Actes de CORIA-TALN 2026
Nantes, France, 29 juin au 3 juillet 2026

Les systèmes GraphRAG en pratique : vers une taxonomie des questions, des méthodes et défis d'évaluation

Carmelle Meli Songuon[1, 2] Yoan Chabot [1] Raphaël Troncy [2]
(1) Orange Research, Belfort, France
(2) EURECOM, Sophia Antipolis, France
{carmelle.melisonguon, yoan.chabot}@orange.com,
raphael.troncy@eurecom.fr

RÉSUMÉ

La génération augmentée par la récupération (RAG) s'est imposée comme une approche efficace pour enrichir les grand modèles de langue (LLMs) avec des connaissances externes afin d'améliorer leur fiabilité. Cependant, le RAG traditionnel, principalement basé sur des corpus textuels, présente certaines limites, notamment la perte de contexte global due à la segmentation des documents et l'absence de modélisation explicites de relations entre les entités. En réponse à ces limitations, le GraphRAG propose d'exploiter des structures de graphe afin de renforcer les capacités de raisonnement des LLMs. Dans cet article, nous analysons des méthodes GraphRAG existantes, en mettant en avant leurs caractéristiques spécifiques. Pour compléter cette étude, nous proposons une taxonomie des types de questions et des réponses, étroitement liées aux choix architecturaux. Enfin nous présentons des méthodes d'évaluation de ces systèmes, et plus largement des systèmes RAG.

ABSTRACT

GraphRAG Systems in Practice : Towards a Taxonomy of Questions, Methods and Evaluation Challenges

Retrieval-Augmented Generation (RAG) has emerged as an effective approach to enhance large language models by incorporating external knowledge, thereby improving their performance. However, traditional RAG, which primarily relies on textual corpora, exhibits several limitations, including the loss of global context due to document chunking and the lack of explicit modeling of entity relationships. In response to these limitations, GraphRAG aims to leverage graph structures to enhance the reasoning capabilities of LLMs. In this paper, we analyze existing GraphRAG methods, highlighting their specific characteristics. To complement this study, we propose a taxonomy of questions and responses types, which are closely related to GraphRAG architectural choices. Finally, we provide an overview of evaluation methods for these systems, and more broadly for RAG systems.

MOTS-CLÉS : GraphRAG, LLMs, graphes de connaissances, question-réponse, évaluation.

KEYWORDS: GraphRAG, LLMs, Knowledge Graphs, Question Answering, evaluation.

1 Introduction

Les grands modèles de langue (*Large Language Models*, LLMs) démontrent des capacités remarquables en traitement du langage naturel, notamment pour les systèmes de question-réponse (*Question Answering*, QA). Cependant, ils sont souvent sujets à des hallucinations, lorsqu'ils font face à des

Actes de l'atelier sur l'évaluation des modèles génératifs (LLM) et challenges (EvalLLM 2026), pages 122–140.

questions nécessitant des informations actualisées ou spécifiques à un domaine (Huang *et al.*, 2025). Le RAG (*Retrieval Augmented Generation*) est apparu comme une approche efficace pour injecter des connaissances externes dans le LLM afin de soutenir le processus de génération (Lewis *et al.*, 2020; Gao *et al.*, 2024b). Malgré ce succès, le RAG standard, principalement basé sur des corpus textuels, rencontre certaines limites. Le découpage des documents en passages indépendants peut entraîner une perte de contexte global, pénalisant les tâches qui nécessitent une compréhension globale comme la synthèse (Edge *et al.*, 2025). De plus, l'absence de structure explicite entre les informations peut introduire de la redondance, voire des incohérences, et limite les capacités de raisonnement complexe impliquant des relations entre entités ou passages (Peng *et al.*, 2025; Zhang *et al.*, 2025a).

Dans ce contexte, le GraphRAG s'est révélé être une extension prometteuse pour adresser ces limites. En exploitant la structure relationnelle des graphes, il fournit, en plus d'une mémoire externe aux LLMs, un support de raisonnement (Ma *et al.*, 2025a; Karki *et al.*, 2026). Le défi consiste alors à construire puis à utiliser des graphes de connaissances (*Knowledge Graphs*, KGs) pour permettre d'intégrer et d'organiser des données provenant de sources hétérogènes, tout en préservant leur richesse sémantique.

Dans cet article, nous proposons une analyse des approches GraphRAG existantes en les caractérisant selon le rôle du graphe, les formes d'interaction avec le LLM et les stratégies de raisonnement. Par ailleurs, certains travaux ayant montré une corrélation entre les performances de ces systèmes et les scénarios d'utilisation (Xiang *et al.*, 2026; Yu, 2025), nous proposons également une taxonomie des types de questions et de réponses afin de mieux guider les choix architecturaux. Les contributions de ce travail peuvent être résumées comme suit : (1) Une taxonomie des types de questions et de réponses, organisée selon des dimensions complémentaires et accompagnée d'un tableau récapitulatif des jeux de données utilisés dans le GraphRAG et de leurs caractéristiques. (2) Une taxonomie des approches GraphRAG centrée sur les différentes formes d'implémentation et d'interaction avec le graphe, accompagnée d'un tableau récapitulatif des différents systèmes, avec leurs descriptions et les benchmarks d'évaluation associés via un inventaire exhaustif des jeux de données utilisés.

2 Méthodologie de recherche et positionnement

Pour constituer cette revue des approches GraphRAG, nous avons principalement utilisé les bases de recherche Google Scholar, arXiv et l'anthologie ACL [1]. Notre point de départ inclut deux publications récentes (Peng *et al.*, 2025; Ma *et al.*, 2025a) afin d'avoir une vue d'ensemble du domaine, puis nous avons exploré les travaux cités ainsi que leurs références associées pour explorer le domaine de proche en proche. En parallèle, nous avons recherché des articles récents publiés à l'aide des combinaisons des mots clés "GraphRAG", "Graph", "Knowledge Graph", "Large Language Models", "Question Answering", en portant une attention particulière aux références croisées et aux baselines expérimentales. Cette démarche nous a permis de couvrir un ensemble représentatif de méthodes GraphRAG, majoritairement publiées en 2024 et 2025. La sélection plus fine des articles s'est basée sur le titre et le résumé. Au total, cette étude s'appuie sur 25 approches GraphRAG qui sont analysées en détail (Tableau 2) et sur une vingtaine de ressources décrivant les benchmarks couramment utilisés pour évaluer les systèmes de questions réponses (Tableau 3). Cette analyse nous a permis d'élaborer une taxonomie des types de questions et de réponses qui est présentée dans la section suivante.

Cette étude est complémentaire des revues de l'état de l'art publiées récemment : (Ma *et al.*, 2025a)

1. https://aclanthology.org/

catégorise les interactions entre LLMs et KGs en fonction du rôle du KG et des types de questions complexes ; (Karki *et al.*, 2026) se concentre sur le moment de l'intégration du KG dans le LLM, tandis que (Peng *et al.*, 2025) fait une catégorisation selon les différentes étapes du GraphRAG. Nous visons à compléter ces travaux en proposant une catégorisation plus fine des types de questions, des approches et des méthodes d'évaluation.

3 Analyse des types de questions dans un système RAG

3.1 Taxonomie des questions

Lors de la conception d'un système de question-réponse, il est important de déterminer les types de questions que le système est destiné à traiter (Ragas, 2025; Filice *et al.*, 2025). Ceci est d'autant plus important dans la construction d'un système GraphRAG, puisque le type de question conduit souvent à des opérations spécifiques dans le graphe (e.g., parcours de chemins *vs* extraction de sous-graphes). Il devient donc utile de pouvoir positionner un système de QA en fonction du type de questions qu'il adresse. Dans cette section, nous caractérisons les questions selon trois dimensions complémentaires, illustrées par des exemples dans le Tableau 1 (Annexe A) : (1) le degré d'ouverture, (2) la complexité de la recherche d'information, et (3) le type de raisonnement.

3.1.1 Catégorisation selon le degré d'ouverture de la question

Cette dimension capture l'ouverture de la question ainsi que la nature de la réponse attendue.

— **Questions fermées.** Elles visent à obtenir une information spécifique et exacte, sans faire intervenir de jugement personnel. Ce type de question est aussi appelé dans la littérature *questions spécifiques* (Ragas, 2025; Yu, 2025; Filice *et al.*, 2025), par exemple : « *Quelle est la capitale de la France ?* ».

— **Questions ouvertes.** Elles invitent à des réponses plus élaborées qui relèvent généralement d'une analyse, d'une synthèse, ou d'un jugement personnel. Ce type de question est aussi appelé dans la littérature *questions abstraites* (Ragas, 2025; Yu, 2025), par exemple, « *Que sait-on de la capitale de la France* » (question descriptive), « *Que se passerait-il si la capitale de la France changeait ?* » (question hypothétique), « *Comment expliquer le rôle de Paris comme capitale de la France ?* » (question de synthèse).

