	0.0%	−26.7%
Panic attack	14	**50.0%**	0.0%	−50.0%

limits traditional bottom-up statistical graphs while minimizing the grounding issues of top-down KG structures. Our experiments demonstrates OmniRAG's significant improvement over baseline approaches on both In-Domain and Out-of-Domain datasets. On DDXPlus (In-Domain), OmniRAG (EHR retrieval disabled) achieved a 79% Top-1 accuracy. On Symptom2Diagnosis (Out-of-Domain), OmniRAG (EHR Retrieval disabled) achieved a 6.57 out-of 10 Retrieval Content Score, a 2.57 improvement from MedRAG.

While OmniRAG exhibits strong knowledge retrieval qualities, there are several avenues for future research. Currently, while our framework supports medical fine-tuned models such as OpenBioLLM, our experiments were conducted using general-purpose models. We observed that while OpenBioLLM is more medically knowledgeable, it struggles with instruction-following and formatting issues. We believe that future research could be conducted to benchmark medical fine-tuned models for OmniKG construction and also serve as the Backbone LLM. Secondly, during KG construction, we noted engineering failures when generating disease node subgraphs (e.g. Panic attack). We believe future work could be conducted to build more robust error handling or utilizing medically knowledgeable models. In addition, while the results seem promising, our setup only utilized a subset of 500 samples. Further large-scale experiments may be conducted to verify our results. Lastly, while we received high Retrieval Content Score on Out-of-Domain dataset, the Top-1 accuracy is relatively low. We believe this is due to the limitations of our Backbone LLM. Future research may be conducted with more knowledgeable models to verify the significance of our KG retrieved content.

References

[1] Malaikannan Sankarasubbu Ankit Pal. Openbiollms: Advancing open-source large language models for healthcare and life sciences. `https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B`, 2024.

[2] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

[3] Yanjun Gao, Ruizhe Li, Emma Croxford, John Caskey, Brian W Patterson, Matthew Churpek, Timothy Miller, Dmitriy Dligach, and Majid Afshar. Leveraging medical knowledge graphs into large language models for diagnosis prediction: Design and application study. *JMIR AI*, 4:e58670, 2025.

[4] Yuanting Gao, Ruoqi Li, Emma Croxford, Jason Caskey, Bruce W. Patterson, Matthew Churpek, Timothy Miller, Dmitriy Dligach, and Majid Afshar. Leveraging medical knowledge graphs into large language models for diagnosis prediction: Design and application study. *JMIR AI*, 4:e58670, February 2025.

[5] Gretel.ai. Symptom to diagnosis dataset. `https://huggingface.co/datasets/gretelai/symptom_to_diagnosis`, 2023. Accessed: 2025-12-08.

[6] Mingyi Jia, Junwen Duan, Yan Song, and Jianxin Wang. medIKAL: Integrating knowledge graphs as assistants of LLMs for enhanced clinical diagnosis on EMRs. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9278–9298, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[7] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day, 2023.

[8] Linfeng Li, Peng Wang, Jun Yan, Yao Wang, Simin Li, Jinpeng Jiang, Zhe Sun, Buzhou Tang, Tsung-Hui Chang, Shenghui Wang, and Yuting Liu. Real-world data medical knowledge graph: construction and applications. *Artificial Intelligence in Medicine*, 103:101817, 2020.

[9] Dun Liu, Qin Pang, Guangai Liu, Hongyu Mou, Jipeng Fan, Yiming Miao, Pin-Han Ho, and Limei Peng. Snomed ct-powered knowledge graphs for structured clinical data and diagnostic reasoning, 2025.

[10] Yiqun Miao, Yuhan Zhao, Yuan Luo, Huiying Wang, and Ying Wu. Improving large language model applications in the medical and nursing domains with retrieval-augmented generation: Scoping review. *J Med Internet Res*, 27:e80557, Oct 2025.

[11] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[12] Arsene Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. DDXPlus Dataset. 6 2022.

