Med-KAG, une approche de génération augmentée par connaissances : analyse des performances et limites de la récupération par embedding

Edouard HADDAG[1], Gabriel H.A. MEDEIROS[1], Lina F. SOUALMIA[1]

[1] Univ Rouen Normandie, INSA Rouen Normandie, LITIS UR 4108, FR-76000 Rouen, France

edouard.haddag@univ-rouen.fr, henrique382@gmail.com, lina.soualmia@litislab.fr

Résumé

L'adoption clinique des grands modèles de langue (LLM) est freinée par leur opacité et le risque d'hallucinations. Si la Génération Augmentée par la Récupération standard (Vector RAG) limite ces risques en sourçant l'information, elle souffre parfois d'une perte sémantique. Pour surmonter cette limite, cette étude explore l'approche GraphRAG. En structurant l'information via des graphes de connaissances, cette méthode préserve les relations complexes inhérentes au domaine biomédical, améliorant ainsi théoriquement la précision des modèles tout en réduisant les hallucinations. Dans cette optique, nous présentons l'architecture Med-KAG, un nouvel assistant d'intelligence artificielle conçu pour fiabiliser la décision clinique en intégrant un graphe de connaissances basé sur l'UMLS au paradigme RAG. Bien que l'objectif initial soit de réduire les hallucinations factuelles, une évaluation préliminaire sur le jeu de données PubMedQA révèle que cette version enrichie obtient des performances inférieures au modèle de référence (Qwen3). Une analyse approfondie identifie le module de récupération et la structure même du graphe de connaissances comme les principales sources d'erreurs. Ces résultats mettent en évidence les défis critiques liés à la qualité de l'indexation et de l'extraction pour l'intégration réussie d'architectures de type GraphRAG en médecine.

Mots-clés

Graphe de connaissances, Génération augmentée par la récupération, RAG, PubMedQA, Qwen3, Aide à la décision clinique

Abstract

The clinical adoption of Large Language Models (LLMs) is hindered by their opacity and the risk of hallucinations. While standard Retrieval-Augmented Generation (Vector RAG) limits these risks by sourcing information, it sometimes suffers from semantic loss. To overcome this limitation, this study explores the GraphRAG approach. By structuring information via knowledge graphs, this method preserves the complex relationships inherent to the biomedical domain, thereby theoretically improving model accuracy while reducing hallucinations. To this end, we introduce the Med-KAG architecture, a novel artificial intelligence assistant designed to enhance the reliability of clinical decision-making by integrating a knowledge graph into the RAG paradigm. Although the primary objective is to reduce factual hallucinations, a preliminary evaluation on the PubMedQA dataset reveals that this enriched version yields lower performance than the baseline model (Qwen3). An in-depth analysis identifies the retrieval module and the structure of the knowledge graph as the main sources of error. These findings highlight the critical challenges associated with indexing and extraction quality for the successful integration of GraphRAG architectures in the medical field.

Keywords

Knowledge Graph, Retrieval-Augmented Generation, RAG, PubMedQA, Qwen3, Clinical decision-making

1 Introduction

L'intégration de l'Intelligence Artificielle (IA) en pratique clinique offre un potentiel majeur, notamment pour l'analyse approfondie des dossiers médicaux électroniques [1]. Toutefois, l'adoption des grands modèles de langue (LLM) reste freinée par leur opacité intrinsèque et le risque persistant d'hallucinations factuelles [2].

Le paradigme de la génération augmentée par la récupération *Vector RAG* (ou RAG), introduit par Lewis *et al.* [3], consiste à coupler un modèle de langue à un système de récupération d'information. Ce dispositif permet d'adjoindre au contexte du modèle des données pertinentes et sourcées, limitant ainsi les hallucinations tout en permettant au système de traiter des domaines spécialisés absents de son entraînement initial [4]. Ce système se décompose en deux phases distinctes : l'indexation et la recherche. Lors de la phase d'indexation, les données non structurées sont transformées et regroupées au sein d'une base de données vectorielle afin d'en faciliter l'interrogation. Durant la phase de recherche, le système calcule le plongement numérique (embedding) de la requête utilisateur pour identifier les K segments de données les plus proches sémantiquement, lesquels sont ensuite fournis comme contexte au LLM.