3.1.2 Catégorisation selon la complexité de la recherche d'information

Cette dimension capture la difficulté de la recherche d'information et du raisonnement requis pour répondre à une question.

— **Questions multi-sources.** Elles nécessitent de combiner des informations provenant de plusieurs sources distinctes (Ma *et al.*, 2025a; Wang *et al.*, 2026).

— **Questions multi-modales.** Elles nécessitent d'intégrer différentes modalités d'information (e.g., textes, images, tableaux) pour parvenir à la réponse (Wang & al., 2025; Ma *et al.*, 2025a).

— **Questions multi-tours.** Elles s'inscrivent dans un contexte conversationnel où les interactions antérieures influencent les interprétations et les réponses futures (Wang & al., 2025).

3.1.3 Catégorisation selon le type de raisonnement

Cette dimension décrit les caractéristiques des questions qui influent sur les opérations et les étapes requises pour aboutir à la réponse finale.

— **Questions factuelles.** Elles portent sur un fait simple et la réponse peut être directement extraite d'une seule source de connaissance sans traitement particulier (récupération à **un saut**) (Ragas, 2025). Par exemple, l'utilisateur peut chercher a obtenir un attribut d'une entité comme dans la question « *Quelle est la capitale de la France en 2026 ?* » (question factuelle et fermée).

— **Questions multi-sauts.** Elles nécessitent de relier plusieurs éléments d'information et d'effectuer un raisonnement en plusieurs étapes pour parvenir à la réponse finale (Ma *et al.*, 2025a; Wang & al., 2025). Par exemple, « *Quand est né le CEO actuel d'Apple ?* », est une question à deux sauts. Dans certains cas, le raisonnement prend la forme d'un chemin linéaire reliant les entités de la question à la réponse (e.g, Figure 1 (a)), et dans d'autres, il prend la forme d'un sous-graphe satisfaisant les contraintes associées à la question (e.g., Figure 1 (b)). Le premier cas peut être vu comme un cas particulier du second. Ce type de questions est souvent appelé *compositional questions* dans la littérature (Trivedi *et al.*, 2022; Pahilajani *et al.*, 2024), et des études montrent que les architectures de type GraphRAG sont particulièrement adaptées pour les traiter (Xiang *et al.*, 2026; Yu, 2025; Gutiérrez *et al.*, 2024) puisque les KGs possèdent naturellement cette structure connectant des entités sous forme de chemin ou de graphe.

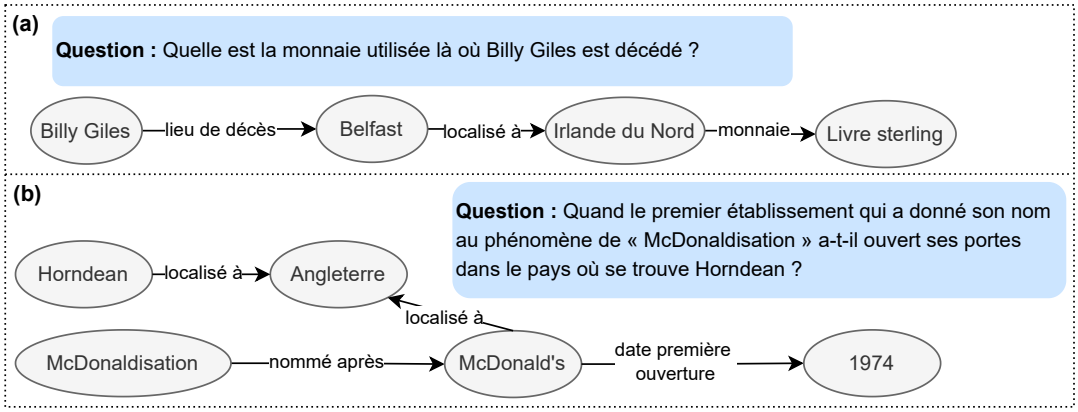

FIGURE 1 – Formes de raisonnement pour les questions multi-sauts. (a) Le raisonnement suit un chemin séquentiel de relations jusqu'à la réponse (*path-based reasoning*). (b) Le raisonnement prend la forme d'un sous-graphe qui représente les contraintes de la question (*subgraph-based reasoning*). Questions extraites du dataset MuSiQue (Trivedi *et al.*, 2022).

— **Questions temporelles.** Elles contiennent des contraintes temporelles telles que des dates précises (e.g., « *Qui était le CEO d'Apple entre 2003 et 2010 ?* »), ou des notions de temporalité relative (e.g., "le plus ancien") (Gao *et al.*, 2024a; Ma *et al.*, 2025a). À ce niveau, la source de connaissance externe et la technique utilisées pour traiter la question peuvent conduire à des réponses fausses ou incohérentes. Ainsi, lorsque les informations sont dispersées dans des morceaux de documents textuels, il peut être plus difficile d'extraire l'information correcte à cause d'un contexte fragmenté.

— **Questions analytiques.** Ces questions nécessitent des opérations ou agrégations sur un ensemble de résultats, telles que le comptage, le filtrage, le classement ou le calcul des valeurs maximales et minimales. Certains types de systèmes, notamment ceux capables d'interroger

des bases de données avec des requêtes (e.g., SQL, SPARQL, CYPHER), peuvent être particulièrement adaptés pour répondre à ce type de questions (Liu *et al.*, 2024).

— **Questions de comparaison.** Elles comparent deux entités ou plus selon un attribut commun, afin de les classer avec un opérateur d'ordre (Wang & al., 2025).

— **Questions exploratoires.** Elles n'ont pas de réponse unique ou définie. Pour y répondre, il faudrait cartographier le voisinage d'une ou plusieurs entités (e.g., « *Que sait-on sur Paris ?* »).

3.2 Taxonomie des réponses

Après avoir analysé les différents types de questions pouvant être adressées à un système QA, il est également important de s'intéresser aux types de réponses que ces systèmes QA sont amenés à produire. Dans cette partie, nous classifions les types de réponses en fonction de leur structure, en distinguant notamment les réponses textuelles des réponses structurées. Nous écartons volontairement de cette étude les questions amenées à générer des médias (graphiques, images ou vidéos) car celles-ci ne sont pas prises en compte dans ce travail.

— **Réponses textuelles.** La réponse fournie par le système est un texte non structuré ou semi-structuré. Elle peut être sous forme courte ou longue. Les **réponses courtes** renvoient généralement aux questions qui recherchent des informations factuelles et spécifiques telles que des noms, des dates ou des réponses binaires (e.g., oui/non). La formulation de la réponse est concise et directe. Les **réponses longues** quant à elles font référence aux questions ouvertes qui nécessitent plus de détails. Elles concernent également des questions fermées dont la formulation de la réponse est plus longue.

— **Réponses structurées.** Certaines questions nécessitent de produire des réponses structurées, qui peuvent être sous forme de tableaux ou de graphes. Les tableaux constituent un format couramment utilisé pour représenter de façon structurée, et parfois ordonnée, un ensemble de résultats contenant plusieurs entités accompagnées de leurs attributs. On peut également distinguer des questions visant à produire une réponse sous forme de **graphe** par exemple lorsqu'il s'agit d'identifier un chemin ou un sous-graphe expliquant les relations qui existent entre plusieurs entités. Enfin, nous notons que la réponse peut être structurée dans un format spécifique (e.g. JSON) pour être par la suite ré-utilisée par d'autres systèmes automatiques, ce qui impose aux LLMs de savoir respecter des instructions (Zhang *et al.*, 2025b). [2]

4 Analyse des systèmes GraphRAG

Dans cette section, nous commençons par présenter un aperçu général du GraphRAG, puis nous proposons une taxonomie des approches GraphRAG existantes, analysées sous différents aspects. La Table 2 (Annexe B) propose un inventaire des systèmes étudiés liés aux datasets utilisés pour leur évaluation (Table 3, Annexe C).

4.1 L'approche GraphRAG

L'approche GraphRAG vise à exploiter la représentation structurée des graphes afin d'améliorer les capacités des modèles de langage (Xiang *et al.*, 2026; Karki *et al.*, 2026). Son principe général consiste

2. https://github.com/xiaoya-li/Instruction-Tuning-Survey

d'une part à augmenter un modèle de langage avec des informations pertinentes récupérées dans des graphes, et d'autre part, à utiliser la représentation structurée et relationnelle de ces graphes pour soutenir le raisonnement. Selon la technique utilisée et la source de connaissance externe disponible, les étapes du système peuvent varier. On retrouve généralement les trois étapes principales d'une architecture RAG classique que nous détaillons dans la suite.

