國立臺灣大學電機資訊學院資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia

College of Electrical Engineering and Computer Science

National Taiwan University

Master's Thesis

結合圖譜推理與語意搜尋以實現知識本位的醫學回應生成

Integrating Graph Reasoning and Semantic Search for Knowledge-Grounded Medical Response Generation

趙亦天

I-Tien Chao

指導教授: 周承復 博士

Advisor: Cheng-Fu Chou Ph.D.

中華民國 114 年 7 月

July, 2025

doi:10.6342/NTU202502664

國立臺灣大學碩士學位論文
口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

結合圖譜推理與語意搜尋以實現知識本位的醫學回應生成

Integrating Graph Reasoning and Semantic Search for Knowledge-Grounded Medical Response Generation

本論文係 趙亦天 （學號 R12944037）在國立臺灣大學資訊網路與多媒體研究所完成之碩士學位論文，於民國 114 年 7 月 9 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Graduate Institute of Networking and Multimedia on 9 July 2025 have examined a Master's Thesis entitled above presented by I-TIEN CHAO (student ID: R12944037) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

周承復	吳曉光	
（指導教授 Advisor）		
吳政修	吳怡青	蔡崇煒

系（所）主管 Director: 鄭卜壬

doi:10.6342/NTU202502664

Acknowledgements

在本論文完成之際，謹向所有在我碩士求學與研究歷程中給予指導與協助的師長與先進，致上最誠摯的感謝。

首先衷心感謝我的指導教授周承復老師，不僅提供我參與台大醫院耳科計畫的寶貴機會，更在每次個人會議中耐心指導，協助我釐清研究中的盲點，幫助我在研究方法與實驗設計上能有進步與成長。感謝台大醫院計畫的主持人吳振吉醫師與陳姵妤醫師，於每次會議中皆給予我們許多精闢的建議，使計畫的研究能不斷優化。此外，感謝帶領計畫的學長呂羿賢學長與楊大煒學長，在計畫會議過程中總是提供給我許多的寶貴經驗與悉心協助！

謝謝秉茂給了我很多研究上的建議，還有常常陪我去買甜點跟炸炸！謝謝如萱、冠伊、家愷、冠珅、榮浩、冠瑜和曉敏，一起討論研究和吃飯聊天都是每週的 highlight，以後要一起回去吃親來～～～

最後，感謝最支持我的家人，爸爸、媽媽、姐姐！謝謝他們常常關心我、給我很多鼓勵、買很多好吃的水果跟餅乾讓我都可以吃飽飽，還有買很多ちいかわ！

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

摘要

大型語言模型（Large Language Model）因擁有龐大的參數量與大規模的預訓練，在一般性問答與推理任務中展現出卓越的表現。於高度專業化領域中，大型語言模型生成內容的可靠性顯得尤為關鍵。此類任務對模型推理的精確性有高度要求，且模型必須避免對事實內容進行扭曲、改寫或捏造。

本研究提出一種全新的混合式檢索增強生成（Retrieval-Augmented Generation, RAG）框架，專為醫學文獻問答系統而設計。我們的系統結合了同時對知識圖譜資料庫與向量資料庫進行搜尋，實現結構化與非結構化知識的統一檢索。在離線階段，我們針對醫學領域構建專屬的醫學知識圖譜資料庫，將非結構化文本做去雜訊處理，並應用專門設計的醫學實體與關係抽取流程，系統性地提取必要內容。在線上問答階段，我們引入「圖索引機制」(Graph-Based Indexing)，將使用者查詢重新轉化為符合我們設計之知識圖結構的形式，提升檢索的精確度。於檢索階段，我們提出「圖引導式檢索」（Graph-Guided Retrieval）方法，首先查詢知識圖譜以獲得可信且結構化的資訊，並以此作為引導，進行向量資料庫的語義檢索，確保檢索結果兼具語境完整性與事實依據。在最終的答案生成階段，我們應用「圖增強生成」（Graph-Enhanced Generation）技術，對檢索結果進行重排序，優先採用最具相關性與可信度的資訊進行回答綜整。透過這種混合式的檢索與生成策略，我們不僅保留了結構化醫學知識的完整性，更優化了語義相關性，使得生成的回答在準確性、相關性及可解釋性上皆有顯著提升。

doi:10.6342/NTU202502664

我們以中文耳鼻喉科醫學語料庫進行實證評估，並採用 RAGAs（Retrieval-Augmented Generation Assessment）自動化評估工具進行量化分析，包括真實性 (Faithfulness)、答案相關性 (Answer relevance)、內容精確性 (Context precision) 與內容召回率 (Context recall) 等多項指標。為驗證系統於真實情境中的應用價值，我們亦邀請國立臺灣大學醫學院附設醫院耳鼻喉科臨床醫師參與使用者研究，針對多套問答系統從事跨維度評估。實驗結果顯示，本研究所提出之架構於長篇問答生成與專業醫學推理任務中皆顯著優於現有方法，有效驗證其於醫學知識問答領域中之實用性與可靠性。

關鍵字：大型語言模型、檢索增強生成、知識圖譜

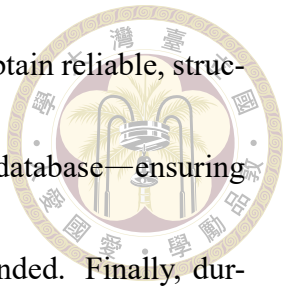
doi:10.6342/NTU202502664

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in general-purpose question answering and reasoning, owing to their extensive parameterization and large-scale pretraining. However, in specialized domains such as science and medicine, reliability becomes paramount: models must reason accurately over domain-specific knowledge while avoiding distortion, misrepresentation, or fabrication of facts.