Ce système conventionnel présente cependant des limites : malgré l'apport d'informations externes, le modèle de langue peut subir une perte de sémantique lors du traitement des données récupérées. Bien que des solutions émergent pour pallier ce problème [5], cet article explore une voie alternative fondée sur les graphes de connaissances (KG). À l'instar de la Vector RAG, les architectures de *GraphRAG* [6] structurent les connaissances sous forme de graphes. Le processus conserve la dualité indexation/recherche, mais l'indexation génère ici un KG, tandis que la recherche extrait un sous-graphe pertinent transmis au modèle.

Cette méthode s'avère particulièrement prometteuse dans le domaine biomédical, notamment car elle préserve les relations sémantiques entre les concepts et les données. Dans ce domaine, cette structuration est d'autant plus cruciale que les concepts sont interconnectés de manière complexe et standardisés via des ontologies. Des évaluations récentes [7] démontrent un gain significatif de précision et une diminution notable des hallucinations lors de l'usage du GraphRAG.

Pour garantir la fiabilité indispensable aux environnements critiques, nous proposons une nouvelle architecture d'assistant diagnostique : Med-KAG[1]. Notre approche enrichit la RAG [8] par l'intégration d'un graphe de connaissances (KG) issu du Metathesaurus de l'Unified Medical Language System (UMLS) [9]. Cette démarche s'inscrit dans une tendance de fond de l'état de l'art, où la synergie entre les graphes de connaissances et les modèles de fondation devient un paradigme central pour renforcer la factualité des réponses.

En ancrant le modèle dans cette base structurée qu'est l'UMLS, nous visons deux objectifs majeurs :

— **Minimiser les hallucinations** en contraignant les réponses par des données factuelles rigoureuses ;
— **Fournir une justification** en permettant la traçabilité du raisonnement à travers les nœuds du graphe.

Cet article présente l'architecture proposée de Med-KAG, et démontre, via une évaluation sur le jeu de données PubMedQA, comment la synergie entre RAG et KG peut influencer la précision de l'aide au diagnostic. Le choix du jeu de données PubMedQA [10] est motivé par sa prévalence dans la littérature scientifique actuelle pour l'évaluation de systèmes de référence, qu'il s'agisse de Vector RAG ou de GraphRAG [11, 7]. Notre analyse de ce corpus a confirmé sa pertinence vis-à-vis d'un KG fondé sur l'UMLS : les interrogations se concentrent sur les interactions et les effets entre diverses entités cliniques, principalement des pathologies et des agents pharmacologiques ou composés chimiques, sans inclure de problématiques liées à l'éthique médicale. Cette spécificité garantit une adéquation optimale avec la nature factuelle et relationnelle de notre base de connaissances.

Cet article est structuré comme suit : la Section 2 introduit l'architecture propsée et justifie les choix techniques opérés lors de sa mise en place ; les résultats expérimentaux, ainsi que les métriques associées, sont décrits en Section 3 ; l'analyse critique des biais et des sources d'erreurs sont discutés en Section 4 ; et enfin nous concluons et donnons quelques perspectives en Section 5.

2 Méthode

Le système Med-KAG vise à fiabiliser le diagnostic médical en substituant la récupération de texte non structuré par l'exploitation d'une base de connaissances structurée. Inspirée par le cadre MedRAG [11], notre architecture adopte une approche de *Knowledge Augmented Generation* (KAG). Bien que l'implémentation actuelle repose sur un pipeline linéaire (RAG Natif), sa conception modulaire anticipe une transition vers un RAG Modulaire [8], autorisant des flux complexes tels que l'auto-correction ou le raffinement itératif des requêtes.