L'indexation. Cette première étape vise à collecter, organiser et stocker les informations provenant des sources de connaissances sous une forme facilitant leur récupération lors de la phase d'inférence. Certains travaux qui se basent sur des corpus textuels construisent un graphe à partir de ceux-ci (Sarthi *et al.*, 2024; Gutiérrez *et al.*, 2024; Edge *et al.*, 2025), tandis que d'autres utilisent directement des graphes existants tels que Wikidata (Sun *et al.*, 2024a; Walter & Bast, 2025). Par la suite, des index vectoriels peuvent être construits pour permettre une récupération par similarité sémantique (He *et al.*, 2024; Sapidis *et al.*, 2025), ou des index qui conservent explicitement la structure du graphe afin de permettre un raisonnement par parcours du graphe (Ma *et al.*, 2025b; Xu *et al.*, 2024; Sun *et al.*, 2024b). Cette étape initiale d'indexation, voire de construction du graphe de connaissances, peut être absente lorsque le système exploite directement les API et outils fournis par la base de connaissances pour l'accès aux informations. C'est notamment le cas de SPINACH (Liu *et al.*, 2024) qui utilise l'API de Wikidata et son service de requête SPARQL.

La récupération des informations. Dans cette étape, des informations pertinentes sont extraites de la base de connaissances indexée pour aider à répondre à la question. La méthode appliquée varie selon le type d'indexation faite, ce qui conditionne également le type d'éléments récupérés (des nœuds d'un graphe, des fragments de texte, des triplets ou des sous-graphes). Pour une indexation sous forme de graphe, des méthodes comme (Sun *et al.*, 2024a,b) parcourent le graphe pour collecter progressivement des entités et relations qui permettront de répondre à la question. Avec une indexation vectorielle, des méthodes comme (He *et al.*, 2024; Hu *et al.*, 2025; Sarthi *et al.*, 2024) procèdent par un calcul de similarité pour récupérer les éléments du graphe. Nous détaillons ces techniques dans la Section 4.2.3.

La génération de la réponse. Une fois les informations pertinentes récupérées, elles sont ajoutées au contexte du LLM, avec la requête d'origine pour générer la réponse finale. En fonction du type et de la structure des données récupérées, une étape intermédiaire de verbalisation peut-être nécessaire pour convertir les informations en un format facilement exploitable par le LLM. Le contexte peut alors être injecté directement dans la fenêtre de contexte du LLM sous forme textuelle (Gutiérrez *et al.*, 2025; Sarthi *et al.*, 2024; Sun *et al.*, 2025), ou au niveau des représentations vectorielles du modèle (Hu *et al.*, 2025; He *et al.*, 2024). La section 4.2.2 détaille ces différentes méthodes d'injection du graphe dans un modèle de langage.

4.2 Taxonomie des systèmes GraphRAG

Nous proposons une catégorisation des systèmes GraphRAG suivant trois axes complémentaires : le rôle du graphe, l'injection du graphe dans le modèle de langage et les stratégies de raisonnement utilisées (Figure 2).

4.2.1 Le rôle du graphe

L'utilisation du graphe peut intervenir à différents niveaux dans un système GraphRAG. On distingue les approches qui exploitent un graphe existant et celles qui en construisent un à partir de corpus

textuels. Dans le premier cas, le graphe peut servir de source principale ou source partielle de connaissances (combiné avec des textes) (Ma *et al.*, 2025b; Sarmah *et al.*, 2024; Sapidis *et al.*, 2025). Il peut également permettre de guider le raisonnement du LLM (Ma *et al.*, 2025a). Par exemple, RoG (Luo *et al.*, 2024b) utilise un LLM entraîné sur Freebase pour prédire des chemins de relations servant de plan de raisonnement pour la récupération et la génération. Dans le second cas, le graphe est construit à partir des textes. Par exemple, HybridRAG (Sarmah *et al.*, 2024) génère un graphe de connaissances, tandis que RAPTOR (Sarthi *et al.*, 2024) construit un arbre hiérarchique de résumés. Le graphe peut aussi servir à étendre la recherche. HippoRAG (Gutiérrez *et al.*, 2024) et HippoRAG-2 (Gutiérrez *et al.*, 2025) construisent un KG à partir de textes, puis l'utilisent pour identifier des nœuds pertinents avant de fournir au LLM les passages textuels associés. Ainsi, le graphe construit sert d'outil de recherche plutôt que de source directe de contenu. Par ailleurs, il est important de noter que la qualité du graphe, qu'il soit construit ou pré-existant, influence la qualité de la réponse produite. Par exemple, des structures incomplètes peuvent limiter un raisonnement multi-sauts (Chepurova *et al.*, 2026).

4.2.2 Injection du graphe dans le modèle de langage

Cette caractérisation constitue une distinction importante dans les systèmes GraphRAG car elle vise à identifier où et comment les connaissances récupérées sont intégrées dans le LLM. On peut distinguer trois stratégies principales d'injection :

- **L'injection dynamique.** Les informations issues du graphe sont linéarisées et intégrées sous forme textuelle dans la fenêtre de contexte du LLM au moment de l'inférence.
- **L'injection statique.** Les connaissances du graphe sont intégrées directement dans les paramètres du modèle, généralement via un processus de *fine-tuning*. Cette approche permet d'intégrer de manière permanente les informations ou la structure du graphe dans les poids du LLM (Luo *et al.*, 2024b,a).
- **L'injection modulaire.** Cette stratégie consiste à étendre le modèle de langage de base par l'ajout de modules spécialisés capables d'encoder et d'exploiter la structure du graphe, tels que des réseaux de neurones de graphe (*Graph Neural Network*, GNN). Elle vise ainsi à préserver l'information topologique qui peut être perdue lors de la linéarisation du graphe (Gao *et al.*, 2024a). Par exemple, KG-Adapter (Tian *et al.*, 2024) introduit un module adaptateur conçu pour les LLMs de type décodeur, capable d'encoder les informations du KG et de permettre un raisonnement conjoint avec le modèle de langage pour générer la réponse.

Ces différentes méthodes présentent des compromis différents entre flexibilité, performances et coût. Ainsi, la méthode adéquate dépendra des scénarios d'utilisation ainsi que des ressources disponibles (Song *et al.*, 2025).

4.2.3 Différentes stratégies utilisées dans un système GraphRAG

Nous distinguons cinq grandes catégories de méthodes, selon la manière dont le graphe est interrogé et utilisé pour traiter la requête.

Text2SPARQL (parsing sémantique). Ces méthodes reposent sur la génération d'une forme logique ou d'une requête à partir de la question en langage naturel, qui est ensuite exécutée sur un KG afin d'obtenir la réponse. Dans certains systèmes, un LLM est entraîné pour apprendre la structure du graphe afin d'améliorer la génération des requêtes (e.g., ChatKBQA (Luo *et al.*, 2024a)). Dans

d'autres cas, le LLM est utilisé comme un agent, équipé d'outils lui permettant d'explorer le graphe de façon autonome pour construire progressivement la requête SPARQL (Liu *et al.*, 2024; Walter & Bast, 2025). Cette approche basée sur la génération d'une requête exécutable offre généralement une bonne précision lorsque la requête générée est correcte, et est particulièrement adaptée aux questions nécessitant d'effectuer des opérations complexes telles que des filtres et des agrégations.

Recherche de chemins pertinents. Cette catégorie regroupe les méthodes qui exploitent les chemins dans le graphe comme support de raisonnement. L'objectif est de construire des chaînes de triplets reliant les entités de la question à la réponse. Par exemple, ToG (Sun *et al.*, 2024a) et GoG (Xu *et al.*, 2024) identifient premièrement les entités présentes dans la question, puis itèrent des étapes d'exploration et de filtrage afin de construire progressivement des chemins pertinents. Cette stratégie de raisonnement est appropriée pour les questions multi-sauts, dont la résolution peut être formulée comme une chaîne séquentielle de relations reliant les entités de la question à la réponse.

Recherche de sous-graphes pertinents. Ces méthodes visent à extraire un sous-graphe pertinent permettant de répondre à la question, plutôt que de construire un chemin unique reliant les entités. Différents niveaux de granularité peuvent être considérés, allant des nœuds ou triplets individuels à des sous-graphes plus complexes. Par exemple, HybridRAG récupère des triplets contenant les entités mentionnées dans la question, tandis que GRAG (Hu *et al.*, 2025) sélectionne les sous-graphes pertinents préalablement indexés, sur la base de leur similarité vectorielle avec la question.

