Optimisation du RAG sur hypergraphes

Vers une meilleure extraction de faits et récupération de chunks

Houda Khrouf, Pedro Fillastre, Sebastiao Correia

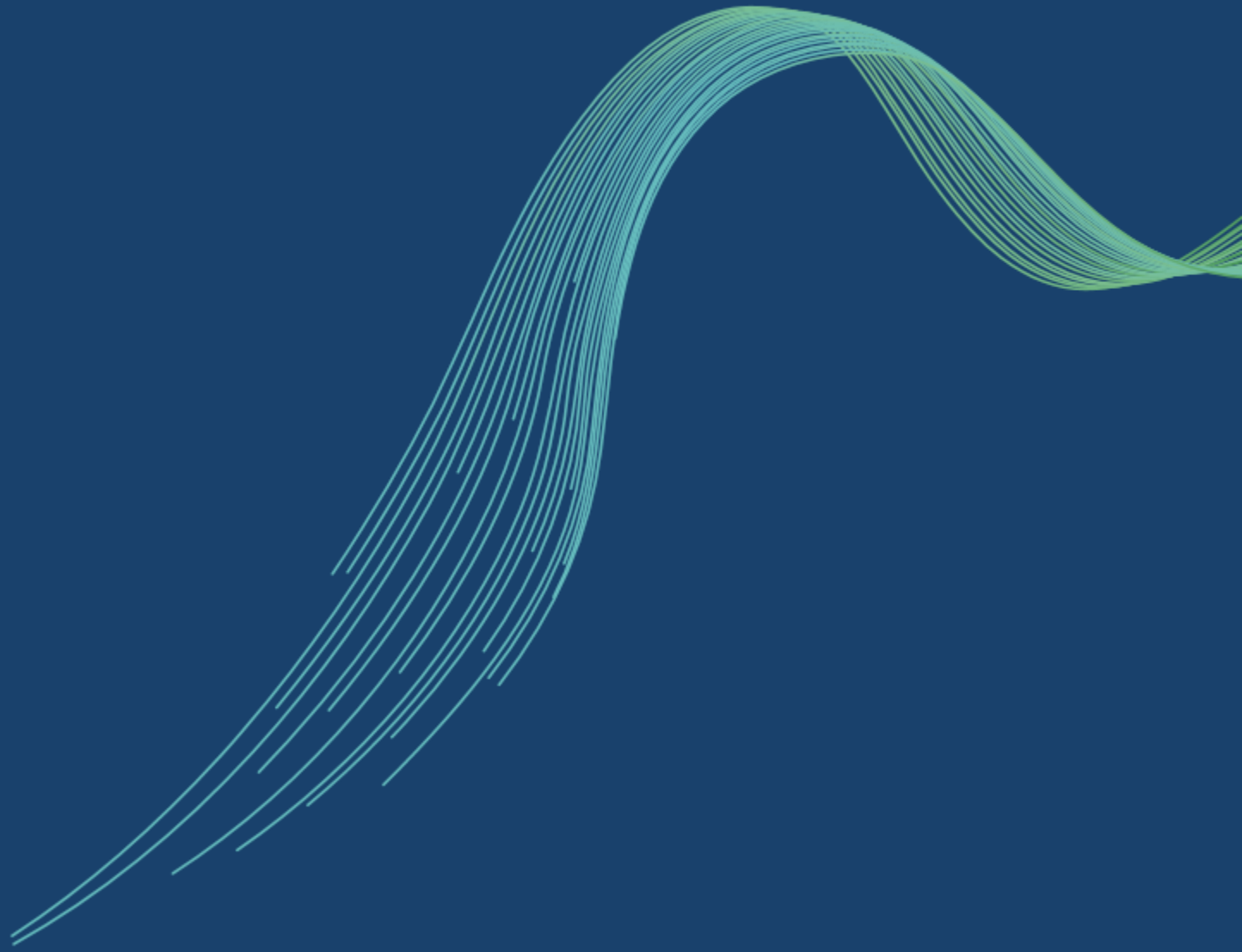
Applied Research, Qlik

Plate-Forme Intelligence Artificielle (PFIA) 2026 - Arras

01

Contexte

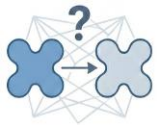
Limites de RAG et Graph RAG



Limites du RAG standard

Recherche de données non efficace

Corpus de texte → Découpage en chunks → Index vectoriel → Requête utilisateur → Réponse générée



Similarité superficielle

La recherche dense ne capture que la similarité sémantique locale — sans raisonnement relationnel entre les entités .

Perte de contexte

Le découpage en chunks casse les relations inter-passages ; les questions transversales restent sans réponse complète.

Requêtes globales impossibles

"Quels sont les thèmes principaux ?" — la recherche vectorielle ne peut pas synthétiser un corpus entier.