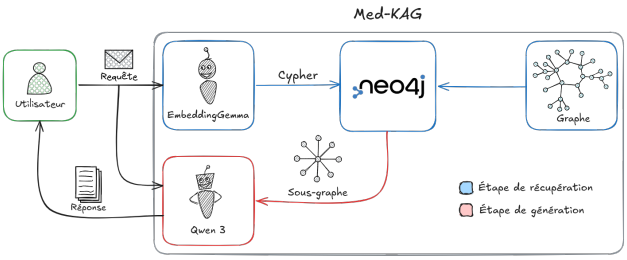

FIGURE 1 – Schéma de l'architecture du système Med-KAG.

Le cœur du système repose sur un graphe de connaissances construit à partir du Metathesaurus de l'UMLS [12]. Ce KG[2] est composé près de 13 millions de nœuds et 52 millions d'arêtes, garantissant des connexions vérifiables entre entités médicales.

Pour la phase de recherche, nous utilisons un récupérateur basé sur le modèle d'embedding EmbeddingGemma [13]. Ce modèle s'appuie sur l'architecture encodeur de Gemma pour transformer des séquences textuelles en vecteurs denses de haute dimension, optimisés pour la similarité sémantique.

Étant donnée la nature concise des entrées, qu'il s'agisse des requêtes utilisateurs ou des définitions terminologiques de l'UMLS, aucune stratégie de segmentation (chunking) n'est requise ici. Chaque élément est traité comme une unité sémantique atomique. Cette approche permet

1. https://github.com/c2fc2f/Med-KAG/tree/0.2.0

2. https://zenodo.org/records/10911980

notamment de projeter les intentions des utilisateurs et les concepts médicaux dans un même espace vectoriel, facilitant une mise en correspondance précise sans perte de contexte liée à un découpage qui pourrait s'avérer arbitraire.

Le module de recherche identifie les entités clés au sein de la requête, puis extrait un sous-graphe composé de ces nœuds et de leurs voisins immédiats. Ce contexte structuré, reflétant le voisinage sémantique pertinent, est ensuite traité par le générateur (Qwen3 [14]), choisi pour son architecture légère et performante, ainsi que son statut de modèle open-source de référence dans l'état de l'art. Sa fenêtre de contexte étendue est particulièrement adaptée aux exigences du GraphRAG, tandis que ses capacités de raisonnement (thinking) facilitent la synthèse cohérente de données relationnelles complexes.

Afin d'isoler l'impact de l'intégration du graphe de connaissances et des modules de post-traitement, nous comparons quatre configurations expérimentales distinctes :

- **Native** : le modèle Qwen3 utilisé seul, servant de référence (baseline) ;
- **RAG** : le modèle augmenté par notre module de récupération basé sur le graphe de connaissances (KG) UMLS ;
- **RAG-Cleaner** : une variante où le modèle n'est pas contraint par un format de réponse strict lors de la génération ; un modèle tiers intervient a posteriori pour formater la réponse ;
- **RAG-Sum-Cleaner** : une configuration étendue où le sous-graphe récupéré n'est pas transmis sous forme de triplets bruts, mais converti en langage naturel par un modèle intermédiaire, puis formaté par le module de nettoyage.

2.1 Prompts du Système

Cette sous-section détaille les instructions (prompts système) utilisées pour piloter les différentes configurations de l'architecture Med-KAG.

2.1.1 Configuration Native

Utilisé pour établir la référence (baseline), ce prompt demande au modèle de s'appuyer uniquement sur ses connaissances internes.

> *You are an answer generator. Your sole task is to answer a multiple-choice question. Respond with only CHOICES_KEYS corresponding to the correct choice. Do not include any other text, explanation, or introductory phrases.*

2.1.2 Configurations RAG (Vecteur)

Dans ces configurations, le contexte récupéré est injecté dans le prompt. Le modèle doit prioriser ces informations structurées.

Générateur RAG Standard (Format strict). Ce prompt impose au modèle de répondre par un seul caractère tout en lui fournissant les définitions des relations UMLS (PAR-parent, CHD-child, SY-synonym, etc.).