Recherche de passages pertinents. Ces méthodes utilisent un graphe pour identifier les passages de texte pertinents à intégrer dans le contexte du LLM afin de générer la réponse finale. Dans ce contexte, certaines construisent un graphe qui préserve la relation d'inclusion entre les entités et les passages, puis se servent des entités apparaissant dans la question pour parcourir le graphe et récupérer, au final, les passages importants (Gutiérrez *et al.*, 2024, 2025).

Approches agentiques. Ces approches modélisent le processus de QA comme une séquence d'actions réalisées par un agent basé sur un LLM et équipé d'un ensemble d'outils lui permettant d'interagir avec le graphe. Elles introduisent un caractère hybride au niveau des stratégies de raisonnement dans la mesure où l'agent peut combiner différentes formes d'interaction avec le graphe (e.g, générer des requêtes, récupérer des entités et triplets). L'objectif final peut être soit de construire progressivement la requête permettant de répondre à la question (comme dans SPINACH), soit de produire directement la réponse finale (comme dans RoboData (Musumeci *et al.*, 2025)). Par ailleurs, certains systèmes reposent sur des architectures multi-agents, dans lesquelles plusieurs agents spécialisés collaborent en accomplissant différentes sous-tâches (Zhao *et al.*, 2025).

Approches basées sur une récupération hybride. Il s'agit des approches qui combinent la récupération des éléments du graphe avec des fragments de texte provenant des corpus non structurés. L'objectif est de tirer parti des avantages complémentaires de ces deux types de sources. HybridRAG illustre bien cette approche en combinant le RAG vectoriel et le GraphRAG. Leurs résultats montrent que cette combinaison offre un meilleur équilibre en termes de performance.

De manière générale, les différentes stratégies discutées ci-dessus présentent des compromis entre coût computationnel, précision et scalabilité. Les méthodes basées sur l'exploration du graphe peuvent s'avérer coûteuses en raison du nombre d'appels au LLM nécessaires pour naviguer dans le graphe et filtrer les informations pertinentes. Les approches basées sur la récupération de sous-graphes quant à elles, impliquent un coût non négligeable lié à l'indexation du graphe et au calcul de similarité, en particulier pour un graphe de grande taille comme Wikidata (Liu *et al.*, 2024). Par ailleurs, ces deux familles d'approches peinent à traiter certaines questions nécessitant de récupérer un grand nombre

d'entités, éventuellement combinées à des opérations de calcul, en raison des limites de taille des sous-graphes pouvant être extraits. Les approches agentiques, bien que flexibles, reposent souvent sur des architectures complexes, qui entraînent une croissance importante de la mémoire, accompagnée d'une possible dégradation des capacités de raisonnement du LLM (Sun *et al.*, 2025; Ledel, 2025).

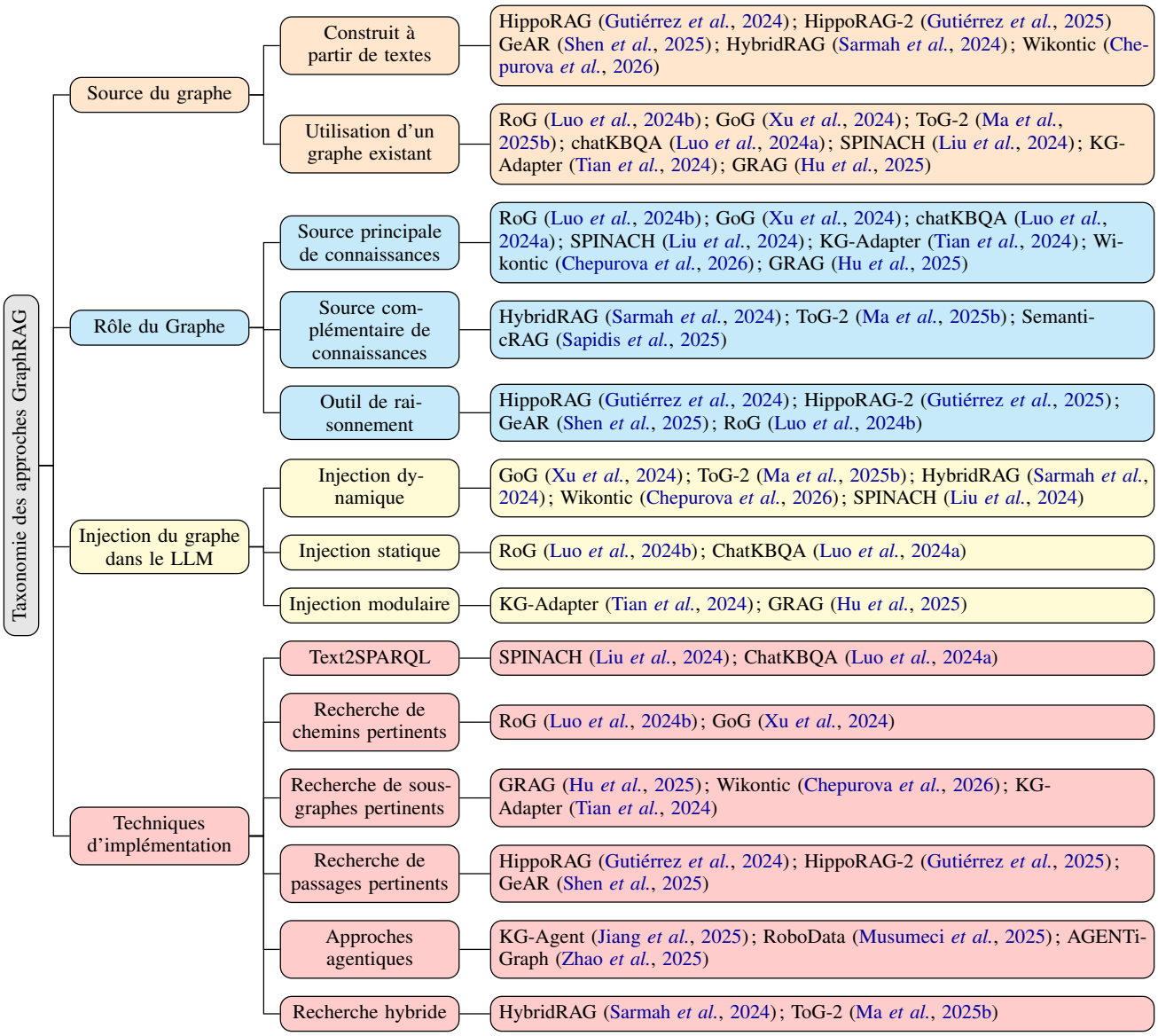

FIGURE 2 – Taxonomie des méthodes GraphRAG.

5 Évaluation des systèmes RAG et GraphRAG

L'évaluation des réponses (ou prédictions textuelles) peut reposer sur des métriques lexicales, sémantiques (à base de similarités entre représentations vectorielles), ou être intermédiées (*LLM-as-a-Judge*) (Lemesle *et al.*, 2024; Gu *et al.*, 2025; Xiang *et al.*, 2026). Par ailleurs, certaines approches intègrent l'évaluation humaine pour avoir une appréciation plus fine, mais au prix d'un coût plus important. Dans cette section, nous présentons les différentes méthodes couramment utilisées pour évaluer les systèmes RAG selon quatre aspects complémentaires. Le Tableau 2 (Annexe B) présente

les métriques utilisées par les systèmes GraphRAG que nous avons recensés avec les jeux de données associés.

5.1 Évaluation de la qualité de la génération

— **L'exactitude de la réponse.** Elle évalue à quel point la réponse prédite par le système est correcte, en se basant généralement sur une réponse de référence. Les métriques lexicales incluent : **BLEU** (Papineni *et al.*, 2002), **ROUGE-L** (Lin, 2004) et la **correspondance exacte** (*Exact Match*, EM). Dans certains cas, l'implémentation de l'EM est assouplie, en considérant une réponse comme correcte si elle contient la réponse de référence (ou inversement) (Sun *et al.*, 2024a; He *et al.*, 2024). Pour des questions à réponses multiples, notamment dans les jeux de données de type KGQA (e.g., WebQSP (Yih *et al.*, 2016)), où la réponse est souvent une liste d'entités, d'autres métriques sont privilégiées telles que le **Hit@1** ou le **score F1**. Pour mesurer la similarité sémantique, les métriques comme **BERTScore** (Zhang *et al.*, 2020) ou **ParaPLUIE** (ParaPhrase, Llm Used for Improved Evaluation) (Lemesle *et al.*, 2024) sont utilisées. Enfin, l'évaluation des réponses structurées sous forme de tableaux (e.g., la sortie des requêtes SPARQL) utilise des métriques telles que la précision, le rappel ou le score F1 calculé au niveau des cellules ou des lignes. En particulier, SPINACH (Liu *et al.*, 2024) propose une généralisation des métriques EM et F1 dans laquelle les colonnes supplémentaires prédites mais absentes dans la réponse de référence ne sont pas pénalisées.