Hallucinations amplifiées

Des passages redondants ou hors-sujet récupérés par similarité polluent le contexte et favorisent les hallucinations.

Graph RAG

Du RAG standard vers RAG à base de graphes binaires

Apports du Graph RAG

Compréhension contextuelle

Les relations entre nœuds capturent le sens au-delà de la similarité sémantique superficielle.

Raisonnement multi-sauts

La structure de graphe permet de naviguer entre des entités distantes pour répondre à des questions complexes ou globales.

Désambiguïsation d'entités

Le graphe relie explicitement les entités, réduisant les confusions entre noms similaires ou coréférences.

Limites persistantes

Fragmentation du contexte N-aire

Un graphe binaire ne peut connecter que deux nœuds par arête — les faits n-aires impliquant plus de 2 entités sont fragmentés.

Sensibilité aux erreurs d'extraction LLM

Une négation, une hypothèse ou une nuance temporelle peut être capturée comme un fait affirmé.

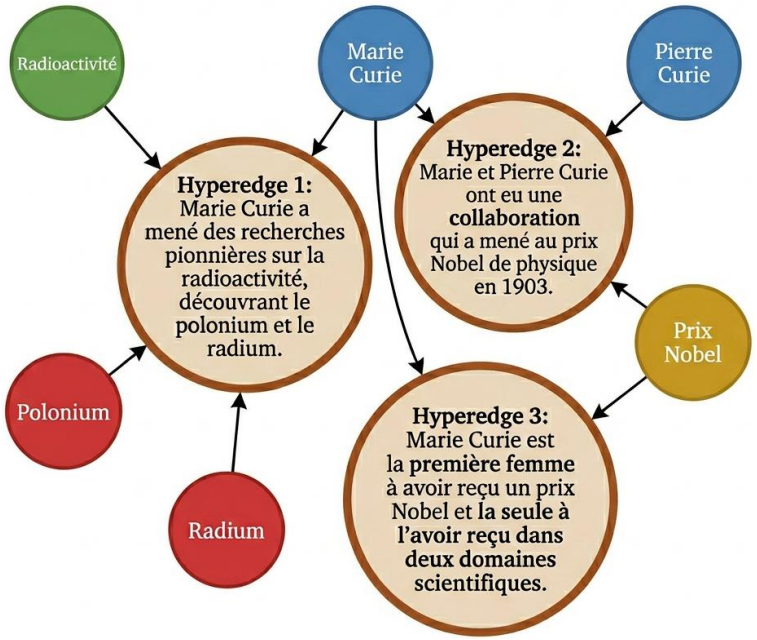
Manque de contexte relationnel

La décomposition en triplets binaires ne garantit pas la préservation des attributs contextuels qualifiant une relation, tels que le caractère « premier » d'une collaboration.

HyperGraph RAG

Emergence des approches à bases d'Hypergraphes

Qu'est-ce qu'un Hypergraphe de connaissance ?



Une hyperarête = un fait n-aire connectant N entités

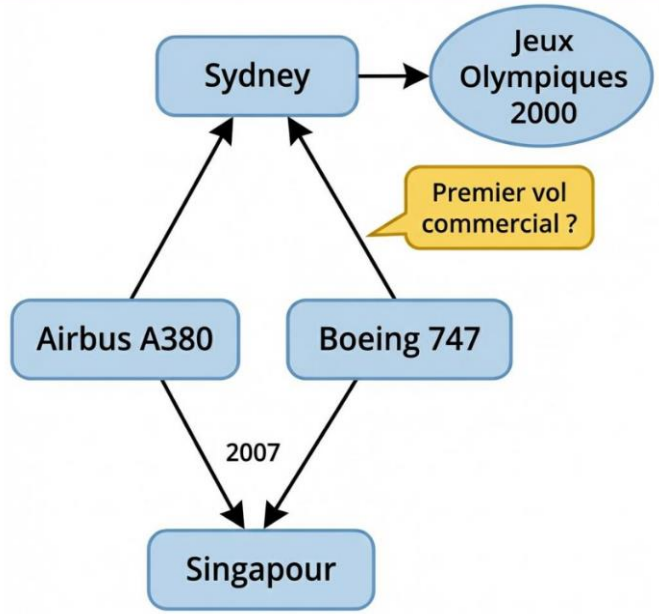
Avantage clé : un seul élément structurel préserve l'intégralité du fait — temporalité, modalité et relations entre toutes les entités sont maintenues sans perte d'information.

