HydroGraphRAG: exploiter les graphes de connaissances pour une interrogation fiable des données de production hydroélectrique en langage naturel

Nassim BOUDJENAH [1] [*], Fatma-Zohra HANNOU[2], Leila HASSANI[2], Yann REBOURS[3], Marie JUBAULT[2]

[1] Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France
[2] EDF Lab Paris Saclay, SEQUOIA
[3] EDF Hydro - Centre d'Ingénierie Hydraulique (CIH)

Résumé

Bien que les Large Language Models démontrent des performances remarquables en compréhension et génération de texte, ils peinent à interpréter un vocabulaire technique. Dans ce travail, nous présentons HydroGraphRAG, une solution neuro-symbolique qui exploite les graphes de connaissance pour fiabiliser les solutions basées sur les LLMs et les spécialiser aux domaines métiers. HydroGraphRAG implémente une architecture Graph RAG permettant d'interroger, en langage naturel, une base de connaissance industrielle issue de la production hydroélectrique d'EDF. Evalué sur un dataset de prompts métier, la solution améliore significativement la pertinence et la couverture des réponses, notamment face aux acronymes, et aux formulations liées à un contexte technique, tout en réduisant drastiquement le risque d'hallucination observé dans une solution RAG classique.

Mots-clés

IA neurosymbolique, GraphRAG, LLMs, Chatbot, EDF, Production hydroélectrique

Abstract

Although Large Language Models have proven their outstanding language comprehension and generation skills, they are still struggling with unseen technical language. In this work, we introduce HydroGraphRAG, a neuro-symbolic framework that exploits knowledge graphs in order to make LLM-based solutions more reliable. HydroGraphRAG implements a Graph RAG architecture that allows for the querying in natural language of an industrial knowledge base derived from EDF's hydroelectric production. Evaluated on a dataset of technical prompts, the framework significantly improves the relevance and coverage of the answers, most notably for queries dealing with acronyms and technical context, while drastically lowering the risk of hallucination pervasive in classical RAG solutions.

Keywords

Neuro-Symbolic AI, GraphRAG, LLMs, Chatbot, EDF, Hydroelectric production

[*]Travail réalisé lors d'un stage chez EDF Lab Paris Saclay, SEQUOIA.

1 Introduction

Les grands modèles de langage (Large Language Models, LLMs) démontrent des capacités impressionnantes en compréhension et en génération de texte, offrant ainsi des possibilités de simplification d'accès aux connaissances techniques grâce à des systèmes de question/réponse en langage naturel. Pourtant, leur utilisation dans un contexte industriel se heurte à des défis importants : difficultés dans l'interprétation du vocabulaire technique tel que les acronymes, difficultés d'adaptation aux contextes métier, et le risque bien connu des hallucinations. [1, 2].

Pour pallier ces limites, les approches de génération augmentée par la recherche d'information (Retrieval-Augmented Generation, RAG) constituent la famille d'approche la plus utilisée afin d'appuyer la génération sourcée à partir d'un corpus documentaire [3, 4]. A *EDF Hydro*[1], comme dans de nombreuses industries, l'enjeu de simplifier l'accès à la connaissance technique est crucial. Les activités de production et d'ingénierie génèrent et exploitent des volumes de données importants et hétérogènes incluant des référentiels techniques, des bases documentaires, et des données métier décrivant les infrastructures de production, leurs équipements et les opérations associées. Dans ce contexte, un premier chatbot LLM-RAG a été développé pour faciliter l'interrogation de ces connaissances en langage naturel. Cependant, les résultats restent limités. Par exemple, face à la requête suivante :

Q : *Quels sont les équipements des GUs ?*

La Figure 1 illustre les réponses obtenues selon différentes configurations du système[2]. Sans mécanisme de RAG, le LLM interprète incorrectement l'acronyme **GU** comme **Gardes de l'Union** produisant une hallucination. Même l'enrichissement du prompt par une précision du contexte de production hydroélectrique d'EDF n'améliore pas la réponse, ce qui illustre les difficultés du modèle à desambiguiser les acronymes métier. L'usage du RAG permet de

1. https://www.edf.fr/groupe-edf/comprendre/production/hydraulique

2. Pour respecter les contraintes de gouvernance et de sécurité des données, les travaux présentés dans cet article utilisent exclusivement des modèles ouverts déployés localement dans l'environnement de l'entreprise.

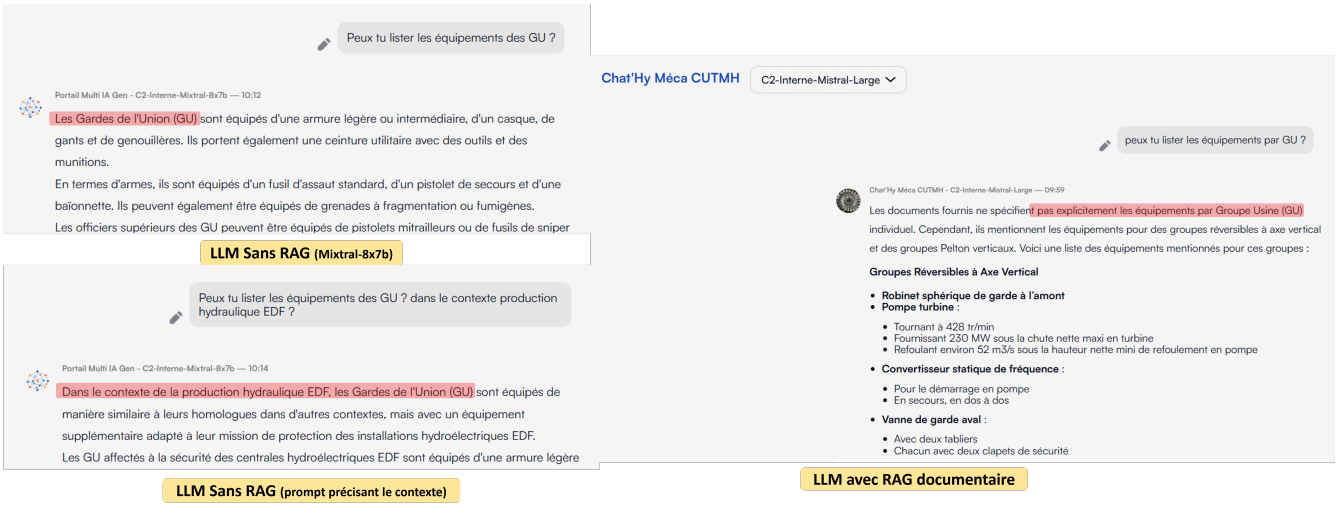

FIGURE 1 – Réponses des chatbots à la requête **Q** : Quels sont les équipements des GUs ? (1) LLM sans RAG (Mixtral 8-7b) produit une hallucination en citant les gardes de l'Union. (2) Le prompt a été enrichi pour préciser le contexte de production hydroélectrique, mais la réponse du LLM demeure erronée. (3) Le LLM avec RAG produit une réponse partielle en échouant dans la récupération de tous les équipements de GU, qu'il interprète comme *Groupes d'Usines* au non *Groupements d'Usines*.

récupérer des documents pertinents, mais l'acronyme n'est toujours pas correctement interprété (Groupes d'Usines au lieu de Groupement d'usines), et les réponses restent partielles, dû à une difficulté à aggréger des informations dispersées sur plusieurs documents.