In this work, we propose a novel hybrid Retrieval-Augmented Generation (RAG) framework specifically designed for medical literature question answering. Our approach unifies a domain-specialized knowledge graph and a vector database, enabling the retrieval of both structured and unstructured knowledge in a single, coherent pipeline. In the offline phase, we construct a medical knowledge graph by denoising unstructured texts and applying a tailored medical entity and relationship extraction pipeline. In the online phase, we introduce a graph-based indexing mechanism that reformulates user queries to align with the knowledge graph schema, thereby improving retrieval precision. Our

doi:10.6342/NTU202502664



graph-guided retrieval strategy first queries the knowledge graph to obtain reliable, structured evidence, which then guides semantic search over the vector database—ensuring retrieval that is both contextually comprehensive and factually grounded. Finally, during answer synthesis, we apply graph-enhanced generation, re-ranking retrieved results to prioritize the most relevant and trustworthy information. This hybrid retrieval–generation paradigm preserves the structural integrity of medical knowledge while enhancing semantic relevance, yielding responses that are accurate, relevant, and interpretable.

We evaluate our framework on a curated Mandarin otolaryngology corpus from the National Taiwan University Hospital, using the RAGAs (Retrieval-Augmented Generation Assessment) benchmark to measure faithfulness, answer relevance, context precision, and context recall. To assess real-world applicability, we further conduct a user study with clinicians from the Department of Otolaryngology at NTUH, who rated multiple QA systems across key quality dimensions. Results consistently show that our approach outperforms both standard RAG and GraphRAG baselines, achieving significant improvements in factual accuracy, semantic relevance, and contextual grounding—validating its effectiveness for reliable, domain-specific medical question answering.

Keywords: Large Language Model, Retrieval-Augmented Generation, Knowledge Graph

doi:10.6342/NTU202502664

Contents

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

List of Figures

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

List of Tables

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

Denotation

LLM	Large Language Model
CoT	Chain-of-Thought
RAG	Retrieval-Augmented Generation

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

Chapter 1 Introduction

Pre-trained neural language models have demonstrated remarkable capabilities in capturing linguistic patterns and factual knowledge from large-scale text corpora. Modern large language models (LLMs)—such as OpenAI's GPT-4 [1], and Meta's LLaMA series[2–4] —owing to their vast parameter counts and extensive pretraining, achieve exceptional performance on a wide range of generative and reasoning tasks. Though impressive in scope, LLMs still contend with inherent limitations. Their capacity to encode world knowledge is finite, often resulting in sparse or inconsistent coverage in specialized domains such as medicine or law. Moreover, because this knowledge is embedded in static model weights, it cannot be readily revised or expanded, making the models vulnerable to obsolescence as real-world information evolves. Compounding these challenges is the opaque nature of LLM reasoning, which not only hinders interpretability but also increases the risk of hallucination. In high-stakes contexts, these shortcomings—limited domain coverage, inflexible memory, and lack of transparency—erode confidence in the model's outputs and raise serious concerns about their trustworthiness in real-world deployment.

Retrieval-Augmented Generation (RAG) [5] extends the capabilities of large language models (LLMs) beyond the static knowledge encoded during pre-training. At inference time, RAG introduces an external memory as a dynamic knowledge source, enabling

doi:10.6342/NTU202502664

the model to retrieve relevant information before generating its response. This retrieval step grounds the output in factual evidence, thereby reducing hallucinations and improving factual consistency. Compared to supervised fine-tuning (SFT), which requires extensive labeled data and costly retraining for each target domain, RAG offers a more efficient and scalable solution: domain adaptation and real-time knowledge updates can be achieved simply by updating the retrieval corpus.

While RAG works well when essential information is concentrated in discrete text passages, its effectiveness diminishes in domains such as medicine, where knowledge is vast, dispersed across multiple sources, and inherently structured. Medical expertise is the result of decades of research and practice, represented not only in narrative form but also in standardized terminologies, taxonomies, and complex interrelationships. Although modern LLMs can handle summarization and basic question answering via in-context learning, these capabilities fall short of meeting the rigorous precision, completeness, and interpretability requirements of medical applications. As a result, there is a pressing need for specialized retrieval systems that integrate both structured and unstructured medical knowledge. Such systems should enhance LLM reasoning with accurate, standardized, and interpretable domain knowledge—minimizing hallucinations while maintaining high factual fidelity.

In this paper, we propose a hybrid database method - graph and vector based - retrieval-augmented generation framework tailored for the medical domain. Our approach integrates structured graph reasoning with semantic vector retrieval, combining the interpretability of symbolic knowledge with the flexibility of neural language models. The framework consists of two main components: (1) an offline stage for constructing a domain-specific medical knowledge graph, and (2) an online stage for graph-guided retrieval dur-

doi:10.6342/NTU202502664

ing question answering.

At the core of our system lies the construction of a Neo4j-based medical knowledge graph from unstructured Mandarin medical documents (Figure 4.1). The process follows a five-stage pipeline designed to transform complex, free-form medical text into a structured, queryable knowledge base. First, the raw medical documents undergo semantic chunking, ensuring that semantically coherent content remains within the same unit for optimal graph construction. This is followed by denoising and normalization, where the internal structure of each chunk is analyzed to identify recurring patterns, complex expressions are replaced with simpler equivalents, and potential source–target relationships are explicitly highlighted. Next, medical entities are extracted using Chain-of-Thought (CoT) [6] prompting, enabling the identification of domain-specific concepts such as diseases, symptoms, treatments with greater reasoning depth and accuracy. Using these detected entities, a second CoT-based reasoning process is applied to extract predefined relationships, with each triple anchored to its original source text to ensure traceability and factual grounding. The extracted entities and relationships are then encoded into Cypher queries and stored in the Neo4j graph database. Finally, to enhance structural coherence and minimize redundancy, a community detection algorithm is applied to cluster semantically similar entities, assigning each group a shared community identifier to support more precise and efficient retrieval.