Hypergraphe vs Graphe Binaire

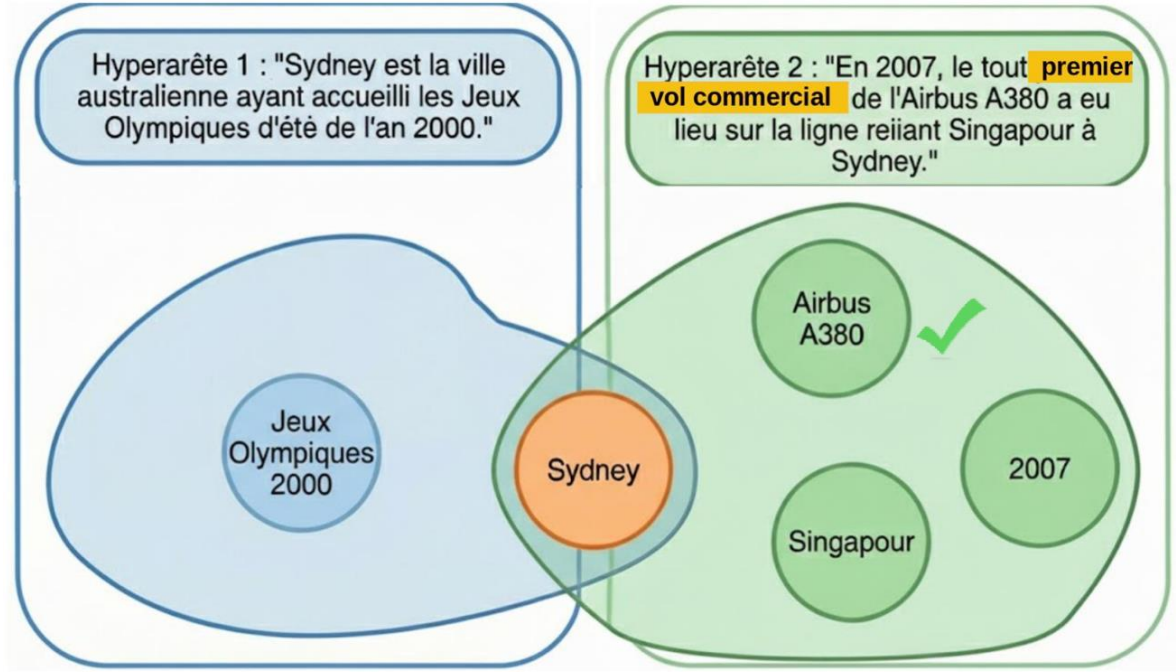
Example de requête

Question : Quel modèle d'avion a été utilisé en 2007 lors du premier vol commercial reliant Singapour à la ville australienne ayant accueilli les Jeux Olympiques 2000 ?

Graphe Binaire



Hypergraphe



- Impossible de distinguer Airbus A380 vs Boeing 747
- Fait n-aire préservé

HyperGraphRAG

L'approche de référence

HyperGraphRAG *Luo et al., 2025 — Première adaptation du paradigme RAG aux hypergraphes de connaissances*

① Phase d'Extraction

Construction de l'hypergraphe à partir d'une extraction few-shot suivie par une vérification itérative pour identifier et compléter les faits manquant

② Phase de Récupération

Recherche vectorielle sur entités et hyperarêtes, suivie d'une expansion topologique à 1 saut et une récupération standard des chunks

D'autres travaux émergents fin 2025:

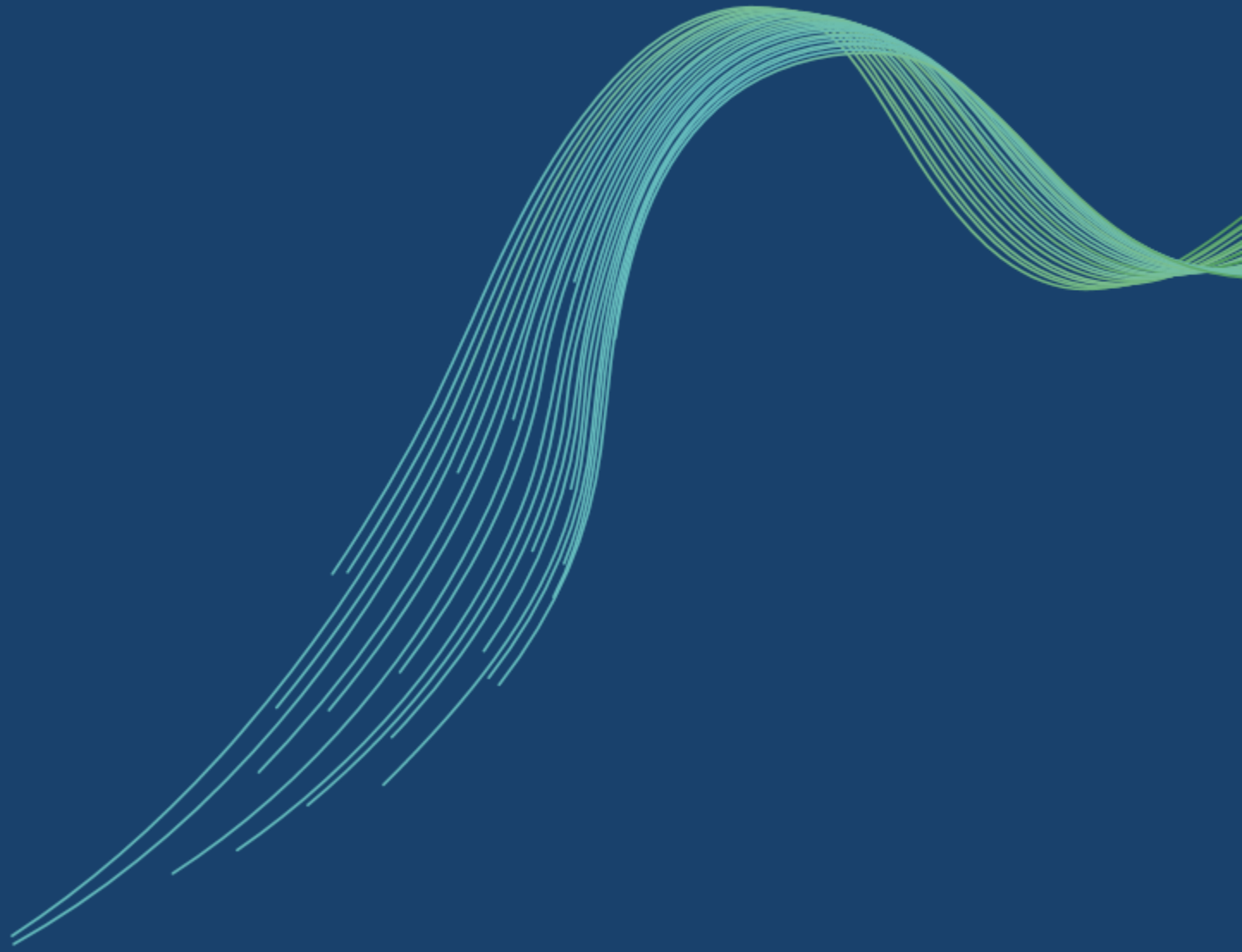
- **PRoH** (Zai et al., 2025): Décomposition de la requête et récupération itérative pour un raisonnement adaptatif
- **Hyper-RAG** (Feng et al., 2025): Exraction zero-shot et une récupération similaire à celle de HyperGraphRAG

Questions ouvertes (motivations directes de ce travail) **:**

- Fiabilité de l'extraction à grande échelle
- Exploitation de la topologie globale lors de la récupération

02

Motivation & Contributions



Limites d'HyperGraphRAG

Deux lacunes majeures

① Extraction instable par LLM

Qualité de l'hypergraphe

- Omissions d'entités → hyperarêtes isolés (taux d'isolation allant jusqu'à 62%)
- Hyperarêtes courts dénuées de sens ("Que suis-je ?")
- Résolution de coréférences défaillante → discontinuité du graphe

Impact: Graphe fragmenté
→ raisonnement multi-sauts dégradé

Limites d'HyperGraphRAG

Deux lacunes majeures

① Extraction instable par LLM

Qualité de l'hypergraphe

- Omissions d'entités → hyperarêtes isolés (taux d'isolation allant jusqu'à 62%)
- Hyperarêtes courts dénuées de sens ("Que suis-je ?")
- Résolution de coréférences défaillante → discontinuité du graphe

Impact: Graphe fragmenté → raisonnement multi-sauts dégradé

② Récupération sous-optimale

Sous-exploitation de la topologie de graphe

- Limitée à une recherche sémantique + expansion de voisinage à 1 saut
- Problème d'horizon : le contexte pertinent au-delà du voisinage immédiat est manqué
- Des chunks topologiquement déconnectés de la requête
- Bruit contextuel et signal utile dilué → plus d'hallucinations

Impact: Rappel contextuel insuffisant

Contribution 1 — EXT++ : extraction par auto-cohérence

Self-consistency prompting (Chen et al. 2024)

- **Inspiration : Universal Self-Consistency (USC)**
  - **Mécanisme :** Multiples chemins de génération au lieu d'un seul décodage (greedy decoding).
  - **Avantage :** Consensus robuste sans un évaluateur externe.
- **Instructions dans le prompt EXT⁺⁺ :**
  - 3 itérations d'extraction indépendantes en un seul appel LLM
  - Agrégation interne des extractions par union dédupliqué et résolution des contradictions
  - Résolution immédiate des coréférences
  - Structuration topologique stricte (entités listées immédiatement après chaque hyperarête associée)

🎯 Réduction des hallucinations

Filtre les relations erronées ou spéculatives qui ne font pas consensus entre les passes.

⚓ Meilleure stabilisation

Compense les baisses d'attention du LLM : capture les informations "enfouies" malgré la densité du texte.

🌐 Maximiser la complétude

Fusionne les découvertes et produit un hypergraphe final plus riche et interconnecté.