> *You are a rigid MCQ-answering engine. You respond ONLY with a single uppercase letter. Base your answer strictly on the Context relationships provided. If the Context is empty, use your internal knowledge.*

Générateur RAG avec Post-traitement (Cleaner). Utilisé dans la configuration **RAG-Cleaner**, ce prompt permet au modèle de générer une réponse libre avant qu'elle ne soit normalisée.

> *You are a MCQ-answering engine. Base your answer strictly on the Context relationships provided. If the Context is empty, use your internal knowledge.*

2.1.3 Modules de support

Module de Résumé (Summarizer). Ce prompt est la pierre angulaire de la configuration **RAG-Summary-Cleaner**. Il transforme les triplets bruts en langage naturel fluide.

- **Rôle :** Expert Knowledge Graph Descriptive Analyst ;
- **Instruction clé :** Effectuer une verbalisation haute fidélité des données structurées. Traduire les types de relations en verbes naturels (p.ex. : STY devient "a pour type sémantique").
- **Contrainte :** Intégrité structurelle totale ; aucune donnée du graphe ne doit être omise.

Module de Nettoyage (Cleaner). Ce module intervient en fin de chaîne pour garantir la conformité de la réponse aux exigences des benchmarks.

> *You are a precise Response Extraction Tool. Your sole purpose is to identify the correct answer choice from a provided text and output only that letter. No prose, no reasoning, no explanation.*

2.1.4 Format des requêtes

Toutes les configurations reçoivent les questions et les options de réponse selon le formatage suivant :

> *Question : REQUEST*
> *Choices : CHOICES*

2.2 Stratégie de récupération des connaissances

Alors que la section précédente détaillait l'interaction avec le modèle de langage via les prompts, cette partie décrit les mécanismes techniques permettant d'extraire les connaissances pertinentes de l'UMLS pour alimenter ces prompts.

2.2.1 Représentation vectorielle et similarité

Afin d'aligner les requêtes textuelles avec les concepts médicaux, nous utilisons le modèle d'embedding Embedding-Gemma. Étant donné la nature concise des prompts utilisateurs et des descriptions atomiques de l'UMLS, aucune segmentation (*chunking*) n'est appliquée.
La proximité entre le vecteur de la requête ($\mathbf{u}$) et les vecteurs des concepts ($\mathbf{v}$) est calculée par la **similarité cosinus** :

$$\text{similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

2.2.2 Extraction du graphe de connaissances

Une fois l'embedding généré, nous interrogeons la base Neo4j [15] dans laquelle est stocké le KG UMLS pour identifier les concepts les plus pertinents et leur contexte structurel immédiat. Nous utilisons pour cela une requête Cypher combinant recherche vectorielle et expansion de chemin :

```
CALL db.index.vector.queryNodes(
    'umls_index', 5, <embedding>
)
YIELD node
MATCH p = (node)-[*0..1]-(neighbor)
RETURN p
```

Cette méthode permet de récupérer les 5 concepts les plus proches sémantiquement ainsi que leurs voisins directs (profondeur 0 à 1). L'inclusion de ce voisinage immédiat est cruciale pour capturer les relations hiérarchiques de type (PAR/CHD) ou associatives qui enrichissent le contexte fourni aux générateurs décrits précédemment.

3 Résultats

Nous évaluons la pertinence de l'architecture Med-KAG via une étude comparative entre un modèle de référence et différentes configurations de notre système augmenté.
L'expérimentation repose sur le jeu de données PubMedQA. Bien que le benchmark MIRAGE [11] propose un cadre d'évaluation exhaustif, nous nous concentrons ici sur la tâche de Question-Réponse médicale de PubMedQA afin d'isoler l'impact de l'ancrage des connaissances sur la fiabilité du raisonnement.

La figure ci-dessous illustre un exemple typique du jeu de données :

Question : *Does desflurane alter left ventricular function when used to control surgical stimulation during aortic surgery ?*
Options : (A) yes, (B) no, (C) maybe
Réponse correcte : B

FIGURE 2 – Exemple de question à choix multiples extraite du jeu de données PubMedQA.