Cet exemple illustre une limite parmi les nombreuses déjà identifiées dans la littérature : les solutions RAG classiques peinent à traiter efficacement le cas où l'information est présente sur plusieurs documents [5, 6] : gestion imparfaite des contextes longs, incapacité à agréger des informations fragmentées, absence d'exploitation des relations structurelles entre entités...

Face à ces limites, les graphes de connaissances offrent un support particulièrement adapté : ils permettent d'identifier les ressources métier de manière univoque à travers plusieurs sources, exploitent un vocabulaire métier structuré, représentent des liens sémantiques explicites entre entités qui à la différence des textes, vont au delà de la simple proximité textuelle.

Dans cette perspective, les approches **Graph RAG** sont des systèmes neuro-symboliques [7], permettant de combiner les capacités de génération et de raisonnement des LLMs en s'appuyant sur des structures de connaissances explicites pour guider la récupération de l'information. Cependant, ce type d'hybridation soulève plusieurs verrous techniques, car les LLMs utilisent traditionnellement des mécanismes de recherche basés sur des représentation vectorielles, contrairement à la structure relationelle explicite des graphes. Les transformations naïves des graphes en vecteurs induisent souvent une dégradation des liens sémantiques et donc une sous-exploitation dans la récupération de l'information. De plus, l'extraction du contexte à partir du graphe de connaissance requiert l'identification d'un sous graphe pertinent, une tâche complètement différente des métriques de similarité vectorielle utilisés dans les RAGs.

Dans cet article, nous présentons ***HydroGraphRAG***, un système implémentant une approche de type GraphRAG permettant l'interrogation d'une base de connaissance industrielle de la production hydroélectrique. L'approche vise à fiabiliser les réponses en appuyant la génération sur un contexte structuré, riche sémantiquement, issu des graphes de connaissances, afin de préserver les relations entre entités. L'approche se base sur une chaîne de traitement en 3 étapes, qui nécessite les définition des mécanismes d'identification d'entités pertinentes, d'extraction de contexte à partir du graphe et de formulation de la réponse en langage naturel. **HydroGraphRAG** a été évalué grâce à un jeu de données de 50 requêtes métier, avec des niveaux de complexité variables. La suite de cet article est organisée comme suit. La Section 2 présente l'état de l'art sur les approches RAG et Graph RAG. La Section 3 décrit le graphe de connaissance de la production hydroélectrique exploité par notre système. La Section 4 détaille l'architecture de la solution *HydroGraphRAG*. La Section 5 présente l'implémentation et l'évaluation du système, dont les résultat sont discutés dans la Section 6. Enfin, la Section 7 conclut l'article en présentant les perspectives des travaux futurs.

2 Etat de l'art

Les systèmes neuro-symboliques s'imposent aujourd'hui comme une tendance majeure dans le domaine de l'Intelligence Artificielle Explicable (XAI), offrant un pont prometteur entre les capacités génératives des grands modèles de langage (LLMs) et la rigueur factuelle des graphes de connaissances (KG). Cette dynamique a récemment catalysé le développement de nombreuses architectures d'entreprise de type GraphRAG (Graph Retrieval-Augmented Generation), à l'instar des implémentations proposées par Microsoft [8], IBM [9] ou encore LinkedIn [10].

Bien qu'efficaces sur des corpus ouverts, ces approches

généralistes montrent leurs limites dans des domaines hautement spécialisés. D'une part, elles peinent à appréhender les vocabulaires métiers (jargons, acronymes) qui échappent à la sémantique classique des LLMs. D'autre part, la navigation précise au sein de graphes denses et interconnectés engendre d'importantes difficultés computationnelles. Face à ces enjeux méthodologiques, l'état de l'art de l'optimisation GraphRAG se structure autour de deux axes interdépendants : l'extraction d'un sous-graphe pertinent et son injection optimisée dans le contexte du modèle.

Pour l'étape d'extraction de sous-graphes, l'objectif est de récupérer le contexte pertinent à la requête sans capturer d'informations inutiles (bruit). En l'absence de mécanismes intuitifs tels que la similarité vectorielle textuelle utilisée par le RAG classique pour évaluer la pertinence, différentes approches sont employées. Certaines adoptent une approche neuronale, à l'instar de SubgraphRAG [11], qui déploie un classifieur pour sélectionner les triplets pertinents et construire le sous-graphe. Cependant, de telles solutions nécessitent un entraînement préalable et sacrifient l'interprétabilité associée aux approches symboliques. À l'inverse, d'autres méthodes privilégient des approches symboliques, comme KG-RAG [12], qui applique des règles métier pour filtrer l'information, ou encore PathRAG [13], qui utilise un mécanisme de propagation à travers le graphe pour sélectionner le contexte. Toutefois, ces dernières requièrent souvent une connaissance approfondie du domaine métier.

Une fois le contexte identifié, son intégration au modèle varie selon les approches. PathRAG s'appuie sur la pertinence attribuée aux triplets pour les ordonner, afin de présenter l'information la plus cruciale dans les portions du contexte les mieux exploitées par le modèle, évitant ainsi le problème du "Lost in the Middle" [14]. D'autres approches combinent informations textuelles et topologiques pour enrichir le contexte. C'est le cas de KG^2RAG [15], qui fusionne la structure du graphe avec des fragments textuels (chunks) associés aux nœuds, utilisant la topologie pour mieux structurer le texte fourni au modèle.

Par ailleurs, certaines méthodes exploitent le sous-graphe pour stimuler les capacités de raisonnement du LLM. MindMap [16] s'appuie sur un prompting structuré pour forcer le modèle à expliciter son cheminement logique sur le graphe avant de répondre. Poussant ce contrôle plus loin pour éradiquer les hallucinations, GCR (Graph-constrained Reasoning) [17] utilise le graphe comme un masque restrictif (KG-Trie) lors du décodage, contraignant la génération aux séquences validées par les faits du KG.