Contribution 2 — PPR sur hypergraphe

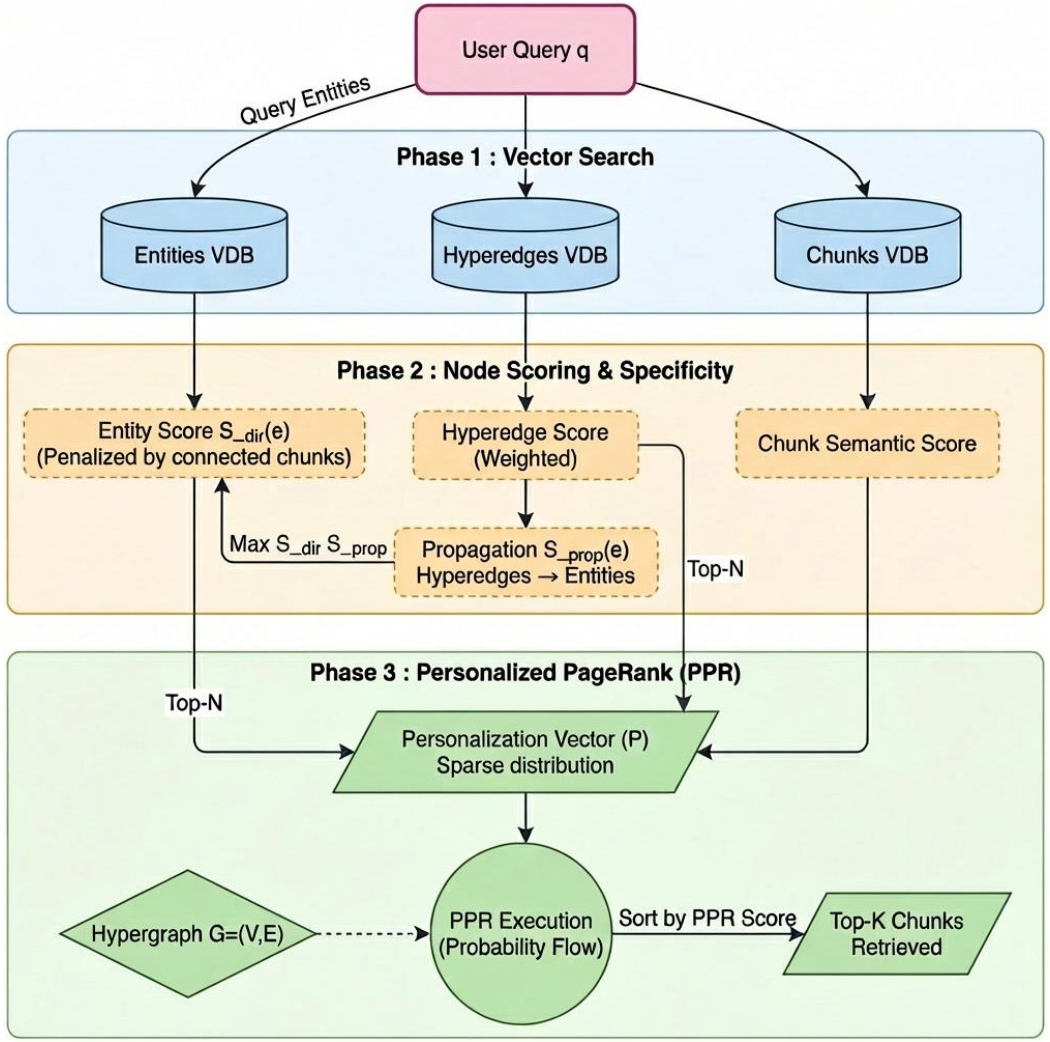
Récupération Optimisée via PageRank Personnalisé (PPR)

Objectif :

Identifier les chunks non seulement sémantiquement pertinents, mais aussi fortement connectés au contenu de la requête.

Pipeline en 3 étapes :

1) **Recherche sémantique** : Top-k entités, hyperarêtes et chunks
2) **Notation des nœuds** : Calibration des scores pour initialiser le vecteur de personnalisation PPR
3) **PageRank Personnalisé** : Propagation probabiliste de pertinence sur l'hypergraphe