At online query time, our system employs a novel hybrid retrieval strategy (illustrated in Figure 1.2) to seamlessly integrate structured and unstructured knowledge sources. Upon receiving a user query, a query indexing module first performs medical entity recognition and reformulates the query into a graph-aligned representation that conforms to the schema of our medical knowledge graph. This reformulated query is then executed against

doi:10.6342/NTU202502664

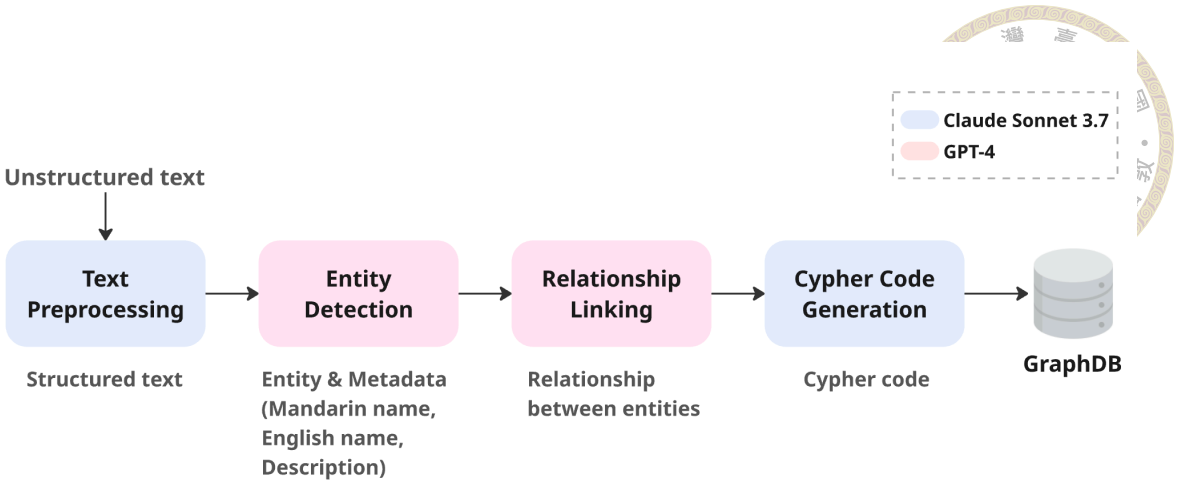


Figure 1.1: Overview of the medical knowledge graph construction pipeline. Unstructured medical documents are first preprocessed using semantic chunking, denoising, and normalization. Medical entities and relationships are then extracted via Chain-of-Thought (CoT) prompting, producing structured triples. These triples are transformed into Cypher queries for storage in the Neo4j database, followed by community detection to cluster semantically related entities.

the Neo4j database to retrieve relevant subgraphs, capturing the most pertinent entities and relationships. The retrieved graph context subsequently guides a graph-aware semantic retrieval process over the vector database, ensuring that semantic search is grounded in verified domain knowledge. The combined retrieval results are re-ranked and filtered based on contextual and semantic relevance. Finally, both structured graph-derived evidence and unstructured vector-retrieved evidence are passed to a question-answering LLM, which synthesizes the information into a factually grounded, contextually relevant final response.

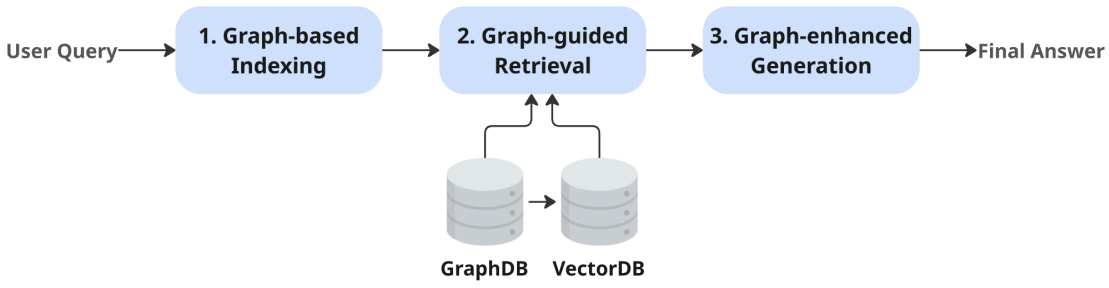
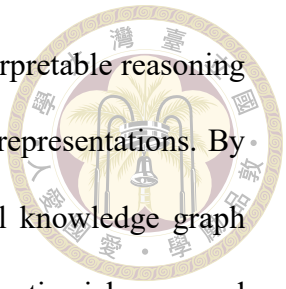


Figure 1.2: Overview of the proposed query-time workflow in our hybrid RAG framework. The process begins with query reformulation through graph-based indexing to detect the key medical entities, followed by retrieval from the medical knowledge graph and vector database. Retrieved results are then re-ranked and filtered to guide the final answer generation.

doi:10.6342/NTU202502664



Our proposed hybrid RAG architecture enables flexible and interpretable reasoning by leveraging both symbolic (graph-based) and neural (vector-based) representations. By aligning the semantic retrieval process with the underlying medical knowledge graph structure, our framework achieves accurate factual grounding, semantic richness, and adaptability to the specific requirements of Mandarin clinical question answering. Furthermore, our system enhances LLM performance by producing evidence-based responses and providing cross-lingual medical term explanations in both Mandarin and English. This dual-language capability strengthens the trustworthiness of the generated answers, supports professional clinical applications, and addresses the terminology complexity inherent in medical literature. Our key contributions are as follows:

- We present the first specialized framework for applying a hybrid RAG approach to both the Mandarin and English medical domain.
- Preservation of structured medical knowledge by reasoning over factual relationships within a domain-specific medical knowledge graph.
- Optimization of semantic search through graph-guided retrieval, enabling more targeted and context-aware information access.
- Improved accuracy, relevance, and interpretability of generated answers through the integration of structured and unstructured retrieval results.

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

Chapter 2 Preliminaries

2.1 Chain-of-Thought Prompting

Chain-of-Thought (CoT) prompting [6] is a method designed to enhance the reasoning capabilities of large language models by encouraging them to articulate a sequence of intermediate reasoning steps before producing a final answer. This approach draws inspiration from human problem-solving behaviors, where complex tasks such as solving a multi-step math word problem are typically decomposed into smaller, manageable steps before arriving at a conclusion.

An illustration of Chain-of-Thought prompting in action is shown in Figure 2.1.

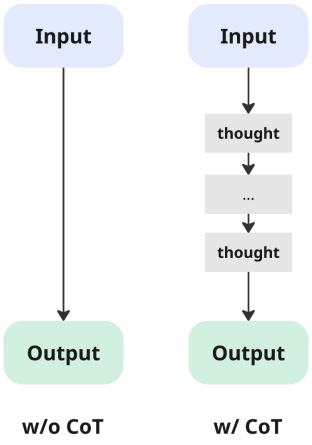


Figure 2.1: Chain-of-thought prompting enables large language models to tackle complex reasoning tasks. Instead of generating an immediate answer, CoT prompting guides the model to "think aloud," progressively unfolding its reasoning process.