Bien que performantes, ces approches souffrent de deux limitations majeures. Premièrement, concernant l'extraction de sous-graphes, la majorité des méthodes s'appuie sur des "nœuds de départ" appariés aux entités de la requête. Pour cet appariement, elles utilisent quasi exclusivement la similarité vectorielle entre représentations denses, ce qui s'avère peu adéquat pour gérer des contextes courts et spécifiques (vocabulaire métier, noms propres, acronymes représentés souvent par un seul ou juste quelques tokens) . Deuxièmement, la plupart des approches manipulent les graphes comme des réseaux plats, ignorant une ressource sémantique cruciale : l'ontologie (graphe de schéma). Celle-ci définit formellement la structure des données et présene le potentiel de guider l'extraction avec une grande précision. C'est pour surmonter cette fragilité de l'appariement vectoriel et exploiter pleinement la richesse sémantique formelle de l'ontologie que nous introduisons *HydroGraphRAG*

3 Le Graphe de connaissances

Le système *HydroGraphRAG* s'appuie sur un graphe de connaissances (Knowledge Graph, KG) industriel d'envergure, structuré au format *RDF*[3]. Ce graphe, **GrafHydro**, résulte de l'intégration et de la sémantisation de multiples sources de données internes, ainsi que des outils métier dont les données et les processus sont représentés par des ontologies spécifiques au domaine de l'hydroélectricité.

Périmètre sémantique GrafHydro couvre un large éventail des données résultant de l'activité d'*EDF Hydro*, intégrant des données relatives à l'infrastructure hydraulique (barrages, usines, architecture des sites de production, etc.), à la description technique des équipements, les bases documentaires (rapports techniques, comptes rendus d'intervention), ainsi qu'aux flux de mesures provenant des capteurs. Un ensemble de **19** ontologies couvrant notamment les infrastructures hydrauliques, les équipements industriels, les documents techniques, les flux de mesures et l'organisation des sites de production. Elles définissent les concepts fondamentaux (types de turbines, alternateurs,..), et se complète par l'intégration d'ontologies de référence et de vocabulaires *SKOS, foaf, schema*. Cette représentation modulaire respecte les meilleures pratiques de modélisation offrant un support efficace à la construction et l'alimentation du graphe de connaissance **GrafHydro**[4] à partir de plusieurs systèmes interconnectés.

Volumétrie et caractéristiques structurelles GrafHydro présente une forte densité de relations et une hétérogénéité des types de noeuds (mesures numériques, textes courts, URIs es documents) comme l'illustre la Figure 2. Il contient environ *11 millions d'instances*, avec une connectivité élevée (*~21 millions de relations*) qui reflète la complexité des dépendances opérationnelles en milieu industriel. De plus, les données du graphe proviennent de différentes sources : une combinaison de graphes couvrant des périmètres spécifiques, des numérisations de documents ainsi que des métadonnées issues de bases de gestion documentaire.

Cette hétérogénéité complexifie la structure : au sein du même graphe, plusieurs nœuds peuvent représenter la même entité, mais dans des contextes différents. De plus, on retrouve dans la structure des documents textuels représentés comme des données à part entière avec différents champs associés, tandis que leurs métadonnées sont également intégrées directement au sein du graphe. Ces caractéristiques requierent une optimisation des processus d'interrogation, d'inférence et d'alignement des concepts lors

3. https://www.w3.org/RDF/

4. Nous avons travaillé sur une partie du graphe de connaissance couvrant un périmètre défini par le besoin métier.

FIGURE 2 – Visualisation d'une partie du graphe associée au site "Vieux Près", ses ensembles géo-fonctionnels, ainsi que les documents référencés comme rapports techniques. La relation "isAbout" indique par exemple qu'un document décrit une structure ou activité du site du "VIEUX-PRE".

de l'enrichissement du graphe.
La richesse sémantique de ce graphe, couplée à sa complexité en termes de volumétrie et de connectivité, constitue un défi majeur pour son exploitation automatisée. Dans notre cadre de récupération d'information (retrieval), la capture du contexte pertinent d'une requête est cruciale. Ces caractéristiques imposent la conception d'une solution robuste, capable d'identifier d'abord les nœuds pertinents en tenant compte des ambiguïtés potentielles, puis de naviguer efficacement dans cette structure dense afin d'isoler l'information utile tout en minimisant l'inclusion de bruit. Cette nécessité justifie les choix architecturaux présentés dans la section suivante.

4 Architecture de HydroGraphRAG

La solution proposée, dénommée *HydroGraphRAG*, vise à coupler la puissance de raisonnement des grands modèles de langage (LLM) avec la précision structurelle des graphes de connaissances (KG). Elle y parvient en identifiant et en référençant les concepts (nœuds) du graphe présents dans la requête, puis en récupérant l'information sémantique associée, nécessaire comme contexte pour la réponse. À partir d'une requête textuelle en entrée (prompt classique), le système génère une réponse contextuellement enrichie via une chaîne de traitement structurée en trois étapes clés (Figure 3).

4.1 Approche globale

La chaîne de traitement est conçue pour assurer une correspondance fidèle entre le langage naturel et les données structurées du graphe de connaissance Cette architecture se décompose en trois modules principaux :

— **Extraction d'entités :** Ce premier module identifie les entités clés de la requête utilisateur. D'abord, un LLM extrait les mots-clés pertinents de la requête. Ensuite, pour réaliser l'alignement (*Entity Linking*), ces mots-clés sont directement appariés aux nœuds du graphe de connaissances en s'appuyant sur leur représentation textuelle. Pour ce faire, l'attribut textuel label est exploité ; en son absence, le suffixe de l'IRI du nœud est utilisé comme alternative.
— **Extraction de sous-graphe :** À partir des entités liées, le système isole une portion pertinente du graphe global. L'objectif est de maximiser la pertinence contextuelle tout en minimisant le bruit, afin de ne transmettre au modèle de langage que l'information nécessaire à la résolution de la tâche.
— **Prompting et génération :** Le sous-graphe extrait est transformé dans un format compréhensible par le LLM. Ce contexte structuré guide la génération de la réponse finale, limitant ainsi les risques d'hallucinations en ancrant le modèle dans les faits.

Les sections suivantes détaillent les choix méthodologiques effectués pour chacun de ces modules, illustrés par le prompt suivant.

Exemple 1 (Requête Utilisateur)**.** Considérons la requête suivante (Q) : *« Citez tous les sites associés aux groupements d'exploitation hydraulique. »* . Cette requête servira de fil rouge pour illustrer notre solution.

4.2 Module d'extraction et d'appariement d'entités

Ce module constitue l'interface critique entre la requête en langage naturel et la structure formelle du graphe de connaissances ($\mathcal{G}$). Le processus se décompose en une phase d'extraction sémantique suivie d'une phase d'appariement robuste (*Entity Linking*).

4.2.1 Extraction d'entités par Few-Shot Prompting

L'identification de mots-clés au sein des requêtes utilisateurs susceptibles d'être liés à des nœuds du graphe s'appuie sur une stratégie de *few-shot prompting* via un LLM. Contrairement aux approches de reconnaissance d'entités nommées (*Named Entity Recognition* - NER) classiques, cette méthode présente deux principaux avantage. D'abord, elle évite la dépendance d'un corpus annoté manuellement (requis pour un système de type NER), ressource rarement disponible dans le domaine industriel. Elle offre ainsi une meilleure flexibilité en permettant au modèle de s'adapter au néologismes techniques, et variations de formulations sans nécessiter un réentraînement.