Nous avons mesuré la précision diagnostique en comparant les quatre configurations possibles. Contrairement aux attentes initiales, les résultats synthétisés dans le Tableau 1 mettent en évidence une dégradation de la précision lors de l'intégration du KG. Cette baisse de performance souligne les défis actuels liés à la pertinence du module de récupération et à l'intégration efficace des connaissances structurées dans le flux de génération.

Modèle	Précision	Temps (s)
Native (Qwen3-1.7B)	49,00% (245/500)	15,04
Native (Qwen3-4B)	51,60% (258/500)	108,33
RAG (Qwen3-1.7B)	39,60% (198/500)	76,21
RAG (Qwen3-4B)	37,40% (187/500)	325,18
RAG-Cleaner (Qwen3-1.7B)	46,00% (230/500)	94,23
RAG-Sum-Cleaner (Qwen3-1.7B)	37,00% (185/500)	178,27

Métrique (RAG)	Moyenne	Médiane	Écart Type
Nœuds récupérés	102.55	83,00	73.33
Arêtes récupérées	228.74	185,00	160.61

TABLE 1 – Précision comparative sur PubMedQA (500 questions) et statistiques moyennes des sous-graphes UMLS récupérés.

Les résultats du Tableau 1 montrent que les modèles natifs Qwen3 (1.7B et 4B) obtiennent les meilleures performances, avec une précision respective de 49,00% et 51,60%. L'intégration de l'architecture Med-KAG entraîne une baisse notable de la précision, chutant à 39,60% et 37,40% pour la version RAG de base.

Pour comprendre ce phénomène, nous avons analysé les caractéristiques des sous-graphes renvoyés par le récupérateur. La pertinence des informations extraites semble être le facteur limitant.

Les données révèlent une distribution asymétrique de la récupération. Alors que les requêtes réussies extraient des sous-graphes denses avec une moyenne de 102,55 nœuds et 228,74 arêtes, la médiane de 83,00 nœuds indique une grande variabilité dans la quantité d'information contextuelle fournie au générateur.

L'augmentation du temps de traitement est également significative : le passage au mode RAG multiplie par près de cinq le temps de réponse pour le modèle 1.7B (de 15,04s à 76,21s). L'analyse montre que le module **Sum-Cleaner**, bien que visant à simplifier le contexte via une transcription en langage naturel, allonge considérablement le temps de calcul (178,27s) sans restaurer la précision du modèle natif.

Pour le cas de la question clinique présentée en Figure 2, notre architecture Med-KAG a néanmoins produit une réponse correcte, illustrant le potentiel de l'ancrage sur le KG de l'UMLS, malgré les défis persistants liés au bruit introduit par le module de récupération.

La Figure 3 illustre la transparence du système : le module

de récupération a identifié les entités clés ("disturbance ; heart, functional, postoperative, cardiac surgery" et "Medication Question") et extrait son voisinage sémantique dans l'UMLS, fournissant au générateur un contexte biomédical vérifié.

4 Discussion

Les résultats de notre évaluation mettent en évidence un défi structurel majeur au sein de l'architecture actuelle. Contrairement aux attentes initiales, le système Med-KAG, avec une précision maximale de 46,00% (version **RAG-Cleaner**), n'a pas réussi à surpasser les performances du modèle natif Qwen3-1.7B (49,00%). Cette dégradation de la performance souligne que l'ajout de connaissances structurées, dans sa forme actuelle, introduit un bruit que le générateur ne parvient pas à filtrer efficacement.

L'analyse du module de récupération révèle la source principale de ces difficultés. Bien que l'objectif à long terme de ce projet soit la génération dynamique de requêtes Cypher [15] par un LLM pour naviguer avec précision dans le graphe, la présente étude s'est appuyée exclusivement sur une récupération par modèle d'embedding. Cette approche simplifiée a permis d'obtenir des résultats préliminaires en contournant les erreurs de syntaxe, mais elle se heurte à la topologie même de l'UMLS.