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

Chapter 3 Related Work

3.1 LLM for Medicine

In the medical domain, both fine-tuned and non-fine-tuned LLMs have been explored for question answering and clinical reasoning. Models such as BioGPT[7], Med-PaLM 2 [8], and Clinical Camel [9] have been fine-tuned on biomedical corpora, showing improved performance on medical benchmarks. However, fine-tuning is often computationally intensive and less adaptable across multiple domains or languages. Alternatively, several works have focused on non-fine-tuned approaches, including advanced prompt engineering strategies [10–12] and retrieval-augmented generation pipelines tailored for medical tasks [13–15]. These methods aim to enhance medical QA performance by dynamically injecting relevant context at inference time, offering a more flexible and scalable solution compared to full model retraining.

Despite growing interest in RAG systems, their application in the medical domain remains relatively underexplored and highly specialized. Miao et al. [13] proposed a closed-domain RAG system tailored for nephrology, Xiong et al. [14] introduced a comprehensive benchmark and toolkit for evaluating RAG performance in medical tasks, using datasets such as PubMedQA and MedQA. Their work provides useful infrastructure but focuses primarily on English-language biomedical corpora and standard evaluation. Wu et

doi:10.6342/NTU202502664

al. [15] proposed Medical Graph RAG, which integrates structured graph-based retrieval with LLMs using the USMLE dataset. While this method introduces graph reasoning, it is designed around broad medical knowledge graphs and does not address language or domain-specific complexities such as Mandarin medical texts.

3.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG)[5] has emerged as a powerful approach to enhance the factual grounding of large language models (LLMs). Originally proposed by Lewis et al., RAG combines a neural retriever with a text generator, enabling LLMs to access relevant external documents during inference without requiring additional fine-tuning. This approach significantly improves factual accuracy and reduces hallucinations by grounding responses in retrieved content, the Figure 3.1 shows the steps of RAG in the LLM pipeline. More recently, GraphRAG [16] extended this paradigm by incorporating structured knowledge graphs into the retrieval process. By enabling multi-hop reasoning over entity relationships, GraphRAG supports more context-aware generation. However, its design is tailored toward general-purpose applications and lacks optimization for domain-specific challenges, particularly in complex fields such as medical literature.

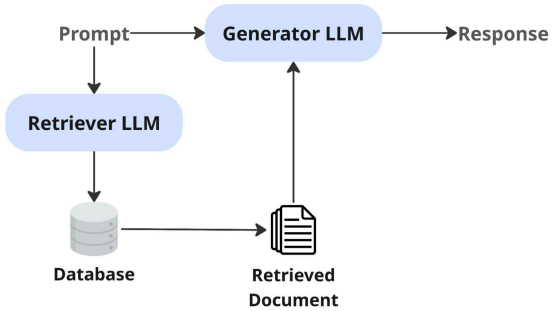


Figure 3.1: The RAG method at query time workflow.

doi:10.6342/NTU202502664

Chapter 4 Methodology

4.1 System Overview

We propose a hybrid Retrieval-Augmented Generation (RAG) framework that combines the strengths of graph-based and vector-based retrieval to enable accurate and interpretable medical question answering. The framework seamlessly integrates a structured medical knowledge graph with a semantic vector database, allowing it to leverage both symbolic reasoning and dense semantic similarity search.

The system operates in two primary stages: **(1) Offline – Database Construction:** We build a domain-specific medical knowledge graph from unstructured clinical texts, alongside a complementary vector database for fine-grained semantic retrieval. **(2) Online – Query Time Workflow:** Given a user query, the system first performs graph-based indexing and retrieval to obtain structured, contextually relevant knowledge, which then guides semantic retrieval from the vector database. Both retrieval outputs are subsequently integrated to generate a final, evidence-grounded answer.

By combining the interpretability and factual grounding of graph retrieval with the flexibility and contextual coverage of vector retrieval, our hybrid architecture delivers responses that are more faithful, semantically relevant, and clinically precise.

doi:10.6342/NTU202502664

4.2 Offline −Database Construction

In the offline phase, we construct two complementary retrieval databases that serve as the foundation for our hybrid RAG framework.

- **Graph-Based Database:** We transform unstructured medical documents into a structured, domain-specific knowledge graph. This process involves extracting medically relevant entities and linking them via predefined relationship types, resulting in an explicit semantic network that encodes factual connections between concepts. The graph structure enables interpretable, symbolic reasoning over the medical domain.

- **Vector-Based Database:** In parallel, we apply semantic-aware chunking to segment denoised document text into coherent units. Each chunk is then converted into a dense vector representation using a domain-adapted sentence embedding model. These embeddings are stored in a vector database to facilitate flexible semantic retrieval, enabling the system to capture nuanced contextual information not easily expressed in graph form.

4.2.1 Graph-Based Database

The overall pipeline for constructing the medical knowledge graph is illustrated in Figure 4.1.

doi:10.6342/NTU202502664

4.2.1.1 Text Preprocessing - Semantic Chunking

To enable structured information extraction from unstructured Mandarin medical texts, we first perform a semantic-aware preprocessing pipeline. Documents are segmented according to natural context boundaries—such as paragraphs or section headings—to ensure that semantically coherent content remains within the same chunk. To maintain compatibility with downstream components such as graph construction and language model input constraints, each chunk is further bounded by a maximum token limit. When a context block exceeds this threshold, it is subdivided at sentence-level boundaries to preserve linguistic coherence.

4.2.1.2 Text Preprocessing - Denoising and Normalization

Each resulting chunk undergoes structural analysis to identify salient medical patterns and relationships. Complex or domain-specific sentence constructions are normalized into simpler, standardized expressions to reduce ambiguity. Finally, logical relationships between source and target entities within the chunk are extracted and explicitly marked, providing the foundation for subsequent knowledge graph construction.