4.2.2 Appariement : limites des approches denses

Pour lier les mots-clés extraits $\mathcal{E} = \{m_1, m_2, \ldots, m_n\}$ aux nœuds du graphe, nous avons initialement évalué l'utilisation d'embeddings denses (type *Sentence-BERT*). Cependant, cette approche montre toutefois ses limites pour notre contexte. Les labels des noeuds sont souvent très courts, réduits à des codes techniques ou acronymes, pour lesquels les représentations vectorielles denses manquent de

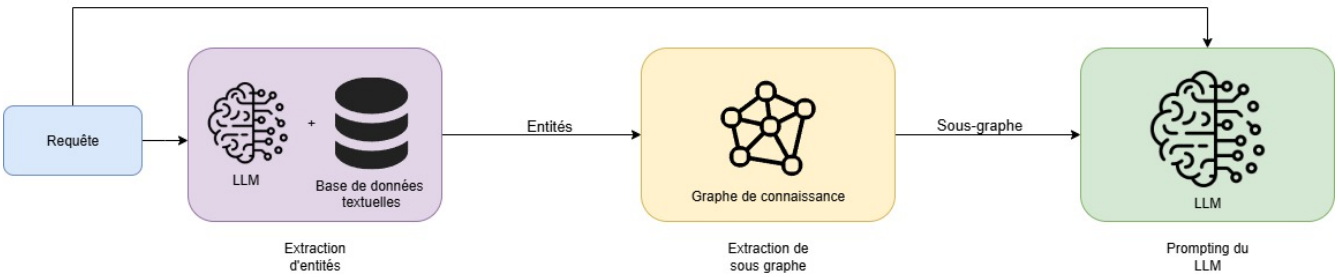

FIGURE 3 – Architecture globale de la solution **HydroGraphRAG** et intéractions entre les modules d'extraction et de génération.

contexte et ne permettent pas un appariement suffisamment fiable. De plus, la compression sémantique effectuée par ces modèles a tendance à atténuer les différences essentielles entre identifiants techniques proches mais distincts.

4.2.3 Approche hybride : BM25 et distance d'édition

Face aux limites des vecteurs denses, nous nous sommes tournés vers des méthodes fréquentielles et lexicales. Le score *BM25* a été retenu pour sa capacité à traiter les correspondances exactes. Néanmoins, BM25 présente une sensibilité accrue aux variations linguistiques (pluriels, conjugaisons) et aux erreurs de saisie, qu'une simple étape de *lemmatisation* ne parvient pas totalement à corriger dans un contexte de labels techniques.
Pour pallier ces faiblesses, nous proposons une fusion de classements via la *Weighted Reciprocal Rank Fusion* (wRRF), intégrant la *distance d'édition (Levenshtein)* pour capturer les similarités morphologiques. Le score final pour un nœud d est :

$$wRRF(d) = w_{\text{lev}} \cdot \frac{1}{k + r_{\text{lev}}(d)} + w_{\text{bm25}} \cdot \frac{1}{k + r_{\text{bm25}}(d)} \quad (1)$$

Dans notre système, un poids prépondérant est accordé à la distance d'édition ($w_{\text{lev}} > w_{\text{bm25}}$). Ce choix technique est motivé par la nécessité de privilégier la proximité syntaxique brute, plus discriminante pour des labels courts et des acronymes techniques que la fréquence statistique des termes. Afin de garantir l'intégrité du processus d'appariement, nous avons fait le choix de ne pas nous limiter au candidat le plus similaire (top-1), mais de conserver les K meilleurs candidats pour chaque entité extraite. Cette stratégie permet de sécuriser la capture des nœuds initiaux corrects, étape cruciale pour l'intégrité de la recherche de connaissances, même si elle introduit potentiellement des nœuds non pertinents (bruit). La gestion et le filtrage de ce bruit sont alors délégués au module (Section 4.3), qui exploitera la structure du graphe pour identifier les sous-graphes les plus cohérents.

Exemple 2 (Extraction et Appariement, suite de l'Exemple 1)**.** À partir de la requête Q, le module d'extraction identifie les mots-clés {"Sites", "groupements d'exploitation hydraulique"}.
ces mots-clés seront appariés par la suite :

— "Sites" est lié à la classe *org1 :Site*.

— "groupements d'exploitation hydraulique" est lié à la classe *org1 :GroupementExploitationHydraulique*.

4.3 Module d'extraction de sous-graphes

Une fois les nœuds initiaux identifiés, l'objectif est d'extraire une structure compacte mais sémantiquement riche, capable de supporter le raisonnement du LLM tout en minimisant le bruit cognitif. Nous dépassons les approches naïves (voisinages k-hop ou plus courts chemins) qui, dans des graphes industriels vastes, capturent soit trop de triplets non pertinents, soit des relations purement structurelles dépourvues de valeur informative.

4.3.1 Enrichissement par l'ontologie

Pour éviter les limites identifiées dans les approches naïves et amorcer une exploration guidée, nous exploitons le schéma ontologique du graphe de connaissances. L'ontologie distingue les *classes* (concepts génériques structurels) des *instances* (entités concrètes portant l'information). Le processus d'enrichissement de l'ensemble de départ, schématisé par la Figure 4, suit une logique de décision stricte pour chaque nœud issu de l'appariement :

— **Nœud de type *Instance* :** Le nœud est conservé tel quel, car il représente directement une entité concrète mentionnée dans la requête.

— **Nœud de type *Classe* :**

  — Si aucune instance de cette classe n'est présente, la classe est *instanciée* : toutes ses instances sont ajoutées à l'ensemble de départ pour garantir la couverture des données pertinentes.

  — Si des instances sont déjà présentes, la classe est écartée. Inclure le nœud de classe risquerait d'activer des chemins purement ontologiques sans valeur informationnelle, introduisant ainsi du bruit.

Cette stratégie présente des avantages pour la qualité :

1. **Couverture exhaustive :** Elle assure l'inclusion de toutes les instances liées à un concept de la requête.
2. **Réduction du bruit :** Elle évite la dérive vers des structures hiérarchiques abstraites.
3. **Désambiguïsation robuste :** Les labels des instances sont souvent courts ou homonymes (sources d'erreurs au module 1), tandis que les classes sont

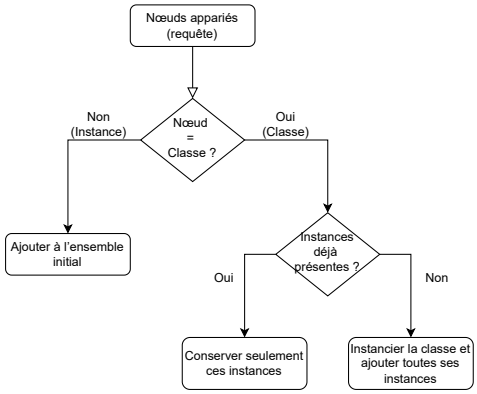

FIGURE 4 – Schématisation du processus d'enrichissement de l'ensemble de départ via l'ontologie.

définies de manière univoque. L'identification d'une classe permet donc de privilégier les candidats liés et d'éliminer les appariements isolés peu fiables.