Contribution 2 — PPR sur hypergraphe

Notation des nœuds et propagation PPR

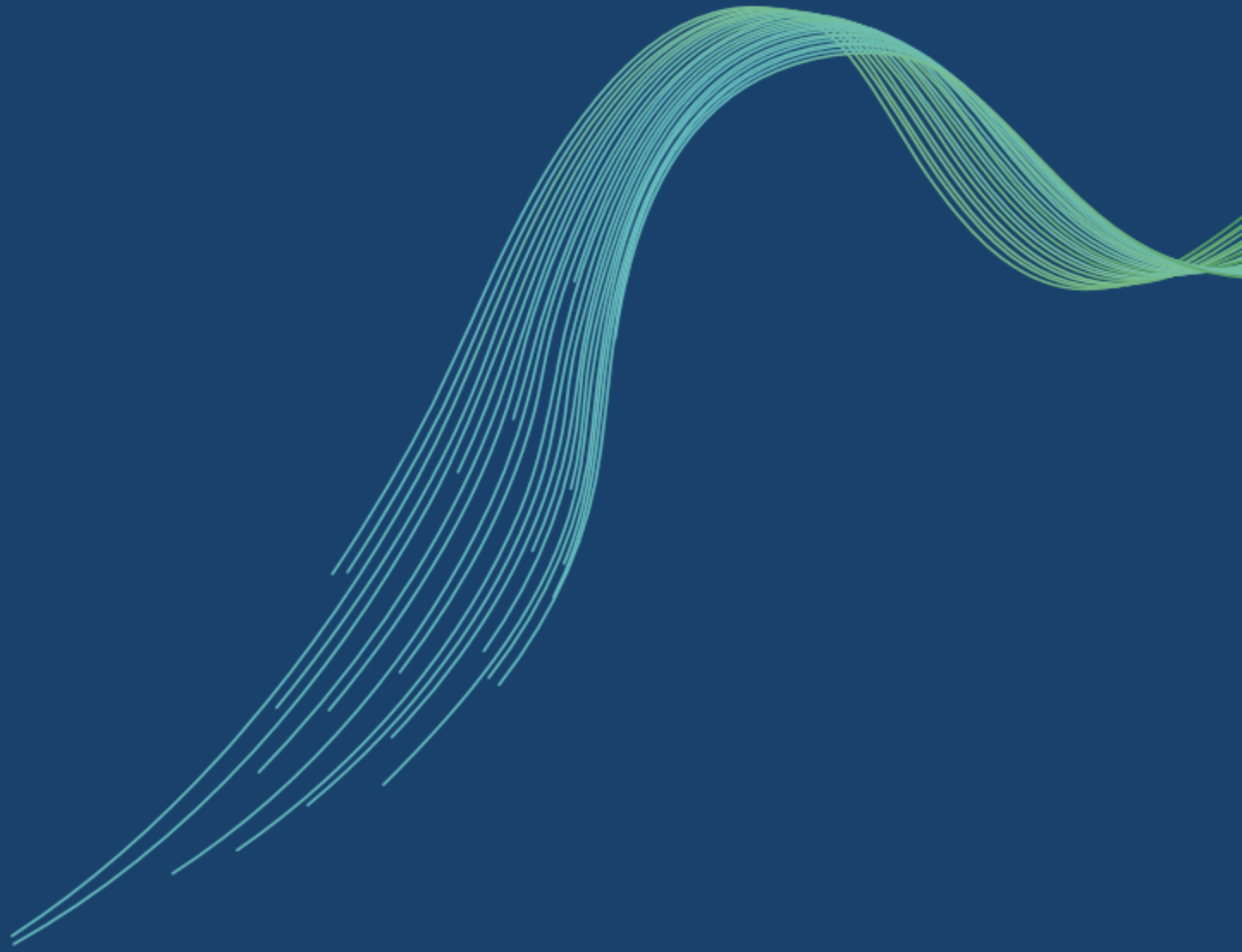
1. **Initialisation:** Chaque nœud (entité, hyperarête, chunk) reçoit un **score local** basé sur sa similarité vectorielle.

2. **Calibration des scores d'entités:**
   - **Couverture du signal - calcul du score propagé :**
     - Une entité reçoit la moyenne des scores des hyperarêtes incidentes.
     - Agrégation Noisy-OR amplifie le score des entités situées au carrefour de plusieurs hyperarêtes pertinentes.
     - Le score de l'entité devient le MAX entre le score local et le score propagé.
   - **Pénalité de généricité :** Le score d'une entité est pénalisé par son degré de connectivité aux chunks.

3. **Exécution du PPR**
   - **Assemblage :** Normalisation Min-Max des scores pour former le vecteur de personnalisation PPR.
   - **Résultat :** Propagation sur le graphe pour extraire les top-K chunks triés par score PPR décroissant.

03

Evaluation



Cadre d'évaluation

Fiction

Source : UltraDomain

13 docs (≤560k tokens)
84 questions dont 72 multi-sauts
Focus: Personnages, intrigue, littérature

CS (computer Science)

Source : UltraDomain

3 manuels universitaires denses
398 questions (175 multi-sauts)
Focus : Software Architecture, algorithmes, ML

MAUD

Source : LegalBench-RAG

21 accords M&A
74 questions factuelles (1 saut)
Focus : Clauses juridiques & financières

Métriques

C-Recall

Rappel Contextuel (0-1)

Proportion des informations attendues présentes dans le contexte récupéré.