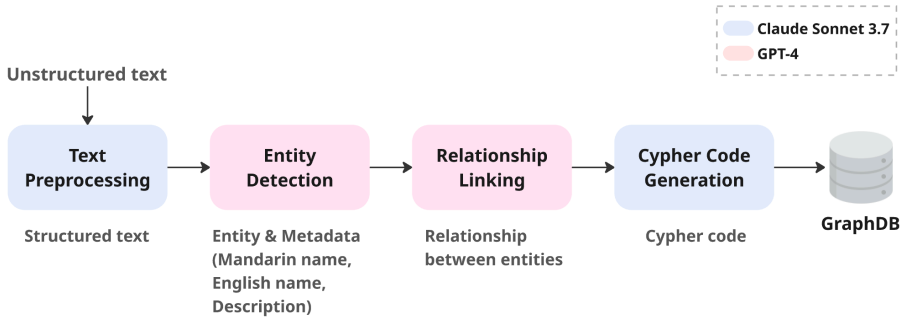


Figure 4.1: Overview of our proposed knowledge graph construction pipeline, illustrating the transformation from unstructured text to structured information stored in the database.

doi:10.6342/NTU202502664

4.2.1.3 Entity Detection and Relationship Linking

To build a domain-specific medical knowledge graph, we extract two fundamental components from unstructured medical texts:

- **Entities** —serving as the nodes of the graph, representing distinct medical concepts.
- **Relationships** —serving as the edges, capturing clinically and semantically meaningful connections between entities.

We employ a domain-adapted large language model (LLM) guided by chain-of-thought [6] prompting to process each semantically segmented text chunk. This prompting strategy encourages the model to reason in a structured, step-by-step manner, which is particularly effective for biomedical information extraction. Within each chunk, the LLM identifies medically relevant entities and classifies them into one of ten predefined categories based on established medical taxonomies: *Body, Gene, Symptom, Instrument, Examination, Chemical, Disease, Drug, Supplement,* and *Treatment*. For each detected entity, we produce a structured record containing:

1. Mandarin name
2. English name
3. Entity type
4. Concise description of its role or significance in the medical context
5. Community ID

These structured entity records constitute the foundational nodes of the knowledge graph.

doi:10.6342/NTU202502664

Following entity extraction, we perform pairwise analysis of the identified entities within each chunk to infer relationships. The LLM receives both the extracted entities and the original text as input, and is prompted with a second CoT [6] reasoning strategy specifically tailored for biomedical relationship inference. The model determines whether a meaningful relationship exists between each entity pair and, if so, classifies it into one of eight predefined medically grounded types: *CAUSES, TREATS, AFFECTS, USES, PREVENTS, LOCATION_OF, ASSOCIATED_WITH,* and *PART_OF*. Each relationship is represented as a structured triple:

$$(\text{source\_entity}, \text{relationship\_type}, \text{target\_entity})$$

These triples serve as the edges connecting the graph nodes, forming the semantic backbone of the medical knowledge graph. An illustrative example is shown in Figure 4.2.

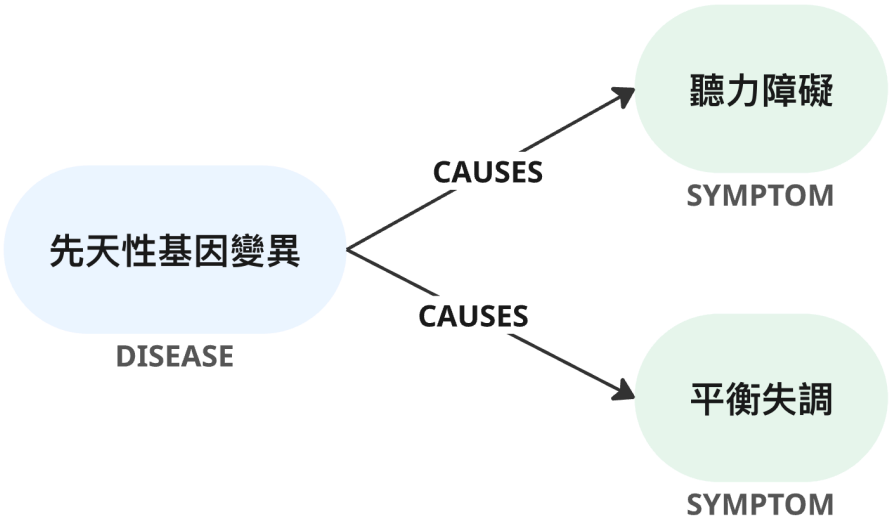


Figure 4.2: Example of medical entity and relationship extraction for knowledge graph construction. Detected medical entities are represented as nodes, while the inferred semantic and clinical relationships between them are represented as edges.

doi:10.6342/NTU202502664

4.2.1.4 Cypher Query Generation and Community Detection

Once all entities and relationships have been extracted, we proceed to automatically generate Cypher queries to populate the Neo4j graph database. Each entity and relationship is transformed into Cypher syntax, ensuring that all node attributes (e.g., Mandarin and English names, type, and description) and relationship types are correctly encoded. This step ensures a consistent and query-efficient representation of the knowledge graph. To further enhance graph structure and support higher-level reasoning, we apply a graph-based community detection algorithm—specifically, the Leiden algorithm. This algorithm clusters semantically and topologically related entities into distinct communities based on their connectivity within the graph. Entities belonging to the same community are annotated with a shared community identifier, enabling future tasks such as hierarchical reasoning, visualization, or community-level retrieval to be more effectively performed.

4.2.2 Vector-Based Database

The vector-based retrieval component is designed to capture the rich semantic context present in unstructured medical text. We begin by segmenting the original source documents into semantically coherent chunks, ensuring consistent token lengths to facilitate uniform representation. Each chunk is then encoded into a high-dimensional dense vector embedding using a pretrained domain-adapted sentence encoder. These embeddings are stored in a vector database, which supports efficient similarity-based retrieval using cosine distance. At query time, the vector-based retrieval complements the structured graph database by retrieving contextually relevant passages that may not be fully represented in the knowledge graph, thereby enhancing both coverage and semantic richness.

doi:10.6342/NTU202502664

4.3 Online - Query Time Workflow

During query time, our system performs a multi-step retrieval and generation process designed to maximize contextual relevance and factual grounding:

- Graph-Based Indexing: Given a user query, we perform entity detection and search the knowledge graph to retrieve a two-hop neighborhood of medically relevant nodes and relationships.
- Graph-Guided Retrieval: The retrieved graph context is used to rewrite and enhance the original query. This enriched query is then used to perform semantic search over the vector database.
- Graph-Enhanced Generation: Finally, both the retrieved graph context and semantically matched vector content are fed into the language model to generate a grounded, medically accurate response.