Cet enrichissement transforme un ensemble de nœuds potentiellement bruyant en une base de départ pertinente et alignée sur le contexte métier.

4.3.2 Calcul de Pertinence par Propagation

L'exploration d'un graphe industriel vaste engendre une explosion de l'espace de recherche. Une approche naïve pour guider cette exploration consisterait à évaluer la pertinence de chaque voisin à l'aide de mesures de similarité textuelle (comme celles utilisées lors de l'appariement). Cependant, réitérer ces calculs à chaque nœud s'avère prohibitif en termes de ressources computationnelles et de latence, ce qui est incompatible avec l'exigence de réactivité d'un système fondé sur les LLM.

Pour pallier cette limite, nous adoptons une méthode inspirée de *PathRAG* [13], fondée sur un calcul de pertinence par propagation. Contrairement à une évaluation locale continue, cette approche distribue globalement un score de confiance. La pertinence est initialisée à 1 pour les nœuds de départ et à 0 sinon, puis propagée selon l'équation 2 :

$$S(v_i) = \sum_{v_j \in N_{in}(v_i)} \alpha \cdot \frac{S(v_j)}{|N_{out}(v_j)|} \qquad (2)$$

Où $N_{in}(v_i)$ représente l'ensemble des voisins entrants, $N_{out}(v_j)$ les voisins sortants, et α un facteur de décroissance. Cette logique favorise les nœuds liés à des points d'intérêt peu denses, souvent plus porteurs de spécificité sémantique, tout en restant significativement moins coûteuse qu'un calcul de similarité répété.

4.3.3 Exploration guidée et élagage

La construction finale du sous-graphe s'appuie sur un algorithme de recherche en profondeur (DFS) optimisé par les scores de pertinence obtenus. Pour garantir un sous-graphe à la fois compact et informatif, deux mécanismes d'élagage (*pruning*) sont intégrés :

1. **Seuil de pertinence :** Tout chemin dont un nœud présente un score $S(v)$ inférieur à un seuil critique est immédiatement abandonné. Cette règle permet d'écarter précocement les branches non pertinentes et de réduire drastiquement le bruit structurel.
2. **Condition de terminaison par recouvrement :** L'exploration s'arrête lorsqu'elle relie deux nœuds de l'ensemble initial. Pour maximiser la valeur ajoutée du contexte, nous privilégions les chemins reliant des instances de classes distinctes, vecteurs des relations inter-domaines essentielles au diagnostic.

Le résultat est un sous-graphe cohérent, dont la taille est maîtrisée pour ne pas surcharger la fenêtre de contexte du modèle de génération (Section 4.4).

Exemple 3 (Extraction du Sous-Graphe, suite de l'Exemple 2)**.** Lors de la phase d'enrichissement (Section 4.3.1), les classes *org1 :Site* et *org1 :groupementsdexploitationhydraulique* sont instanciées pour mieux identifier les nœuds concernés par la requête. Après propagation de la pertinence et élagage, le sous-graphe final consiste en des triplets reliant des instances de groupements d'exploitation hydraulique à des instances de sites, en passant par des instances de la classe groupement d'usines, telles que :

— (MASSIFS DE L'EST,has_gu, JURA LOUE)
— (JURA LOUE,has_site, CRISSEY)

4.4 Module de Génération par LLM

Le dernier module de la pipeline *Hydro GraphRAG* assure la production de la réponse finale en langage naturel. L'enjeu principal réside dans la transformation du sous-graphe extrait (ensemble de triplets $\langle sujet, prdicat, objet \rangle$) en un format textuel optimal pour le LLM, conciliant richesse sémantique et efficacité computationnelle.

4.4.1 Représentation Enrichie des Triplets

Les triplets issus du graphe de connaissances sont initialement identifiés par des IRI (*Internationalized Resource Identifiers*). Cependant, l'utilisation brute d'IRI complets sature rapidement la fenêtre de contexte et nuit à la lisibilité pour le modèle. À l'inverse, l'usage exclusif de suffixes peut s'avérer trop opaque (codes internes).

Pour optimiser l'interprétation du graphe par le LLM, nous introduisons une **représentation enrichie** qui associe à chaque nœud son **label textuel** et son **type ontologique** (ex : *Turbine*, *Capteur*). Cette approche, illustrée par la Figure 5, offre plusieurs avantages :

— **Intelligibilité :** Elle substitue des identifiants techniques par des noms métier explicites, facilitant le raisonnement du modèle.
— **Filtrage du bruit :** Le typage des nœuds aide le LLM à distinguer les entités critiques des nœuds périphériques potentiellement issus du bruit.
— **Précision des réponses :** Elle permet au modèle de manipuler des concepts de haut niveau pour répondre à des requêtes complexes d'agrégation.

Dans notre implémentation, nous privilégions par défaut une combinaison de labels, types et suffixes d'IRI, garantissant ainsi un lien vers la source sans sacrifier la clarté.

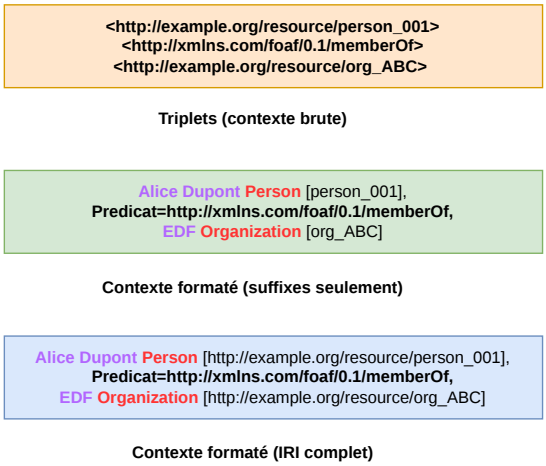

FIGURE 5 – Variantes de représentation des triplets : comparaison entre IRI complets, suffixes simplifiés et notre approche enrichie par labels et types.

4.4.2 Stratégie de Génération Zero-Shot

Contrairement aux approches nécessitant des exemples de démonstration (*few-shot*), nous optons pour une stratégie de **zero-shot prompting**. La richesse sémantique apportée par la transformation des triplets en texte (Section 4.4.1) fournit un contexte suffisamment explicite pour guider le modèle sans nécessiter d'exemples supplémentaires.

Ce choix méthodologique repose sur deux piliers :

1. **Auto-suffisance du contexte textuel :** Le sous-graphe, une fois converti en une suite de triplets textuels enrichis de leurs métadonnées ontologiques, contient toutes les informations nécessaires au raisonnement logique du LLM.
2. **Généralisabilité :** En évitant la dépendance à des paires requête-contexte-réponse prédéfinies, notre système reste portable. Il s'adapte immédiatement à toute mise à jour ou changement du graphe de connaissances hydroélectrique sans nécessiter de nouvelle ingénierie de prompt.