— **La couverture de la réponse.** Elle permet d'évaluer la quantité de connaissances essentielles contenues dans la réponse de référence qui est reprise dans la réponse générée (Xiang *et al.*, 2026; Carmel *et al.*, 2025).

— **La fidélité de la réponse.** Cette dimension vise à évaluer si la réponse prédite s'appuie bien sur le contexte récupéré et évite les hallucinations (ES *et al.*, 2024; Xiang *et al.*, 2026). Une méthode courante utilisée pour calculer la fidélité consiste à extraire des faits atomiques de la réponse générée, puis à évaluer la proportion de ceux qui découlent du contexte extrait en interrogeant un LLM (Carmel *et al.*, 2025; Sarmah *et al.*, 2024; Xiang *et al.*, 2026). (Argese *et al.*, 2026a) introduit le *StoryScore* comme une métrique agrégée permettant d'évaluer la qualité de narratifs générés par des LLMs, en prenant en compte des aspects créatifs (e.g. génération d'une analogie explicative) tout en veillant à ne pas introduire d'hallucinations. Cette métrique est notamment utilisée par le système SciTeller [3] qui permet de générer des narratifs d'articles scientifiques adaptés à des personas (Argese *et al.*, 2026b).

— **La pertinence de la réponse par rapport à la question.** Cette métrique vise à évaluer dans quelle mesure la réponse générée répond à la question initiale. Pour l'évaluer, (ES *et al.*, 2024; Sarmah *et al.*, 2024) utilisent un LLM pour générer n questions potentielles à partir de la réponse fournie, puis calculent la moyenne des similarités cosinus entre la représentation de chaque question générée et celle de la question d'origine.

— **L'attribution des sources.** Elle permet d'évaluer si les références citées par le système RAG sont pertinentes et correctement attribuées. Pour cela, CiteFix (Maheshwari *et al.*, 2025) vérifie, pour chaque assertion accompagnée d'une référence, si celle-ci figure dans le document cité.

3. https://sciteller.tools.eurecom.fr/

5.2 Évaluation de la qualité de la récupération d'information

Cette évaluation se concentre sur le contexte récupéré de la base de connaissances et utilisé pour générer la réponse. Elle varie selon la présence d'un contexte de référence ou non, permettant de mesurer la pertinence des résultats (Ip & Vongthongsri, 2024). Dans le cas où un contexte de référence est présent, comme dans les jeux de données (Yang *et al.*, 2018; Trivedi *et al.*, 2022), des métriques usuelles de recherche d'information telles que la précision et le rappel sont utilisées pour évaluer les phrases ou paragraphes d'appui. (Xiang *et al.*, 2026), par exemple, calcule le rappel des preuves, afin de vérifier si tous les éléments essentiels nécessaires pour répondre correctement à la question sont présents dans le contexte extrait, en utilisant un LLM. Par ailleurs, certains jeux de données tels que 2WikiMultihopQA (Ho *et al.*, 2020), contiennent pour chaque question, une liste de triplets d'appui de référence extraits de Wikidata. WebQSP (Yih *et al.*, 2016), quant à lui, contient des chaînes de relations parcourues dans Freebase pour aboutir à la réponse. En l'absence d'un contexte de référence, (ES *et al.*, 2024) mesure la pertinence du contexte récupéré en utilisant d'abord un LLM pour extraire un ensemble de phrases essentielles permettant de répondre à la question, puis calcule le rapport entre le nombre de phrases extraites et le nombre total de phrases présentes dans le contexte.

5.3 Qualité du raisonnement et de l'explicabilité

Cette dimension évalue l'interpretabilité ou l'explicabilité des systèmes RAG et d'IA en général. Les approches d'interprétabilité reposent sur des méthodes post-hoc visant à identifier les facteurs qui influencent la prédiction du modèle (Chaduvula *et al.*, 2026) (e.g. LIME (Ribeiro *et al.*, 2016), SHAP (Lundberg & Lee, 2017)). D'autres approches plus récentes en interprétabilité mécanistique visent à analyser directement les unités abstraites internes des modèles (e.g. les têtes d'attention de Transformers) afin d'identifier les circuits computationnels responsables de leur comportement (Conmy *et al.*, 2023). Avec l'émergence des systèmes d'IA agentique, l'évaluation de l'explicabilité s'est déplacée vers l'analyse des traces de raisonnement et d'actions produites par l'agent. Pour cela, DeepEval (Ip & Vongthongsri, 2024) décompose le comportement de l'agent en étapes de raisonnement, d'action, d'exécution, et s'appuie sur un LLM pour évaluer leurs qualités.

5.4 Efficacité computationnelle

Dans une optique d'une IA frugale, d'autres critères sont pris en compte dans l'évaluation des systèmes RAG notamment : le temps d'indexation et de réponse, le coût en nombre de tokens (Xiang *et al.*, 2026; Xiao *et al.*, 2025), la taille des modèles utilisés, le nombre d'appels aux LLMs et aux outils (Musumeci *et al.*, 2025), ainsi que les ressources de calcul nécessaires (Gutiérrez *et al.*, 2025). Les résultats montrent que comparés au RAG traditionnel, les systèmes GraphRAG augmentent significativement la longueur des prompts.

5.5 Discussion sur les limites des benchmarks GraphRAG existants

Malgré les avancées récentes, l'évaluation des systèmes GraphRAG souffre encore d'un manque de benchmarks standardisés permettant une évaluation rigoureuse et une comparaison entre approches (Peng *et al.*, 2025). Plusieurs travaux existants se concentrent principalement sur l'évaluation

de la réponse finale, au détriment d'autres aspects tels que la récupération d'information ou le raisonnement. Cette limitation s'explique en grande partie par l'absence de jeux de données adaptés, et par ailleurs multilingues, qui fournissent des annotations sur la récupération ou le raisonnement (e.g. des sous-graphes ou des chaînes de triplets de référence, les séquences d'actions de référence). (Xiang *et al.*, 2026), cherchant à adresser ce problème, a proposé GraphRAG-Bench, [4] un benchmark qui offre une évaluation plus complète des systèmes GraphRAG, en couvrant la construction du graphe, la récupération des connaissances et la génération finale. Cependant, des défis subsistent quant à l'existence de benchmarks plus généraux capables de prendre en compte la diversité des approches GraphRAG et de faciliter leur comparaison. Par exemple, il reste difficile de comparer des approches basées sur l'exploitation de graphes de connaissances pré-existants avec celles construisant un graphe à partir de corpus textuels.

6 Conclusion et travaux futurs

La génération augmentée par la récupération basée sur des graphes (GraphRAG) a émergé comme une extension prometteuse du RAG traditionnel, permettant d'améliorer les capacités de raisonnement et de réponse des modèles de langage. Dans ce travail, nous avons proposé une analyse des approches GraphRAG existantes, en les caractérisant selon le rôle du graphe, les formes d'intégration et de raisonnement avec celui-ci. Nous avons également introduit une taxonomie des types de questions et de réponses pouvant être adressées, ainsi qu'une vue d'ensemble des méthodes d'évaluation du GraphRAG, et plus largement du RAG. Nos analyses montrent qu'il est difficile pour une approche unique d'être optimale pour l'ensemble des types de questions, les choix architecturaux étant liés à la nature des tâches. Elles mettent également en évidence le besoin de benchmarks plus complets et mieux adaptés pour évaluer les systèmes GraphRAG. Enfin, ce domaine en rapide évolution ouvre des perspectives d'amélioration telles que le passage à l'échelle, l'intégration de données multimodales et la mise à jour dynamique des connaissances dans les systèmes GraphRAG.

Remerciements

Ce travail a été partiellement financé par le programme i-Demo de la Banque Publique d'Investissement (Bpifrance) au sein du projet LettRAGraph (projet DOS0256163/00).

Références

ARGESE A., LISENA P. & TRONCY R. (2026a). Hallucination or creativity : How to evaluate AI-generated scientific stories ? In CEUR-WS, Éd., *9th International Workshop on Narrative Extraction from Texts (Text2Story)*, Delft, The Netherlands.

ARGESE A., SILLANO A., LISENA1 P., TRONCY R., CALÒ T. & RUSSIS L. D. (2026b). SciTeller : An LLM-Based Framework for Persona-Adaptive Scientific Storytelling. In CEUR-WS, Éd., *2nd Workshop on Sustainable and Trustworthy Large Language Models for Personalization (LLM4Good)*, Gothenburg, Sweden.