The full workflow is shown at figure 1.2

4.3.1 Graph-Based Indexing

To effectively bridge the gap between natural language queries and the structured requirements of graph-based retrieval, we introduce a query rewriting module that operates during the graph indexing stage. This component reformulates user-submitted medical questions into a graph-aware structure, aligning them with the schema of the underlying medical knowledge graph and improving retrieval precision. Upon receiving a user query, the system first performs medical entity recognition, identifying relevant terms

doi:10.6342/NTU202502664

such as diseases, symptoms, treatments, and body parts. These detected entities serve as anchors for the retrieval process. Next, the system reframes the original question using predefined linguistic patterns and graph-compatible templates. This transformation maps the user's intent onto known semantic relationships in the knowledge graph, such as ASSOCIATED_WITH, CAUSES, SYMPTOM_OF, or TREATS. Colloquial, vague, or underspecified phrases are replaced with precise, domain-specific terminology to ensure compatibility with the ontology of the graph. In cases where the user query contains multiple sub-questions, the module automatically segments them into discrete units. This decomposition ensures that each sub-query corresponds to a focused, atomic retrieval task, preventing ambiguity and improving retrieval granularity. The rewriting process also incorporates relation-sensitive keywords such as "association," "relationship", "cause," "effect," and "symptom," which are aligned with the edge labels defined in the graph schema. These cues guide the retrieval engine to target semantically meaningful subgraphs, thereby enhancing both the accuracy and interpretability of the system's outputs. An example of graph-guided indexing is shown at figure 4.3

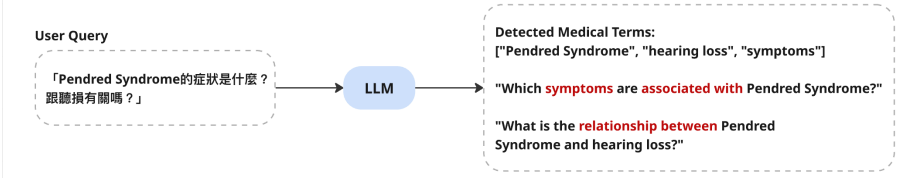


Figure 4.3: Illustration of the graph-based indexing process. Given a user-submitted medical question, the system first performs medical entity recognition to identify domain-specific terms. These detected entities are then used to reformulate the query into a structured, graph-compatible format that aligns with the knowledge graph schema. In the example shown, the medical keywords "Pendred Syndrome", "hearing loss", and "symptoms" are extracted from the query and leveraged to enable precise and schema-aligned graph retrieval.

doi:10.6342/NTU202502664

4.3.2 Graph-Guided Retrieval

Once the user query has been reformulated and relevant medical entities have been identified, the system initiates a targeted retrieval process over the Neo4j medical knowledge graph. The goal is to extract structured knowledge that is semantically aligned with the user's intent, ensuring clinical relevance and factual grounding.

Starting from the detected entities, the system performs a two-hop neighborhood search to retrieve directly and indirectly connected nodes, along with their associated relationships. This captures not only the entities explicitly mentioned in the query but also their clinically significant context—such as symptoms, causes, treatments, and related conditions—embedded within the graph structure. The resulting subgraph is then transformed into a structured, human-readable representation that preserves semantic relationships and facilitates interpretability.

To further enrich the context, the graph-derived knowledge is combined with the original user query to perform a semantic retrieval over the vector database. This hybrid strategy leverages the precision of structured graph knowledge while incorporating the breadth and nuance of unstructured text, thereby ensuring that the retrieved evidence is both knowledge-grounded and contextually relevant.

The output of this retrieval phase consists of:

- A structured summary of the graph-based retrieval results.
- A set of semantically relevant text passages retrieved from the vector database.

These results are subsequently filtered, re-ranked, and evaluated for relevance to the query.

doi:10.6342/NTU202502664

The final curated set forms a comprehensive evidence base for answer generation, balancing precision, coverage, and interpretability. An overview of this process is illustrated in Figure 4.4.

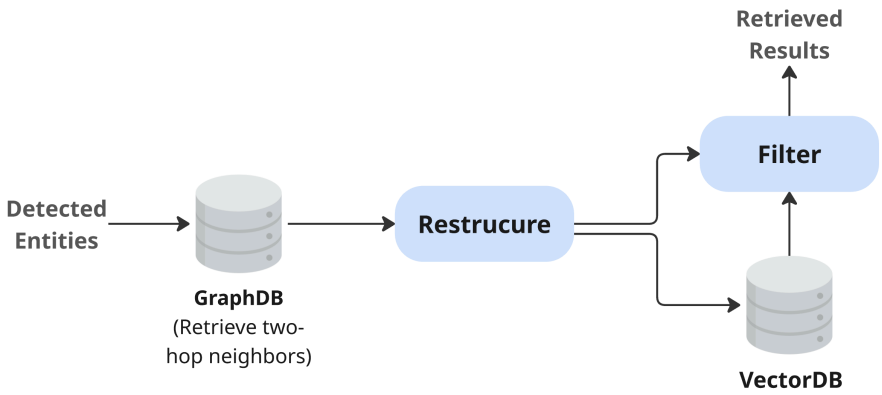


Figure 4.4: The workflow of Graph-Guided Semantic Retrieval.