Cette configuration finale garantit que le modèle produit des réponses strictement ancrées dans les faits extraits du graphe, limitant ainsi drastiquement les risques d'hallucinations en contraignant la génération au périmètre sémantique du sous-graphe.

Exemple 4 (Génération par LLM, fin de l'Exemple 3)**.** Strictement guidé par le contextes structurés, le LLM génère la réponse finale : *« Le groupement d'exploitation hydraulique Massifs de l'Est est associée aux sites suivants : SERRE (LA) - RAVILLOLES- … - CRISSEY - …* Agrégeant les sites par groupement d'exploitation hydraulique pour donner une réponse structurée et exhaustive.

5 Implémentation et évaluation

5.1 Implémentation du système

L'implémentation de **HydroGraphRAG** repose sur une architecture matérielle hybride conçue pour répondre aux exigences spécifiques de notre pipeline. Les manipulations intensives du graphe (chargement RDF, extraction de sous-graphes) sont exécutées sur des nœuds CPU à haute capacité mémoire (jusqu'à 500 Go de RAM) à l'aide des bibliothèques `rdflib` et `NetworkX`. L'inférence des Modèles de Langage (LLMs), orchestrée via `LangChain`, est quant à elle déportée sur des serveurs accélérés par des GPU NVIDIA A100 et H100. Afin d'optimiser le compromis entre temps d'inférence et qualité de génération, nous avons adapté la taille des modèles (famille Qwen2.5 [18]) à la complexité intrinsèque de chaque tâche : le modèle **Qwen2.5-14B-Instruct-1M** assure une extraction d'entités rapide ciblée, tandis que le modèle **Qwen2.5-32B-Instruct** est dédié à la génération de la réponse finale, sa plus grande capacité de raisonnement étant indispensable pour interpréter le sous-graphe injecté. Par ailleurs, les hyperparamètres des modules d'extraction de sous-graphes et d'appariement d'entités, ainsi que les paramètres de génération des LLMs, ont été définis de manière empirique. Ce paramétrage vise à maximiser la performance globale du système tout en contenant le temps de calcul.

5.2 Évaluation sur requêtes SPARQL (analyse quantitative)

L'évaluation repose sur un corpus de 50 requêtes en langage naturel, élaborées à partir de cas d'usage réels fournis par des experts métiers et conçues pour couvrir la diversité des besoins d'EDF Hydro. À chaque requête est associée une version SPARQL de référence permettant d'extraire le graphe cible (vérité terrain).

Le jeu de test se décompose en deux catégories distinctes :

- **Requêtes directes (25) :** Portent sur des données bien structurées avec un schéma explicite (ex : caractéristiques d'un équipement).
- **Requêtes complexes (25) :** Introduisent des ambiguïtés de schéma ou nécessitent la navigation à travers des contextes multi-entités et multi-relations.

Cette étape vise à mesurer la précision de l'extraction de connaissances en comparant les sous-graphes extraits (ext) aux graphes de référence (ref) issus des requêtes SPARQL. Nous utilisons les métriques de bruit et de couverture.

Analyse du bruit. Elle quantifie la proportion d'éléments superflus (non pertinents) extraits par le système :

$$\text{Bruit}_N = \frac{|N_{ext} \setminus N_{ref}|}{|N_{ext}|} \quad ; \quad \text{Bruit}_R = \frac{|R_{ext} \setminus R_{ref}|}{|R_{ext}|} \tag{3}$$

Analyse de la couverture. Elle mesure la capacité du module à capturer l'intégralité des informations nécessaires :

$$\text{Couv.}_N = \frac{|N_{ext} \cap N_{ref}|}{|N_{ref}|} \quad ; \quad \text{Couv.}_R = \frac{|R_{ext} \cap R_{ref}|}{|R_{ref}|} \tag{4}$$

5.3 Évaluation qualitative

En complément de l'analyse structurelle, une évaluation manuelle est menée pour juger :

- L'exhaustivité métier des réponses générées.
- La robustesse des modules d'appariement (matching) entre les entités extraites du texte et les nœuds réels du graphe.
- La cohérence du langage naturel produit.

5.4 Baselines de comparaison

Afin de situer les performances de *HydroGraphRAG*, nous confrontons notre approche à deux systèmes de référence (baselines), adaptés à chaque axe d'analyse :

- **Analyse quantitative (Voisinage de profondeur 2) :** Cette baseline conserve notre premier module d'extraction d'entités, mais remplace l'extraction de sous-graphe par une récupération simple du voisinage (nœuds initiaux, voisins directs et voisins de ces derniers). Couramment utilisée dans l'état de l'art, elle permet de mesurer l'apport exact de notre module sur la réduction du bruit et l'amélioration de la couverture.
- **Analyse qualitative (RAG textuel classique) :** Nous comparons notre système à une solution RAG purement textuelle, déjà déployée sur une sous-partie des documents d'EDF Hydro. Cela permet de mettre en évidence les bénéfices de la structuration en graphe face à une recherche vectorielle standard.

6 Analyse des résultats

6.1 Analyse quantitative sur requêtes SPARQL

Nous détaillons ci-après les performances de *HydroGraphRAG* par rapport à la baseline de voisinage direct. L'analyse est séparée entre l'extraction des entités (nœuds) et celle des liens sémantiques (relations).

Requêtes	Méthode	Couv. Nœuds	Bruit Nœuds
Global (50 Q)	*HydroGraphRAG*	0,74	**0,46**
	Baseline	0,26	0,98
Directes (25 Q)	*HydroGraphRAG*	**0,80**	**0,46**
	Baseline	0,46	0,97

TABLE 1 – Comparaison de l'extraction des **nœuds**. L'ensemble **Global (50 Q)** combine 25 **requêtes directes** (schéma explicite) et 25 **requêtes complexes** (ambiguïtés de schéma et navigation multi-entités).

L'analyse conjointe des Tableaux 1 et 2 montre qu'*HydroGraphRAG* cible efficacement les entités et les chemins pertinents (couvertures respectives de 0,74 et 0,71 sur le périmètre global), extrayant ainsi des sous-graphes cohérents. À l'inverse, la couverture de la *baseline* s'effondre (0,26 et 0,23), soulignant son incapacité à naviguer dans un graphe complexe sans l'enrichissement sémantique de l'ontologie.
Le contraste est encore plus marqué sur le bruit : là où la *baseline* noie l'information sous un taux d'erreur extrême

Requêtes	Méthode	Couv. Rel.	Bruit Rel.
Global (50 Q)	*HydroGraphRAG*	0,71	**0,50**
	Baseline	0,23	0,99
Directes (25 Q)	*HydroGraphRAG*	**0,78**	**0,50**
	Baseline	0,42	0,99

TABLE 2 – Comparaison de l'extraction des **relations**. L'ensemble **Global (50 Q)** combine 25 **requêtes directes** (schéma explicite) et 25 **requêtes complexes** (ambiguïtés de schéma et navigation multi-entités).