A-Correct *LLM-as-a-judge (GPT 4.1-mini)*

Exactitude de la réponse (0-10)

Alignement factuel avec la référence

A-Complete *LLM-as-a-judge (GPT 4.1-mini)*

Complétude de la réponse (0-10)

Couverture de tous les aspects mentionnés dans la référence

Modèle LLM: GPT-4o-mini (extraction de graphe et génération de réponse)

Performance

Comparaison avec les baselines

+51%
Rappel contextuel
vs HyperGraphRAG (Fiction)

+5%
Exactitude
vs HyperGraphRAG (Maud)

+10%
Complétude
vs HyperGraphRAG (Fiction & Maud)

	FICTION			CS			MAUD		
Method	**C-Recall**	**A-Correct**	**A-Complete**	**C-Recall**	**A-Correct**	**A-Complete**	**C-Recall**	**A-Correct**	**A-Complete**
Standard RAG	0.40	7.11	5.01	0.60	8.56	6.34	0.10	3.38	2.22
Contextual Chunking (by Anthropic)	0.45	8.00	5.62	0.55	8.39	6.21	**0.36**	7.62	4.76
Graph Transformer (Neo4j) - binary Graph	0.40	6.96	4.95	0.60	8.65	6.45	0.12	7.73	5.04
HippoRAG2 - binary Graph	0.44	8.51	6.19	0.52	8.51	6.77	0.35	**8.69**	4.12
HyperGraphRAG	0.39	9.54	7.17	0.67	9.37	8.27	0.13	8.28	5.24
HyperGraphRAG + PPR	0.55	9.64	7.82	**0.73**	9.43	**8.28**	0.19	8.48	5.39
HyperGraphRAG + PPR + EXT++	**0.59**	**9.75**	**7.94**	0.71	**9.45**	**8.28**	0.22	8.68	**5.80**

Performance

Comparaison des modèles LLM pour génération de réponse

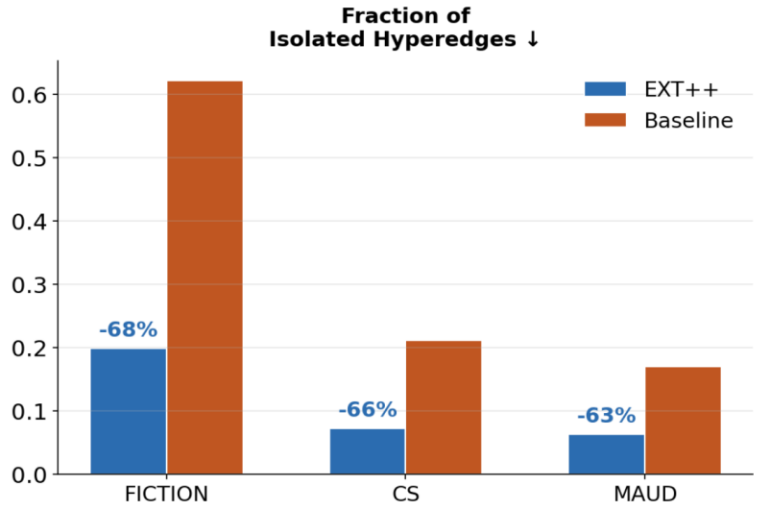
- L'exactitude reste élevée pour tous les modèles (8.68–9.96/10)
- La complétude différencie les modèles puissants (+16-35% Sonnet 4.5 vs GPT-4o mini)

Impact de l'extraction optimisée EXT++

Qualité de l'hypergraphe

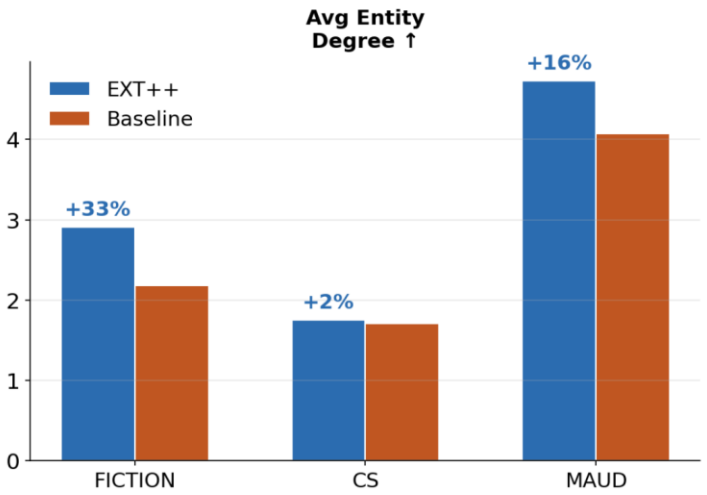
Meilleure connectivité et densité de graphe facilitant le raisonnement multi-sauts