[13] Alibaba Team. Qwen3-vl technical report, 2025.

[14] Meta Team. The llama 3 herd of models, 2024.

[15] Song Wang, Mingquan Lin, Tirthankar Ghosal, Ying Ding, and Yifan Peng. Knowledge graph applications in medical imaging analysis: A scoping review. *Health Data Science*, 2022:9841548, 2022.

[16] Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, Christopher A Harle, Gloria Lipori, Duane A Mitchell, William R Hogan, Elizabeth A Shenkman, Jiang Bian, and Yonghui Wu. Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records, 2022.

[17] Xuejiao Zhao, Siyan Liu, Su-Yin Yang, and Chunyan Miao. Medrag: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot, 2025.

[18] Kaiwen Zuo, Yirui Jiang, Fan Mo, and Pietro Lio. Kg4diagnosis: A hierarchical multi-agent llm framework with knowledge graph enhancement for medical diagnosis. In Junde Wu, Jiayuan Zhu, Min Xu, and Yueming Jin, editors, *Proceedings of The First AAAI Bridge Program on AI for Medicine and Healthcare*, volume 281 of *Proceedings of Machine Learning Research*, pages 195–204. PMLR, 25 Feb 2025.

8 Appendix

Table 6: **Accuracy Comparison on Symptom-to-Diagnosis (195 Cases)**

Method	**Top-1 Acc (%)**	**Flexible Acc (%)**	**Success Rate (%)**	**N**
OmniRAG (OmniKG)	24.10	31.79	100.0	195
MedRAG (PaperKG-195)	32.58	36.52	91.3	178

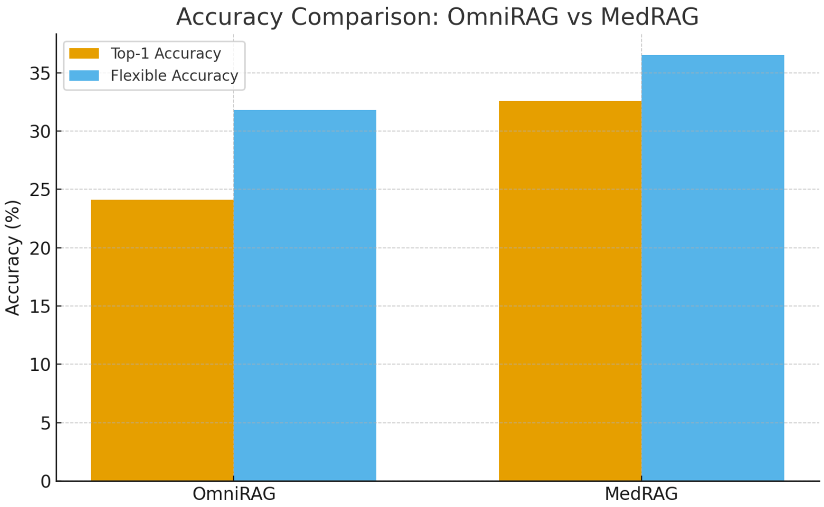


Figure 9: Accuracy comparsion between OmniRAG and MedRAG.

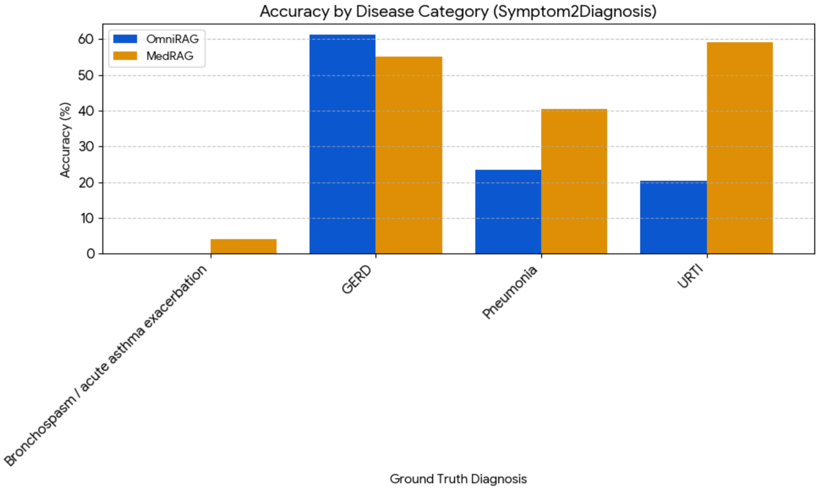


Figure 10: Accuracy by disease category.