4. https://github.com/GraphRAG-Bench/GraphRAG-Benchmark

CARMEL D., FILICE S., HOROWITZ G., MAAREK Y., SOMEKH O. & TAVORY R. (2025). The LiveRAG Challenge at SIGIR 2025. In *48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, p. 4199—-4201.

CHADUVULA S., HO J., KIM K., NARAYANAN A., ALINOORI M., GARG M., RAMACHANDRAM D. & RAZA S. (2026). From Features to Actions : Explainability in Traditional and Agentic AI Systems. DOI : https ://doi.org/10.48550/arXiv.2602.06841.

CHEPUROVA A., BULATOV A., BURTSEV M. & KURATOV Y. (2026). Wikontic : Constructing Wikidata-Aligned, Ontology-Aware Knowledge Graphs with Large Language Models. In *19th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, p. 8304–8319.

CONMY A., MAVOR-PARKER A. N., LYNCH A., HEIMERSHEIM S. & GARRIGA-ALONSO A. (2023). Towards Automated Circuit Discovery for Mechanistic Interpretability. In *37th International Conference on Neural Information Processing Systems*.

EDGE D., TRINH H., CHENG N., BRADLEY J., CHAO A., MODY A., TRUITT S., METROPOLITANSKY D., NESS R. O. & LARSON J. (2025). From Local to Global : A Graph RAG Approach to Query-Focused Summarization. DOI : https ://doi.org/10.48550/arXiv.2404.16130.

ES S., JAMES J., ANKE L. E. & SCHOCKAERT S. (2024). RAGAs : Automated Evaluation of Retrieval Augmented Generation. In *18th Conference of the European Chapter of the Association for Computational Linguistics (EACL) - System Demonstrations*, p. 150–158.

FILICE S., HOROWITZ G., CARMEL D., KARNIN Z., LEWIN-EYTAN L. & MAAREK Y. (2025). Generating Diverse Q&A Benchmarks for RAG Evaluation with DataMorgana. DOI : https ://doi.org/10.48550/arXiv.2501.12789.

GAO Y., QIAO L., KAN Z., WEN Z., HE Y. & LI D. (2024a). Two-stage Generative Question Answering on Temporal Knowledge Graph Using Large Language Models. In *Findings of the Association for Computational Linguistics (ACL)*, p. 6719–6734. DOI : 10.18653/v1/2024.findings-acl.401.

GAO Y., XIONG Y., GAO X., JIA K., PAN J., BI Y., DAI Y., SUN J., WANG M. & WANG H. (2024b). Retrieval-Augmented Generation for Large Language Models : A Survey. DOI : https ://doi.org/10.48550/arXiv.2312.10997.

GU J., JIANG X., SHI Z., TAN H., ZHAI X., XU C., LI W., SHEN Y., MA S., LIU H., WANG S., ZHANG K., WANG Y., GAO W., NI L. & GUO J. (2025). A Survey on LLM-as-a-Judge. DOI : https ://doi.org/10.48550/arXiv.2411.15594.

GUTIÉRREZ B. J., SHU Y., GU Y., YASUNAGA M. & SU Y. (2024). HippoRAG : Neurobiologically Inspired Long-Term Memory for Large Language Models. In *38th International Conference on Neural Information Processing Systems*.

GUTIÉRREZ B. J., SHU Y., QI W., ZHOU S. & SU Y. (2025). From RAG to Memory : Non-Parametric Continual Learning for Large Language Models. In *42nd International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research.

HE X., TIAN Y., SUN Y., CHAWLA N. V., LAURENT T., LECUN Y., BRESSON X. & HOOI B. (2024). G-Retriever : Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In *38th International Conference on Neural Information Processing Systems*.

HO X., NGUYEN A. D., SUGAWARA S. & AIZAWA A. (2020). Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *28th International Conference on Computational Linguistics*, p. 6609–6625. DOI : 10.18653/V1/2020.COLING-MAIN.580.

HU Y., LEI Z., ZHANG Z., PAN B., LING C. & ZHAO L. (2025). GRAG : Graph Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics : NAACL 2025*, p. 4145–4157. DOI : 10.18653/V1/2025.FINDINGS-NAACL.232.

HUANG L., YU W., MA W., ZHONG W., FENG Z., WANG H., CHEN Q., PENG W., FENG X., QIN B. & LIU T. (2025). A Survey on Hallucination in Large Language Models : Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.*, p. 42 :1–42 :55. DOI : 10.1145/3703155.

IP J. & VONGTHONGSRI K. (2024). DeepEval : The LLM Evaluation Framework.

JIANG J., ZHOU K., ZHAO X., SONG Y., ZHU C., ZHU H. & WEN J. (2025). KG-Agent : An Efficient Autonomous Agent Framework for Complex Reasoning over Knowledge Graph. In *63rd Annual Meeting of the Association for Computational Linguistics*, p. 9505–9523.

KARKI N., PANDEY M., TIWARI S., MIHINDUKULASOORIYA N., GROPPE S. & DOBRIY D. (2026). Agentic AI, Context Engineering and Knowledge Graphs : Current Approaches, Challenges and Opportunities. In *QuaLLM-KG 2026, 1st International Workshop on Quality in Large Language Models and Knowledge Graphs*.

LEDEL K. (2025). The fundamental limitations of AI agent frameworks expose a stark reality gap.

LEMESLE Q., CHEVELU J., LOLIVE D., DELHAY-LORRAIN A. & MARTIN P. (2024). ParaPLUIE - une mesure automatique d'évaluation de la qualité sémantique des systèmes de paraphrases. In *31ème Conférence sur le Traitement Automatique des Langues Naturelles*, p. 605–616.

LEWIS P., PEREZ E., PIKTUS A., PETRONI F., KARPUKHIN V., GOYAL N., KÜTTLER H., LEWIS M., YIH W., ROCKTÄSCHEL T., RIEDEL S. & KIELA D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *34th International Conference on Neural Information Processing Systems, NeurIPS 2020*.

LIN C.-Y. (2004). ROUGE : A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, p. 74–81 : Association for Computational Linguistics.

LIU S., SEMNANI S. J., TRIEDMAN H., XU J., ZHAO I. D. & LAM M. S. (2024). SPINACH : SPARQL-Based Information Navigation for Challenging Real-World Questions. In *Findings of the Association for Computational Linguistics : EMNLP 2024*, p. 15977–16001. DOI : 10.18653/V1/2024.FINDINGS-EMNLP.938.

LUNDBERG S. M. & LEE S. (2017). A Unified Approach to Interpreting Model Predictions. In *31st International Conference on Neural Information Processing Systems*, p. 4768–4777.

LUO H., E H., TANG Z., PENG S., GUO Y., ZHANG W., MA C., DONG G., SONG M., LIN W., ZHU Y. & LUU A. T. (2024a). ChatKBQA : A Generate-then-Retrieve Framework for Knowledge Base Question Answering with Fine-tuned Large Language Models. In *Findings of the Association for Computational Linguistics, ACL 2024*. DOI : 10.18653/V1/2024.FINDINGS-ACL.122.

LUO L., LI Y., HAFFARI G. & PAN S. (2024b). Reasoning on Graphs : Faithful and Interpretable Large Language Model Reasoning. In *12th International Conference on Learning Representations (ICLR)*.

MA C., CHEN Y., WU T., KHAN A. & WANG H. (2025a). Large Language Models Meet Knowledge Graphs for Question Answering : Synthesis and Opportunities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 24578–24597. DOI : 10.18653/V1/2025.EMNLP-MAIN.1249.

MA S., XU C., JIANG X., LI M., QU H., YANG C., MAO J. & GUO J. (2025b). Think-on-Graph 2.0 : Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation. In *13th International Conference on Learning Representations (ICLR)*.

MAHESHWARI H., TENNETI S. & NAKKIRAN A. (2025). CiteFix : Enhancing RAG Accuracy Through Post-Processing Citation Correction. In *63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 310–317. DOI : 10.18653/V1/2025.ACL-INDUSTRY.23.

MUSUMECI E., SURIANI V. & NARDI D. (2025). RoboData : Toward Trustable Question Answering over Ontologies through Metacognitive Agentic Epistemology. In *5th Wikidata Workshop (Wikidata 2025) co-located with 24th International Semantic Web Conference (ISWC 2025)*.

PAHILAJANI A., TRIVEDI D., SHUAI J., YONE K. S., JAIN S. R., PARK N., ROSSI R. A., AHMED N. K., DERNONCOURT F. & WANG Y. (2024). GRS-QA – Graph Reasoning-Structured Question Answering Dataset. DOI : https ://doi.org/10.48550/arXiv.2411.00369.