($\approx 0,98$) par sélection aveugle du voisinage, notre solution réduit ce bruit de moitié. Cela valide l'efficacité de l'algorithme de propagation qui filtre dynamiquement les chemins inutiles.
Enfin, le passage du périmètre clair (25 requêtes) au périmètre global (50 requêtes, incluant des cas ambigus) prouve la robustesse de l'approche. Si l'ajout de complexité fait logiquement baisser la couverture (ex. de 0,80 à 0,74 pour les nœuds), **le bruit reste strictement constant**. Face à l'ambiguïté, le système ainsi restreindre son extraction plutôt que de retourner des sous-graphes erronés.

6.2 Analyses Graphiques et Distributions

Distribution de la performance. L'analyse des distributions permet d'observer la dispersion des résultats sur l'échantillon des 50 requêtes. Les Figures 6 et 7 détaillent la répartition de la couverture et du bruit pour notre solution.

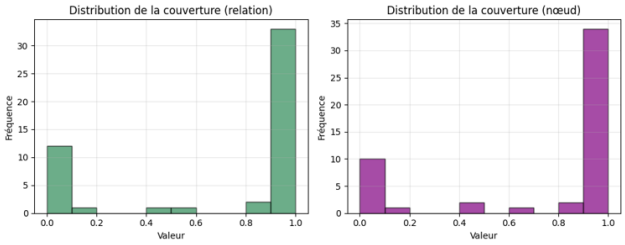

FIGURE 6 – Distribution de la couverture pour l'extraction des **nœuds & relations** avec ***HydroGraphRAG***.

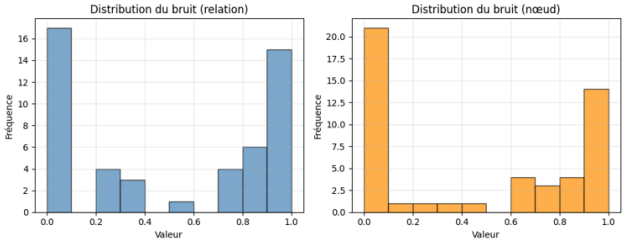

FIGURE 7 – Distribution du bruit pour l'extraction des **nœuds & relations** avec ***HydroGraphRAG***.

Comme l'illustre la Figure 6, la couverture d'*HydroGraphRAG* présente un comportement fortement polarisé. Dans environ 60 % des cas, la couverture est totale, tandis qu'elle est nulle pour 18 % de l'échantillon (généralement sur des requêtes ambiguës ou complexes).

Les situations de couverture partielle, très minoritaires, s'expliquent principalement par la présence de nœuds à forte connectivité sortante. Lors de la propagation, ces nœuds fragmentent les scores de pertinence, ce qui entraîne l'élagage de chemins spécifiques bien que le contexte global ait été correctement identifié.
Concernant le bruit (Figure 7), les résultats montrent que 34 % des requêtes atteignent une précision parfaite (zéro bruit), tandis que la moitié d'entre elles génère plus de 50 % de bruit. Bien que notre méthode surpasse largement la *baseline*, ce bruit résiduel soulève une question cruciale abordée à la Section 6.3 : le LLM final parvient-il à ignorer ce surplus d'information lors de la génération sans altérer la qualité de sa réponse ?

Corrélation avec la taille du graphe. L'analyse de la Figure 8 ne révèle aucune corrélation significative entre les performances d'extraction et la taille du graphe de référence ($|N_{ref}|$). En effet, *HydroGraphRAG* peut échouer (couverture nulle) sur des requêtes de quelques nœuds, tout en réalisant une extraction parfaite (couverture 1, bruit 0) sur des contextes de plusieurs centaines de nœuds.

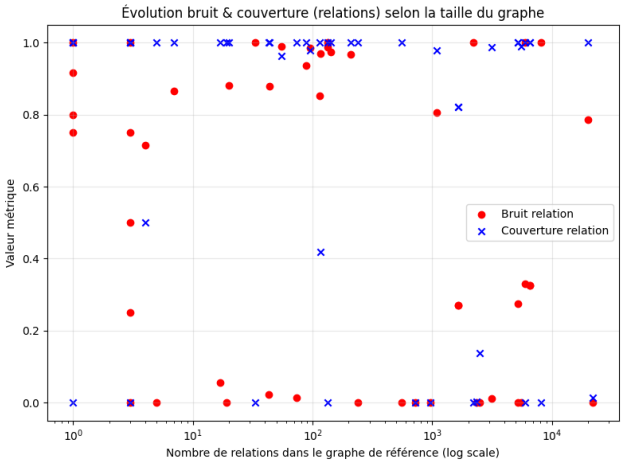

FIGURE 8 – Évolution du bruit et de la couverture en fonction de la taille du graphe cible pour *HydroGraphRAG*.

Ce constat valide la *scalabilité* de l'approche, dont la capacité de récupération ne subit aucune dégradation volumétrique. Il confirme que les limites du système relèvent strictement de la complexité sémantique (présence d'entités ambiguës, chemins topologiques spécifiques à isoler) et non de la taille globale du contexte pertinent à extraire.

6.3 Analyse qualitative et discussion

L'évaluation manuelle permet de juger l'exhaustivité métier et la cohérence des réponses.

Requête A : *« Liste tous les sites associés au groupement d'exploitation hydraulique MASSIFS DE L'EST »*
Analyse : Pour cette requête instance-classe, notre solution fournit un contexte quasi parfait (bruit réduit à un nœud et une relation) et la réponse final du est structurée, agrégeant logiquement les sites via leurs nœuds intermédiaires (ex. groupements d'usines). À l'inverse, le RAG textuel échoue sur ce cas complexe : sa réponse est incomplète, non structurée et redondante, pénalisée par la récupération de multiples *chunks* contenant la même information.

Requête B : *« Citez tous les sites associés aux groupements d'exploitation hydraulique. C'est une requête qui ne concerne que des classes »*
Analyse : De portée globale, cette requête inter-classes englobe toutes leurs instances. Notre solution en extrait un contexte optimal (couverture maximale, bruit minimal) et génère, contrairement au RAG classique, une réponse exhaustive et structurée. Elle réussit notamment à agréger efficacement les informations.

Requête C : *« Est-ce que le site de Couesque contient des instruments du fabricant Endress+Hauser ? »*
Analyse : Cette requête de précision cible un chemin défini entre deux instances. Bien que notre méthode génère un bruit important (53 nœuds récupérés contre 6 attendus), la réponse finale du LLM demeure satisfaisante, démontrant sa capacité à filtrer l'information critique du bruit. À l'inverse, l'échec du RAG classique confirme que de simples documents textuels ne suffisent pas pour capturer des relations structurelles aussi spécifiques.