4.3.3 Graph-Enhanced Generation

In the final stage of the pipeline, we perform Graph-Enhanced Generation, which synthesizes information retrieved from both the graph and vector databases to produce accurate, contextually grounded answers. The top-ranked contexts from both sources are consolidated and formatted into a unified prompt, which is passed to a large language model. The LLM leverages the structured reasoning paths provided by the knowledge graph, along with the semantic richness of unstructured text, to generate faithful, and medically accurate responses. This hybrid generation approach benefits from the precision and interpretability of graph-based retrieval, and the contextual depth and flexibility of vector-based semantic search. By combining these complementary retrieval signals, our system delivers responses that are not only relevant and well-informed, but also grounded in verifiable domain knowledge—a critical requirement for high-stakes applications like medical question answering.

doi:10.6342/NTU202502664

Chapter 5 Experiments

5.1 Experimental Settings

5.1.1 Data Source

We evaluate our proposed framework on a proprietary Mandarin medical corpus provided by the National Taiwan University Hospital. The dataset comprises approximately 660,000 words of professionally curated otology documents, covering specialized topics such as hereditary hearing impairment, gene-associated conditions, and non-syndromic hearing loss. Its clinical richness and domain specificity make it a suitable basis for constructing our medical knowledge graph.

5.1.2 Graph Database Construction

To build the graph database, the corpus undergoes semantic-aware chunking, while adhering to a maximum length of 300 tokens. Medical entities extracted are classified into ten categories: *Body, Gene, Symptom, Instrument, Examination, Chemical, Disease, Drug, Supplement,* and *Treatment*. Relationships between entities are identified according to eight predefined types: *CAUSES, TREATS, AFFECTS, USES, PREVENTS, LOCATION_OF, ASSOCIATED_WITH,* and *PART_OF*. Each entity node is further enriched

doi:10.6342/NTU202502664

with metadata, including its Mandarin and English names, a concise domain-specific description, and a community identifier derived via the Leiden algorithm.

5.1.3 Vector Database Construction

For unstructured semantic retrieval, we employ the `text2vec-base-chinese` embedding model to encode each chunk into high-dimensional dense vectors. These embeddings are stored in `ChromaDB` using cosine similarity as the distance metric. The corpus is segmented into 512-token chunks with a 10% overlap.

5.2 Evaluation with RAGAs

To assess the performance of our medical question answering framework, we adopt the **RAGAs (Retrieval-Augmented Generation Assessment)** framework [17]—an open-source benchmarking suite specifically designed for evaluating Retrieval-Augmented Generation (RAG) systems. RAGAs provides a systematic and automated methodology for measuring both the retrieval and generation components of RAG pipelines, making it particularly suitable for high-stakes domains such as medicine.

In contrast to labor-intensive manual evaluations, RAGAs enables the automatic generation of evaluation datasets from source documents, thereby ensuring scalability, consistency, and reproducibility. For this study, we employ RAGAs to evaluate our system using 100 synthetic questions automatically generated from our Mandarin medical corpus. We focus on four core evaluation metrics spanning both the *generation* and *retrieval* dimensions: **faithfulness**, **answer relevancy**, **context precision**, and **context recall**.

doi:10.6342/NTU202502664

Generation −Faithfulness: This metric quantifies the factual consistency of the generated answer with respect to the retrieved context. It measures the proportion of claims in the answer that are verifiably supported by the retrieved evidence. The score is normalized to the range $[0, 1]$, where higher values indicate greater factual alignment. Formally:

$$\text{Faithfulness} = \frac{\text{Number of claims in the answer supported by the context}}{\text{Total number of claims in the answer}}$$

Generation − Answer Relevancy: This metric evaluates how well the generated answer aligns with the user' s original query. It assesses whether the answer effectively addresses the information need expressed in the question. Operationally, Answer Relevancy is computed as the mean cosine similarity between the embedding of the original question and the embeddings of a set of pseudo-questions reverse-engineered from the generated answer. This method captures the semantic alignment between the query and the response. Let E_{g_i} denote the embedding of the i-th generated pseudo-question, E_o the embedding of the original question, and N the number of pseudo-questions. Then:

$$\text{Answer Relevancy} = \frac{1}{N} \sum_{i=1}^{N} \cos(E_{g_i}, E_o)$$

Retrieval − Context Precision: This metric evaluates the proportion of retrieved context chunks that are directly relevant to the ground truth. For each retrieved chunk, its relevance to the given question is assessed to determine whether it provides meaningful and supportive information. Precision is computed at different cutoff points k, such as Precision@1, Precision@2, and so on, where the score represents the fraction of relevant chunks among the top-k retrieved results. The overall Context Precision score is obtained by averaging Precision@k across all evaluated values of k, offering a comprehensive mea-

doi:10.6342/NTU202502664

sure of retrieval accuracy:

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} (\text{Precision@}k \times v_k)}{\text{Total number of relevant items within the top-}K\text{ results}}$$

This metric reflects how effectively the retrieval process prioritizes the most useful evidence for answering the question, thereby serving as an important indicator of retrieval quality and relevance.

Retrieval −Context Recall: This metric measures the extent to which the retrieved context encompasses all necessary information to substantiate the ground truth answer. Specifically, each claim within the reference answer is evaluated for support by the retrieved context. Context Recall thus quantifies the completeness and adequacy of the retrieval process in capturing relevant evidence essential for accurate and faithful answer generation. Formally, it is defined as:

$$\text{Context Recall} = \frac{\text{Number of claims in the reference supported by the retrieved context}}{\text{Total number of claims in the reference}}$$

These automated metrics provide a rigorous and objective framework for assessing the quality of long-form generated responses. They comprehensively evaluate factual correctness, semantic alignment, and the relevance of both the retrieved evidence and the generated answers. Table 5.1 presents a summary of the evaluation results, benchmarking our hybrid framework against two baselines: the conventional RAG and the GraphRAG models. Our approach consistently outperforms these baselines across all four key metrics, demonstrating substantial gains in factual accuracy, semantic relevance, and contextual completeness. These results substantiate the efficacy of our hybrid retrieval strategy in supporting reliable, domain-specific question answering within clinical contexts.

doi:10.6342/NTU202502664

Table 5.1: Evaluation results using RAGAs metrics.