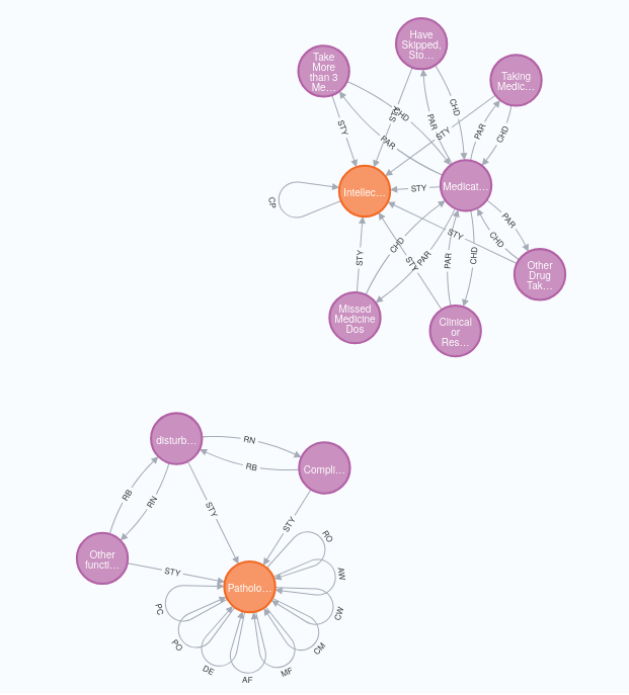

FIGURE 3 – Le sous-graphe UMLS récupéré servant de contexte ancré pour la réponse.

Dans les zones où le graphe est très connexe, la recherche vectorielle extrait des sous-graphes particulièrement volumineux (moyenne de 102,55 nœuds). Ces structures contiennent fréquemment des informations non pertinentes pour la question clinique posée, diluant ainsi le contexte utile dans un ensemble de relations périphériques bruyantes.

Pour dépasser ces limitations, deux axes d'amélioration peuvent être identifiés :

- **Le filtrage sémantique post-récupération** : l'implémentation d'un mécanisme de « reranking » ou d'élagage des nœuds permettrait de réduire la taille des sous-graphes extraits par embedding, ne conservant que les triplets ayant une influence directe sur le raisonnement diagnostique ;
- **La transition vers le Text-to-Cypher** : le passage à une génération de requêtes Cypher guidée par LLM reste la voie privilégiée. Contrairement à la recherche vectorielle qui « subit » la connexité du graphe, une requête Cypher bien formulée permettrait une traversée ciblée des relations, limitant l'extraction aux chemins sémantiques rigoureux.

Si cette approche par embedding confirme la faisabilité de l'ancrage de connaissances, elle met aussi en lumière la nécessité d'une récupération plus sélective. Les travaux futurs se concentreront sur le développement d'un moteur de génération Cypher robuste, capable de transformer l'intention clinique en une exploration précise du graphe afin de restaurer, puis d'améliorer, la fiabilité décisionnelle par rapport aux modèles de langue standards.

En résumé, bien que l'implémentation actuelle demeure perfectible, ces premiers résultats confirment la faisabilité de l'approche et isolent clairement les verrous technologiques à lever. Les travaux futurs se concentreront sur le développement d'un récupérateur Text-to-Cypher plus robuste afin d'exploiter pleinement l'ancrage dans les connaissances biomédicales et d'améliorer la fiabilité globale de l'assistance clinique.

5 Conclusions

Med-KAG introduit une architecture d'assistant IA visant à accroître la fiabilité du diagnostic médical en ancrant le paradigme RAG dans un graphe de connaissances (KG) structuré issu de l'UMLS. L'objectif principal de cette approche est de réduire les hallucinations factuelles des modèles de langue tout en offrant une meilleure explicabilité grâce à la traçabilité du raisonnement.