PAPINENI K., ROUKOS S., WARD T. & ZHU W. (2002). Bleu : a Method for Automatic Evaluation of Machine Translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 311–318. DOI : 10.3115/1073083.1073135.

PENG B., ZHU Y., LIU Y., BO X., SHI H., HONG C., ZHANG Y. & TANG S. (2025). Graph Retrieval-Augmented Generation : A Survey. *ACM Trans. Inf. Syst.* DOI : 10.1145/3777378.

RAGAS (2025). Testset Generation for RAG.

RIBEIRO M. T., SINGH S. & GUESTRIN C. (2016). "Why Should I Trust You ?" : Explaining the Predictions of Any Classifier. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1135–1144. DOI : 10.1145/2939672.2939778.

SAPIDIS I., ZERVOS V., MOUNTANTONAKIS M. & TZITZIKAS Y. (2025). Interactive and Provenance-Aware Search and QA over Documents Using LLMs, RAG and Knowledge Graph Verbalization. In *New Trends in Theory and Practice of Digital Libraries - TPDL 2025 Short Papers and Workshops*, p. 248–257. DOI : 10.1007/978-3-032-06136-2_24.

SARMAH B., MEHTA D., HALL B., RAO R., PATEL S. & PASQUALI S. (2024). HybridRAG : Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction. In *5th ACM International Conference on AI in Finance (ICAIF) 2024*, p. 608–616. DOI : 10.1145/3677052.3698671.

SARTHI P., ABDULLAH S., TULI A., KHANNA S., GOLDIE A. & MANNING C. D. (2024). RAPTOR : Recursive Abstractive Processing for Tree-Organized Retrieval. In *12th International Conference on Learning Representations (ICLR)*.

SHEN Z., DIAO C., VOUGIOUKLIS P., MERITA P., PIRAMANAYAGAM S., CHEN E., GRAUX D., MELO A., LAI R., JIANG Z., LI Z., QI Y., REN Y., TU D. & PAN J. Z. (2025). GeAR : Graph-enhanced Agent for Retrieval-augmented Generation. In *Findings of the Association for Computational Linguistics (ACL) 2025*, p. 12049–12072.

SONG Z., YAN B., LIU Y., FANG M., LI M., YAN R. & CHEN X. (2025). Injecting Domain-Specific Knowledge into Large Language Models : A Comprehensive Survey. In *Findings of the Association for Computational Linguistics : EMNLP 2025*, p. 25297–25311.

SUN J., XU C., TANG L., WANG S., LIN C., GONG Y., NI L. M., SHUM H. & GUO J. (2024a). Think-on-Graph : Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *12th International Conference on Learning Representations (ICLR)*.

SUN J. A., YU H., GOTTI F., MO F., WU Y., HUI Y. & NIE J.-Y. (2025). Search-on-Graph : Iterative Informed Navigation for Large Language Model Reasoning on Knowledge Graphs. DOI : https ://doi.org/10.48550/arXiv.2510.08825.

SUN L., TAO Z., LI Y. & ARAKAWA H. (2024b). ODA : Observation-Driven Agent for integrating LLMs and Knowledge Graphs. In *Findings of the Association for Computational Linguistics (ACL) 2024*, p. 7417–7431. DOI : 10.18653/V1/2024.FINDINGS-ACL.442.

TIAN S., LUO Y., XU T., YUAN C., JIANG H., WEI C. & WANG X. (2024). KG-Adapter : Enabling Knowledge Graph Integration in Large Language Models through Parameter-Efficient Fine-Tuning. In *Findings of the Association for Computational Linguistics (ACL) 2024*, p. 3813–3828. DOI : 10.18653/V1/2024.FINDINGS-ACL.229.

TRIVEDI H., BALASUBRAMANIAN N., KHOT T. & SABHARWAL A. (2022). MuSiQue : Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, p. 539–554. DOI : 10.1162/tacl_a_00475.

WALTER S. & BAST H. (2025). GRASP : Generic Reasoning And SPARQL Generation Across Knowledge Graphs. In *The Semantic Web - ISWC 2025 - 24th International Semantic Web Conference*, p. 271–289. DOI : 10.1007/978-3-032-09527-5_15.

WANG J. & AL. (2025). CRAG-MM : Multi-modal Multi-turn Comprehensive RAG Benchmark. DOI : https ://doi.org/10.48550/arXiv.2510.26160.

WANG P., XU B., ZHANG L., WANG S., DU M., ZHU C. & MAO Z. (2026). WildGraphBench : Benchmarking GraphRAG with Wild-Source Corpora. DOI : https ://doi.org/10.48550/arXiv.2602.02053.

XIANG Z., WU C., ZHANG Q., CHEN S., HONG Z., HUANG X. & SU J. (2026). When to use Graphs in RAG : A Comprehensive Analysis for Graph Retrieval-Augmented Generation. In *14th International Conference on Learning Representations (ICLR)*.

XIAO Y., DONG J., ZHOU C., DONG S., WEN ZHANG Q., YIN D., SUN X. & HUANG X. (2025). GraphRAG-Bench : Challenging Domain-Specific Reasoning for Evaluating Graph Retrieval-Augmented Generation. DOI : https ://doi.org/10.48550/arXiv.2506.02404.

XU Y., HE S., CHEN J., WANG Z., SONG Y., TONG H., LIU G., ZHAO J. & LIU K. (2024). Generate-on-Graph : Treat LLM as both Agent and KG for Incomplete Knowledge Graph Question Answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 18410–18430. DOI : 10.18653/v1/2024.emnlp-main.1023.

YANG Z., QI P., ZHANG S., BENGIO Y., COHEN W. W., SALAKHUTDINOV R. & MANNING C. D. (2018). HotpotQA : A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 2369–2380 : Association for Computational Linguistics. DOI : 10.18653/V1/D18-1259.

YIH W., RICHARDSON M., MEEK C., CHANG M. & SUH J. (2016). The Value of Semantic Parse Labeling for Knowledge Base Question Answering. In *54th Annual Meeting of the Association for Computational Linguistics (ACL)*. DOI : 10.18653/V1/P16-2033.

YU F. J. (2025). What Really Matters to Better GraphRAG Implementation ? — Part 1.

ZHANG Q., CHEN S., BEI Y., YUAN Z., ZHOU H., HONG Z., DONG J., CHEN H., CHANG Y. & HUANG X. (2025a). A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models. DOI : https ://doi.org/10.48550/arXiv.2501.13958.

ZHANG S., DONG L., LI X., ZHANG S., SUN X., WANG S., LI J., HU R., ZHANG T., WU F. & WANG G. (2025b). Instruction Tuning for Large Language Models : A Survey. DOI : https ://doi.org/10.48550/arXiv.2308.10792.

ZHANG T., KISHORE V., WU F., WEINBERGER K. Q. & ARTZI Y. (2020). BERTScore : Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, (ICLR)*.

ZHAO X., BLUM M., GAO F., CHEN Y., YANG B., MARQUEZ-CARPINTERO L., PINA-NAVARRO M., FU Y., MORIKAWA S., IWASAWA Y., MATSUO Y., PARK C. & LI I. (2025). AGENTiGraph : A Multi-Agent Knowledge Graph Framework for Interactive, Domain-Specific LLM Chatbots. In *34th ACM International Conference on Information and Knowledge Management (CIKM)*, p. 6757–6761. DOI : 10.1145/3746252.3761459.

A Annexe : Taxonomie des types de questions avec des exemples

TABLE 1 – Taxonomie des types de questions avec des exemples de combinaisons.

Ouverture	**Raisonnement**	**Exemple**
Questions fermées	Factuel (un saut)	Quelle est la capitale de la France ?
	Multi-sauts	Quelle est la superficie de la capitale de la France ?
	Temporel	Quelle était la capitale de la France en 1800 ?
	Comparaison	Quelle ville est plus peuplée entre Paris et Nice ?
	Analytique (agrégation MAX)	Quelles sont les deux villes les plus peuplées de la France ?
Questions ouvertes	Exploratoire	Que sait-on de Paris ?
	Multi-sauts	Que sait-on de la capitale actuelle de la France ?
	Temporel	Comment Paris a-t-elle évolué au XIXe siècle ?
	Analytique (filtre)	Quelles sont les particularités des villes avec moins de 200 habitants ?
	Comparaison	Comment Paris se compare-t-elle aux autres grandes villes françaises ?