Figure 9 6 are the Top-1 and Flexible Accuracy results on Symptom2Diagnosis dataset. Both configuration have Patient EHR Retrieval module disabled for fair comparison. Due to disease aliases, we believe the Flexible Accuracy is most reflective of both configurations. We can observe that both configurations still struggle on Symptom2Diagnosis dataset.

From Figure 13, we notice that OmniRAG system successfully retrieves the correct diagnostic term (e.g., finding "Pneumonia" in the graph) 39.5% of the time, compared to 32.8% for MedRAG, indicating our KG content retrieval method is successfully matching the out-of-domain symptoms natural language to KG nodes.

We also plotted the confusion matrix between the predicted and true diagnosis for the two system configurations MedRAG and OmniRAG 11 12. We can observe that both configurations have misclassifications that have similar symptoms (e.g. Pneumonia vs. Acute Bronchitis). Interestingly, the MedRAG configuration resulted in 17 refused diagnoses while OmniRAG did not refuse to diagnose any patient cases.

In addition, we analyzed the accuracy by disease category between the two configurations. We see that from Figure 10 OmniRAG outperformed MedRAG in only GERD, showing there are still areas of improvement for our framework.

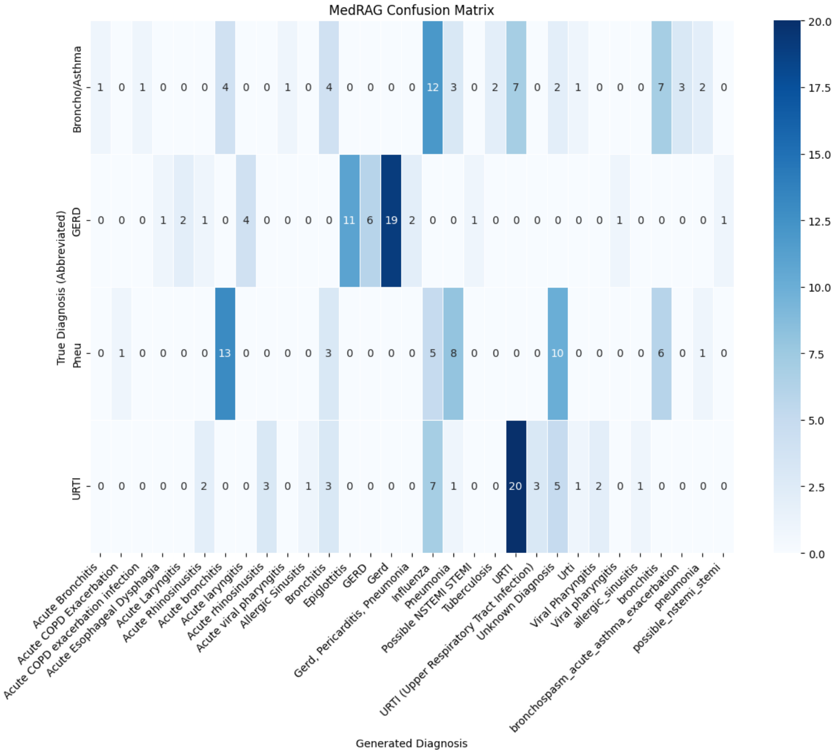


Figure 11: MedRAG confusion matrix.