6.3.1 Analyse des limites

Paire 1 : Impact de l'ambiguïté terminologique

Cas de Succès : *« Liste les relations entre personnes.»*

Cas d'Échec : *« Lister les relations de manager entre personnes »*

Analyse : Cet exemple illustre la sensibilité de la solution à la formulation de la requête et aux ambiguïtés sémantiques. Si le cas de succès présente un bruit nul et une couverture presque parfaite (score de 0,98) même sur un graphe de référence de plus de 2 000 nœuds, le cas d'échec ne porte que sur un sous-ensemble d'informations pourtant bien récupéré dans la requête précédente. L'erreur provient strictement de l'utilisation du terme « manager » qui ne correspond à aucun type de nœud (ce statut s'exprimant par une relation entre personnes dans le graphe), entraînant ainsi un échec de l'appariement.

Paire 2 : Interrogation de documents textuels

Cas de Succès : *« liste les Unites de Production et les Communication CUTMH associées »*

Cas d'Échec : *« Récupérer les communications CUTMH où M. VOLLE est mentionné. »*

Analyse : Cet exemple illustre un cas hybride où les nœuds combinent métadonnées structurelles et texte brut (communications). Si la solution excelle (zéro bruit, couverture parfaite) sur les requêtes ciblant la structure du graphe, elle atteint logiquement ses limites (échec du retrival) sur le contenu textuel non structuré. Ce constat confirme la nécessité d'évoluer vers une hybridation avec un RAG classique pour interroger conjointement graphe et texte.

7 Conclusion et perspectives

Le système *HydroGraphRAG* présenté dans ce travail constitue une implémentation d'une architecture neurosym-

bolique destinée à fiabiliser l'interrogation en langage naturel de graphes de connaissances industriels. Le système combine les capacités génératives des LLMs avec l'exploitation explicite de structures sémantiques issues d'un graphe RDF et de son ontologie afin de guider l'identification des entités, l'extraction du contexte pertinent et la génération de la réponse. Contrairement à une grande partie des travaux de l'état de l'art, principalement évalués sur des benchmarks synthétiques ou des graphes simplifiés, notre approche a été conçue et expérimentée dans un contexte industriel réel, caractérisé par une forte hétérogénéité des données, des graphes hyper-connectés, un vocabulaire métier spécialisé et l'usage massif des acronymes techniques. L'évaluation menée sur 50 requêtes métier issues de besoins réels, exprimés par EDF Hydro, montre que HydroGraphRag permet d'atteindre jusqu'a 78% de couverture tout en réduisant de moitié le bruit lors de l'étape de retrieval par rapport à une approche basée sur le voisinage naïf. L'analyse qualitative met également en évidence une amélioration de l'exhaustivité et de la cohérence des réponses générées comparativement à un système RAG documentaire classique, notamment sur les requêtes nécessitant l'agrégation d'informations dispersées.

Etant donné la variété des métiers de l'entreprise, la conception du système a porté une attention particulière au caractère générique et au potentiel d'extension. Les mécanismes d'appariement, de propagation et d'extraction reposent sur les propriétés structurelles du graphe (URIs, relations, voisinages, typage ontologique) plutôt que sur des heuristiques spécifiques au domaine hydroélectrique. Cette indépendance vis-à-vis des schémas métier particuliers ouvre des perspectives de généralisation vers d'autres graphes de connaissances industriels ou techniques.

Enfin, plusieurs perspectives se dégagent de ces travaux, notamment l'amélioration du traitement des requêtes ambiguës ou impliquant des données textuelles peu structurées. L'intégration d'approches hybrides combinant GraphRAG et RAG documentaire représente une direction prometteuse afin d'exploiter conjointement les relations explicites du graphe et le contenu sémantique des documents textuels

8 Usage de générative

Un LLM a été utilisé dans certaines parties pour synthètiser le texte original des auteurs (limite de pages).

Références

[1] Andrew KERNYCKY et al. "Evaluating the Performance of LLMs on Technical Language Processing Tasks". In : *HCI International 2024 Posters*. Springer Nature Switzerland, 2024, p. 75-85. ISBN : 9783031621109.

[2] Christian HEROLD et al. *Vocabulary Customization for Efficient Domain-Specific LLM Deployment*. 2025. arXiv : 2509.26124 [cs.CL].

[3] Sanat SHARMA et al. *Retrieval Augmented Generation for Domain-specific Question Answering*. 2024. arXiv : 2404.14760 [cs.CL].

[4] Tianjun ZHANG et al. *RAFT : Adapting Language Model to Domain Specific RAG*. 2024. arXiv : 2403.10131 [cs.CL].

[5] Grégoire MARTINON et al. *Towards a rigorous evaluation of RAG systems : the challenge of due diligence*. 2025. arXiv : 2507.21753 [cs.AI].

[6] Zehan QI et al. *Long^{2}RAG : Evaluating Long-Context & Long-Form Retrieval-Augmented Generation with Key Point Recall*. 2025. arXiv : 2410.23000 [cs.CL].

[7] Shirui PAN et al. "Unifying large language models and knowledge graphs : A roadmap". In : *IEEE Transactions on Knowledge and Data Engineering* 36.7 (2024), p. 3580-3599.

[8] Darren EDGE et al. "From local to global : A graph rag approach to query-focused summarization". In : *arXiv preprint arXiv :2404.16130* (2024).

[9] Haoyu HAN et al. "Rag vs. graphrag : A systematic evaluation and key insights". In : *arXiv preprint arXiv :2502.11371* (2025).

[10] Zhentao XU et al. "Retrieval-augmented generation with knowledge graphs for customer service question answering". In : *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*. 2024.

[11] Mufei LI, Siqi MIAO et Pan LI. "Simple is effective : The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation". In : *arXiv preprint arXiv :2410.20724* (2024).

[12] Karthik SOMAN et al. "Biomedical knowledge graph-optimized prompt generation for large language models". In : *Bioinformatics* 40.9 (sept. 2024), btae560. ISSN : 1367-4811.

[13] Boyu CHEN et al. "Pathrag : Pruning graph-based retrieval augmented generation with relational paths". In : *arXiv preprint arXiv :2502.14902* (2025).

[14] Nelson F LIU et al. "Lost in the middle : How language models use long contexts". In : *arXiv preprint arXiv :2307.03172* (2023).

[15] Xiangrong ZHU et al. "Knowledge graph-guided retrieval augmented generation". In : *arXiv preprint arXiv :2502.06864* (2025).

[16] Yilin WEN, Zifeng WANG et Jimeng SUN. "Mindmap : Knowledge graph prompting sparks graph of thoughts in large language models". In : *arXiv preprint arXiv :2308.09729* (2023).

[17] Linhao LUO et al. "Graph-constrained reasoning : Faithful reasoning on knowledge graphs with large language models". In : *arXiv preprint arXiv :2410.13080* (2024).

[18] QWEN et al. *Qwen2.5 Technical Report*. 2025. arXiv : 2412.15115 [cs.CL].