B Annexe : Inventaire des systèmes GraphRAG et de leurs évaluations

TABLE 2 – Synthèse des systèmes GraphRAG et de leurs évaluations

Système	Publication	Description	Code	Datasets	Metriques
AgentiGraph	CIKM 2025	Système multi-agents pour l'interaction avec un KG.	gitHub	TutorQA étendu	Accuracy, F1 score
chatKBQA	ACL 2024	Un LLM finetuné sur Freebase génère des formes logiques candidates.	github	WebQSP, CWQ	F1 score, Hits@1, Accuracy
G-retriever	NeurIPS 2024	Construits des sous-graphes à partir des nœuds et arrêtes pertinents, et génération en exploitant GNN et LLM.	github	GraphQA	Hits@1, Accuracy,
GeAR	ACL 2025	Recherche de passages pertinents en exploitant les triplets d'un KG.	github	MuSiQue, 2WikiMultihopQA, HotpotQA	Recall@k, EM, F1 score
GenTKGQA	ACL 2024	Extraction d'un sous-graphe pertinent en exploitant un LLM pour déduire les relations pertinentes et contraintes temporelles dans la question. Génération à l'aide de GNN et LLM.		CronQuestions, TimeQuestions	Hits@1, Hits@10
GoG	EMNLP 2024	Recherche de chemins pertinents dans un KG avec la possibilité pour le LLM de générer de nouveaux triplets qui n'y figurent pas.	github	CWQ, WebQSP	Hits@1
GRAG	NAACL 2025	Recherche de sous-graphes pertinents et génération en exploitant GNN et LLM.	github	GraphQA (WebQSP and ExplaGraph)	F1 Score, Hit@1, Recall, Accuracy
GraphRAG	arXiv 2025	Recherche d'éléments pertinents dans un KG construit à partir de textes, avec des résumés de communautés.	github	Podcast transcripts, News articles	Exhaustivité de la réponse, diversité, autonomisation, franchise
GRASP	ISWC 2025	Un agent QA génère des requêtes SPARQL sur des KGs RDF.	github	CWQ, WebQSP, QALD-10, QALD-7, SPINACH, WikiWebQuestions	Row-major F1 score, Exact F1 score
HippoRAG	NeurIPS 2024	Recherche de passages pertinents en exploitant un KG construit à partir de corpus textuels et l'algorithme PPR.	github	MuSiQue, 2WikiMultihopQA, HotpotQA	Recall@k, EM, F1 score
HippoRAG-2	ICML 2025	Pallie les limites de HippoRAG en intégrant plus de contexte dans le processus de recherche.	github	NaturalQuestions, PopQA, MuSiQue, 2WikiMultihopQA, HotpotQA, LV-Eval, NarrativeQA	Recall@k, token-based F1 score
HybridRAG	ICAIF 2024	Recherche de passages et triplets pertinents.		Transcriptions des appels de résultats du Nifty 50	Précision et rappel du contexte, fidélité et pertinence de la réponse
KG-Adapter	ACL 2024	Introduit une structure d'adaptateur pour LLM qui encode le KG et permet un raisonnement conjoint avec le LLM.	github	WebQSP, CWQ, OpenBookQA, CommonsenseQA	Accuracy, Hits@1
KG-Agent	ACL 2025	Un LLM agissant comme un agent faisant appels aux outils pour interagir avec le KG.		WebQSP, CWQ, GrailQA, KQA Pro, WebQestion, Natural Questions, TriviaQA	Hits@1, F1 metric, EM
ODA	ACL 2024	Un agent qui combine des cycles d'observation, d'action et de réflexion sur un KG.	github	QALD 10-en, T-REx, Zero-Shot RE, Creak	Hits@1
RAPTOR	ICLR 2024	Recherche de nœuds pertinents dans un arbre de résumés construit à partir de textes.	github	NarrativeQA, QASPER, QuALITY	BLEU, ROUGE-L, METEOR, F1, Accuracy
RoboData	ISWC 2025	Une orchestration agentique des étapes d'exploration, d'évaluation et de mise à jour d'un KG local.	github		
RoG	ICRL 2024	Un LLM finetuné sur Freebase génère des chemins de relations.	github	CWQ, WebQSP	Hits@1, F1
SemanticRAG	TPDL 2025	Recherche vectorielle sur des textes et un KG verbalisé.		FAO PDFs, GRSF KG	
SoG	arXiv 2025	Construit saut après saut des chemins de triplets en explorant le KG.		WebQSP, CWQ, GrailQA, QALD-9, QALD-10, SimpleQA	Hits@1
SPINACH	EMNLP 2024	Un agent imitant l'écriture de requêtes SPARQL sur Wikidata.	github	SPINACH, QALD-7, QALD-9 Plus (en), QALD-10 (en), WikiWebQuestions	Row-major F1 score, EM
Subgraph retriever	ACL 2022	Recherche de plusieurs chemins pertinents, puis fusionnés en un sous-graphe.	github	WebQSP, CWQ	Hits@1, F1 score
ToG	ICLR 2024	Recherche par faisceau de chemins pertinents dans le graphe en utilisant un LLM.	github	CWQ, WebQSP, GrailQA, QALD10-en, SimpleQuestion, WebQuestion, T-REx, Zero-Shot RE	Hits@1
ToG-2	ICRL 2025	Recherche hybride texte et graphe en synergie.	github	WebQSP, AdvHotpotQA, QALD-10-en, FEVER, Creak, Zero-Shot RE, ToG-FinQA	EM, Accuracy
Wikontic	EACL 2026	Recherche itérative, qui décompose les questions en sous-questions à un saut, puis extrait des sous-graphes pour générer des réponses partielles.	github	MuSiQue, HotpotQA	EM, F1 score

C Annexe : Inventaire des jeux de données pour évaluer les systèmes de questions-réponses

TABLE 3 – Inventaire des jeux de données pour évaluer les systèmes de questions-réponses.

Dataset	Année	Taille (nb questions)	Langue	Caractéristiques	Lien
WebQSP	2016	4 737 + 1 073 (annotées partiellement)	Anglais	KGQA sur Freebase ; Questions multi-sauts, analytiques, temporelles, etc. ; Réponses courtes	lien
CWQ	2018	34 689	Anglais	KGQA sur Freebase ; Questions multi-sauts, analytiques, de comparaison, etc. ; Réponses courtes	lien
HotpotQA	2018	112 779	Anglais	Corpus textuel ; Questions à 2 sauts (à pont, comparaison) ; Réponses courtes	lien
2WikiMultihopQA	2020	192 606	Anglais	Corpus textuel ; Questions à 2 sauts (composition, inférence, comparaison, à pont) ; Réponses courtes	lien
GrailQA	2020	64 331	Anglais	KGQA sur Freebase ; Question multi-sauts, analytiques, de comparaison, etc. ; Réponses courtes	lien
BeerQA	2021	163,096	Anglais	Corpus textuel ; Questions de 1-3 sauts ; Réponses courtes	lien
FeTaQA	2022	10 330	Anglais	Table QA ; Réponses textuelles libres avec cellules d'appui	lien
MCWQ	2022	124,187	Anglais, Hébreu, kannada, chinois	KGQA sur Wikidata ; Paires question-SPARQL	lien
MuSiQue	2022	25 000 (MuSiQue-Ans)	Anglais	Corpus textuel ; Questions 2-4 sauts ; Réponses courtes	lien
QALD-10	2022	806	Anglais, chinois, allemand et russe	KGQA sur Wikidata ; Questions multi-sauts, analytiques, de comparaison, etc. ; Réponses courtes	lien
Chat with your Data (cwd)	2023		Anglais	Text2SQL et Text2SPARQL	lien
CRAG	2024	4 409	Anglais	Récupération Web, KG-RAG ; Questions multi-sources ; Réponses courtes et longues	lien
GVLQA	2024	526 320	Anglais	Graph-based Vision-Language QA ; Réponses courtes	lien
GraphQA	2024	2 766 + 100 000 + 4 737	Anglais	Graphe textuel avec les questions relatives ; Réponses courtes	lien
SPINACH	2024	323	Anglais	KGQA sur Wikidata ; paires question-SPARQL	lien
CRAG-MM	2025	6 462 + 1 956	Anglais	Questions multi-modales, multi-tours, multi-sauts, comparaison, analytique ; Réponses longues	lien
GraphRAG-Bench (éducation)	2025	1 018	Anglais	Corpus textuel ; Questions à choix multiples, à sélection multiple, vrai ou faux, à compléter et à réponse libre	lien
GraphRAG-Bench (medical et novel)	2026	2 060 + 2 010	Anglais	Corpus textuel ; Questions fermées et ouvertes incluant synthèse et génération créative ; Réponses longues	lien
Wikidata Query Logs	2026	200 186	Anglais	KGQA sur Wikidata ; paires question-SPARQL	lien
WildGraphBench	2026	1,197	Anglais	Corpus textuel ; Questions multi-sources, de synthèse ; Réponses longues	lien