	Faithfulness	Answer relevancy	Context precision	Context recall
RAG	0.75	0.2424	0.6388	0.6143
GraphRAG	0.6533	0.2443	0.5672	0.7844
Ours	**0.7714**	**0.3441**	**0.8750**	**0.8889**

5.3 User Study

To complement the automatic evaluation and assess the practical utility of our system in clinical settings, we conducted a user study with medical professionals. Participants included doctors from the Department of Otolaryngology at National Taiwan University Hospital (NTUH), who evaluated the quality of responses generated by different QA systems. The study focused on three key dimensions: (1) Faithfulness, which measures whether the generated answer accurately reflects the retrieved context; (2) Answer Relevancy, assessing the relevance of the answer to the original medical question; and (3) Context Precision, evaluating whether the retrieved context chunks are both relevant and necessary for answering the question. Each doctor was presented with six questions, and for each, they provided scores on a 1–10 scale for answer accuracy, answer relevancy, and context relevancy. The final scores for each metric were computed as the average of their respective ratings across all examples. This user-centered evaluation provides insight into the real-world applicability and trustworthiness of our approach in high-stakes medical domains.

The result of our user study is shown in Table 5.2.

Table 5.2: Evaluation results of user study.

	Faithfulness	Answer relevancy	Context precision
RAG	6.3333	5.8333	6.5
GraphRAG	6.8333	7.5	7.6666
Ours	**9.3333**	**9.1666**	**9.5**

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

Chapter 6 Conclusion

In this work, we have presented a novel hybrid Retrieval-Augmented Generation (RAG) framework designed to advance the reliability and interpretability of medical question answering. By integrating graph-based indexing, graph-guided semantic retrieval, and a domain-specific medical knowledge graph, our approach effectively bridges structured and unstructured knowledge sources. This synergy enables precise, contextually grounded responses to complex clinical queries, particularly within the domain of Mandarin medical literature.

Extensive evaluation on the RAGAs benchmark, coupled with assessments by experienced medical professionals, confirms the framework’s ability to significantly improve factual accuracy, answer relevance, and contextual grounding. These results underscore the potential of hybrid retrieval strategies in high-stakes, domain-specific applications where trustworthiness and interpretability are paramount.

Looking forward, we plan to extend this work by incorporating real-time knowledge updates and integrating multimodal medical data, such as imaging and structured electronic health records. These enhancements aim to further improve adaptability, enrich the depth of clinical reasoning, and expand the system’s utility in real-world healthcare environments.

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

References

[1] OpenAI. Gpt-4 technical report, 2024.

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models, 2023.

[3] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

[4] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024.

[5] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[6] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

doi:10.6342/NTU202502664

[7] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. Briefings in Bioinformatics, 23(6), September 2022.

[8] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Towards expert-level medical question answering with large language models, 2023.

[9] Augustin Toma, Patrick R. Lawler, Jimmy Ba, Rahul G. Krishnan, Barry B. Rubin, and Bo Wang. Clinical camel: An open expert-level medical language model with dialogue-based knowledge encoding, 2023.

[10] Thomas Savage, Ashwin Nayak, Robert Gallo, Ekanath Rangan, and Jonathan H Chen. Diagnostic reasoning prompts reveal the potential for large language model interpretability in medicine, 2023.

[11] Khaled Saab, Tao Tu, Wei-Hung Weng, Ryutaro Tanno, David Stutz, Ellery Wulczyn, Fan Zhang, Tim Strother, Chunjong Park, Elahe Vedadi, et al. Capabilities of gemini models in medicine, 2024.

[12] Jiaqi Wang, Enze Shi, Sigang Yu, Zihao Wu, Chong Ma, Haixing Dai, Qiushi Yang, Yanqing Kang, Jinru Wu, Huawen Hu, et al. Prompt engineering for healthcare: Methodologies and applications, 2024.

[13] Jing Miao, Charat Thongprayoon, Supawadee Suppadungsuk, Oscar A Garcia Valencia, and Wisit Cheungpasitporn. Integrating retrieval-augmented generation with large language models in nephrology: advancing practical applications. Medicina, 60(3):445, 2024.

doi:10.6342/NTU202502664

[14] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine, 2024.

[15] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, and Vicente Grau. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation, 2024.

[16] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025.

[17] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation, 2025.

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

Appendix A — Database construction setting hyperparameters

A.1 Knowledge graph construction hyperparameter settings

- `max_new_tokens = 1024`
- `top_p = None`
- `top_k = None`
- `do_sample = False`
- `temperature = None`

doi:10.6342/NTU202502664

doi:10.6342/NTU202502664

Appendix B — Entity detection prompt and Relationship linking prompt

B.1 Entity Detection System Prompt

You are a medical expert.

Given a text document that is relevant to the medical field and a list of medical entity types, identify all entities of those types from the text.

1. Identify all medical-related nouns as entities. For each identified entity, extract the following information:
   - `entity_node_property`: One of the following medical types: `[Body, Gene, Symptom, Instrument, Examination, Chemical, Disease, Drug, Supplement, Treatment]`
   - `entity_name_mandarin`: Name of the noun in Mandarin
   - `entity_name_english`: Name of the noun, capitalized and in English
   - `entity_description`: Description of the noun's attributes.

Format each entity as: `("<entity_node_property>"{tuple_delimiter}`

doi:10.6342/NTU202502664

`<entity_name_mandarin>{tuple_delimiter}<entity_name_english>{tuple_delimit`

`<entity_description>)`

2. Return all the entities identified in step 1. Use `{record_delimiter}` as the list delimiter.
3. **DO NOT** output duplicate entities.
4. When finished, output `{completion_delimiter}`.

B.2 Relationship Linking System Prompt

You are a medical expert.

1. You are given:
   - A medical text document
   - A list of extracted medical entities from the document
2. Identify all pairs of entities that have a clear and direct relationship between them.

   For each pair (*source entity*, *target entity*) that is clearly related, extract:
   - `source_entity`: the **Mandarin name** of the source entity, exactly as provided
   - `target_entity`: the **Mandarin name** of the target entity, exactly as provided
   - `relationship_type`: one of the following types: `[CAUSES, TREATS, PREVENTS, LOCATION_OF, ASSOCIATED_WITH, AFFECTS, PART_OF]`

   Important rules:

doi:10.6342/NTU202502664

- Use the following format for each relationship:

  ("relationship"{tuple_delimiter}<source_entity>{tuple_delimiter}<target_entity>{tuple_delimiter}<relationship_type>)

3. Output all relationships found. Separate multiple relationships using {record_delimiter}.

4. When finished, output {completion_delimiter}.

doi:10.6342/NTU202502664