L'évaluation préliminaire menée sur le jeu de données PubMedQA a toutefois révélé que l'implémentation actuelle n'a pas encore surpassé les performances des modèles de référence natifs. Cette étude a permis d'isoler un goulot d'étranglement critique : l'utilisation exclusive de la récupération par embedding, qui tend à générer des sous-graphes volumineux et « bruités » dans les zones de forte connexité du graphe, diluant ainsi le contexte utile au générateur.

En outre, cette étude reste limitée à un seul jeu de données et à une unique famille de modèles open-source (Qwen3). Si l'objectif de Med-KAG est de réduire les hallucinations, cette première version évalue principalement la précision des réponses et le comportement du module de récupération, sans encore proposer de mesure dédiée aux hallucinations.

Ces premiers résultats ne remettent pas en cause la validité du concept, mais orientent précisément les futurs développements. Nos travaux futurs se concentreront sur :

- **L'optimisation technique :** la transition vers une génération de requêtes Text-to-Cypher guidée par LLM pour permettre une exploration plus ciblée et rigoureuse des relations biomédicales ;
- **L'extension de l'évaluation :** la validation du système sur d'autres jeux de données biomédicaux et sur plusieurs familles de LLM open-source ;
- **La mesure de la fiabilité :** l'intégration de métriques explicitement conçues pour quantifier la réduction des hallucinations ;
- **L'application clinique :** la validation sur des données cliniques réelles issues d'entrepôts de données de santé.

En levant ces verrous technologiques, Med-KAG ambitionne de fournir aux cliniciens un outil d'aide à la décision à la fois précis, vérifiable et sécurisé.

Références

[1] Hanan Alghamdi and Abeer Mostafa. Advancing EHR analysis : Predictive medication modeling using LLMs. *Information Systems*, 131 :102528, 2025.

[2] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, et al. A Survey on Hallucination in Large Language Models : Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.*, 43(2), January 2025.

[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[4] Orlando Ayala and Patrice Bechard. Reducing hallucination in structured outputs via Retrieval-Augmented Generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (Volume 6 : Industry Track)*, pages 228–238, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[5] Antonio Moreno-Cediel, Eva Garcia-Lopez, Antonio Garcia-Cabot, and David De-Fitero-Dominguez. Optimising retrieval performance in RAG systems : A new growing window semantic chunking strategy to address weak semantic boundaries. *Knowledge-Based Systems*, 331 :114896, 2026.

[6] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, and Apurva Mody et al. From Local to Global : A Graph RAG Approach to Query-Focused Summarization. *arXiv :2404.16130*, 2025.

[7] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, et al. Medical Graph RAG : Evidence-based Medical Large Language Model via Graph Retrieval-Augmented Generation. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 28443–28467, July 2025.

[8] Yunfan Gao, Yun Xiong, Meng Wang, and Haofen Wang. Modular RAG : Transforming RAG Systems into LEGO-like Reconfigurable Frameworks. *arXiv :2407.21059*, 2024.

[9] Olivier Bodenreider. The Unified Medical Language System (UMLS) : integrating biomedical terminology. *Nucleic Acids Research*, 32(Database issue) :D267–D270, Jan 2004.

[10] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. PubMedQA : A Dataset for Biomedical Research Question Answering. *arXiv :1909.06146*, 2019.

[11] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking Retrieval-Augmented Generation for Medicine. *Findings of the Association for Computational Linguistics : ACL 2024*, pages 6233–6251, August 2024.

[12] Gabriel H. A. Medeiros, Lina F. Soualmia, and Cecilia Zanni-Merk. Harnessing the Core Propagation Phenomenon Ontology to Develop a Knowledge Graph for Tracking Health-Related Phenomena. *Studies in Health Technology and Informatics*, 316 :1933–1937, Aug 2024.

[13] Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, et al. EmbeddingGemma : Powerful and Lightweight Text Representations. *arXiv : 2509.20354*, 2025.

[14] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 Technical Report. *arXiv :2505.09388*, 2025.

[15] Neo4j. Neo4j Cypher Manual : Overview, 2025.