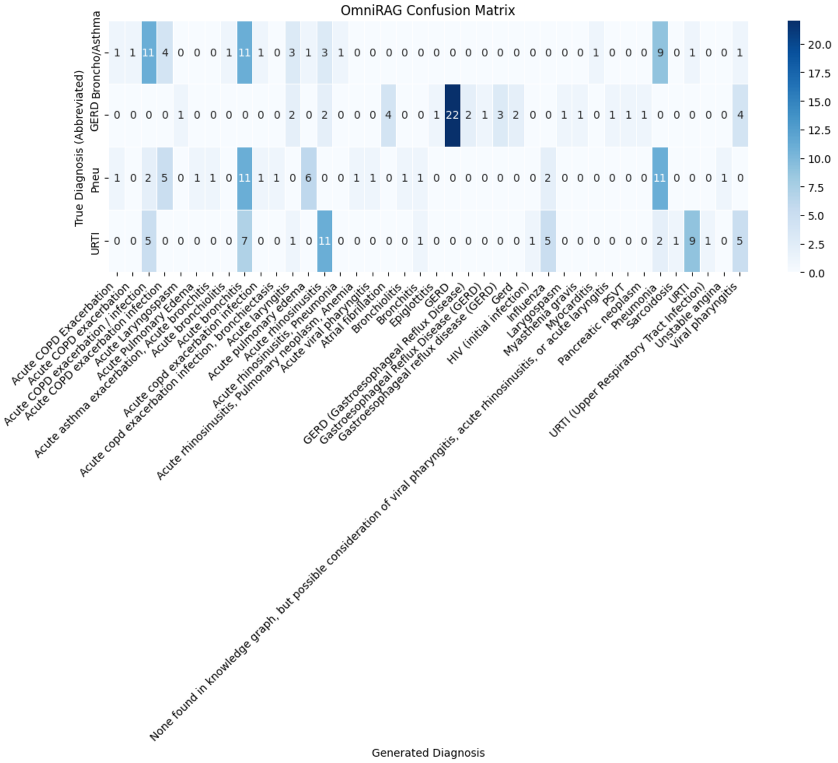


Figure 12: OmniRAG confusion matrix.

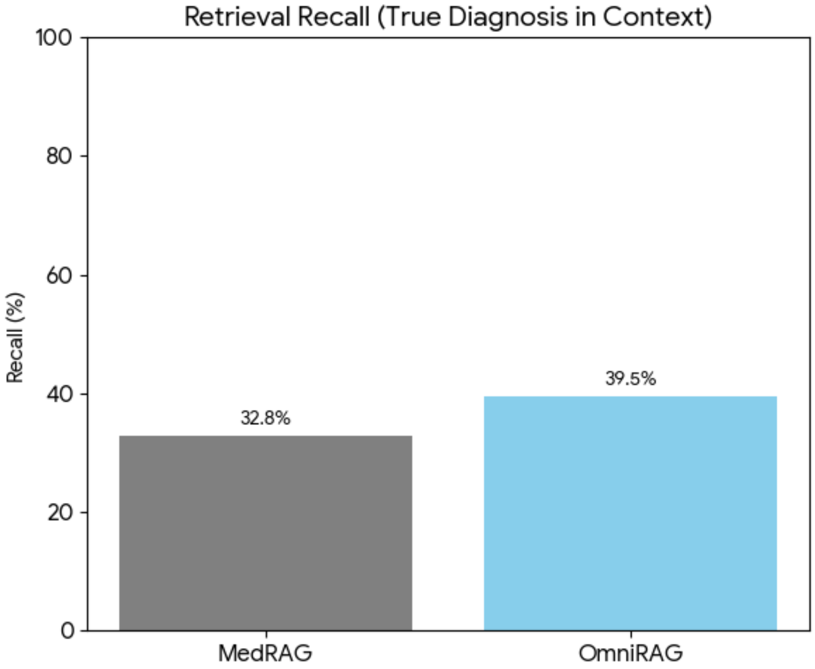


Figure 13: Retrieved Recall.