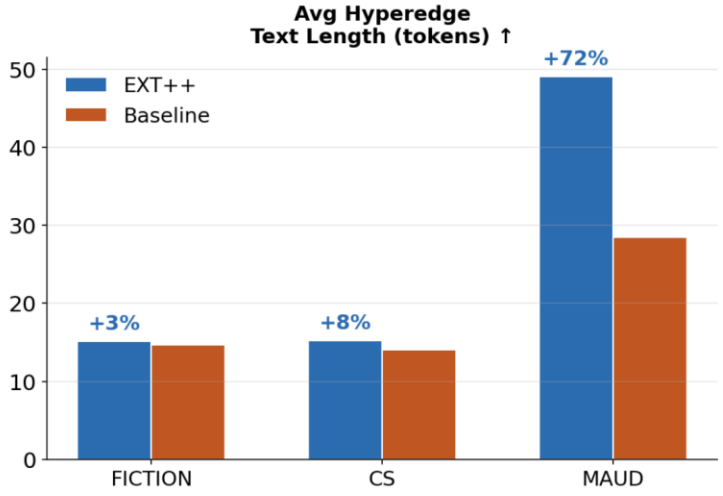
- Réduction importante des hyperarêtes isolées (de –63% à -68%)
- Augmentation du degré de connectivité des entités (+2 à +33%)



Hyperarêtes plus informatives

- Hausse de la moyenne de tokens par hyperarête +3% à +72%
- +23% d'entités par hyperarête connectée dans le dataset Maud





Impact de l'extraction optimisée EXT++

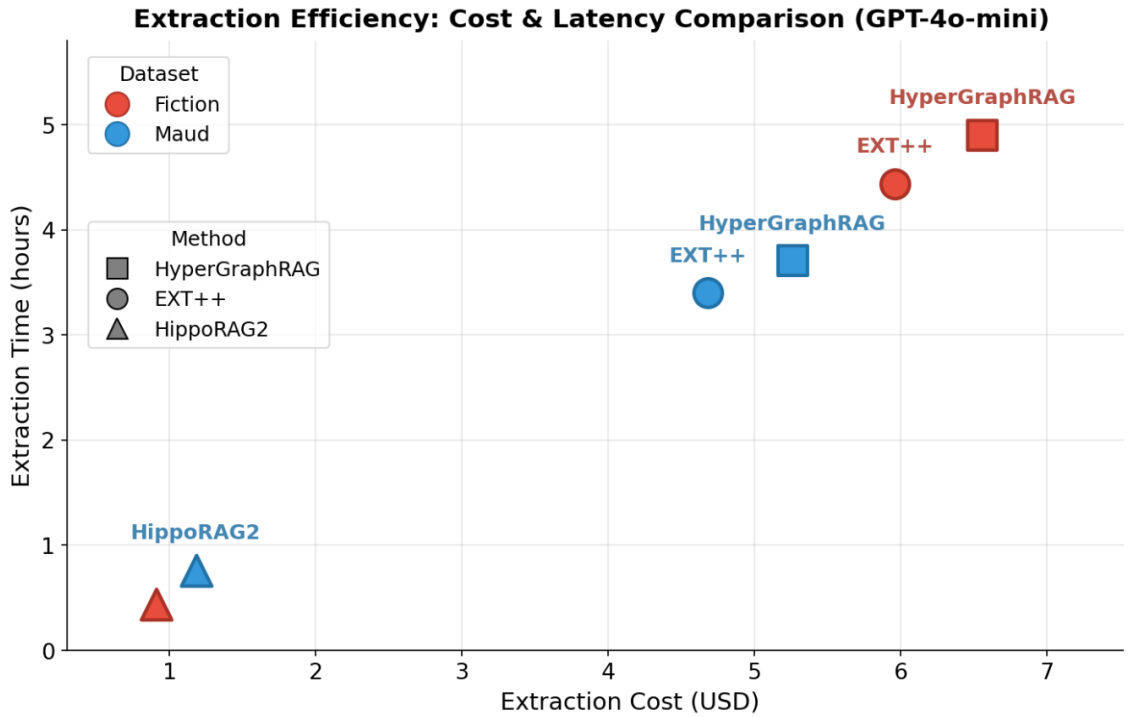
Coût et Latence

L'extraction EXT++ avec 3 passes internes s'avère **plus rapide et moins coûteuse** que l'extraction baseline de HyperGraphRAG.

Deux facteurs clés :

- **Exploitation du Prompt Caching :** Le prompt étendu bénéficie du cache des API LLM (+20 à +24% de tokens en cache)
  ➡️ Pas d'impact important sur le coût.
- **Baisse des tokens générés :** La déduplication interne réduit le volume des extractions redondantes (-14% à -20% de tokens)
  ➡️ Réduction de la latence et du coût

HippoRAG2 reste plus économique (x5 à x7), mais au prix d'une perte de richesse sémantique (graphe binaires vs hypergraphe)

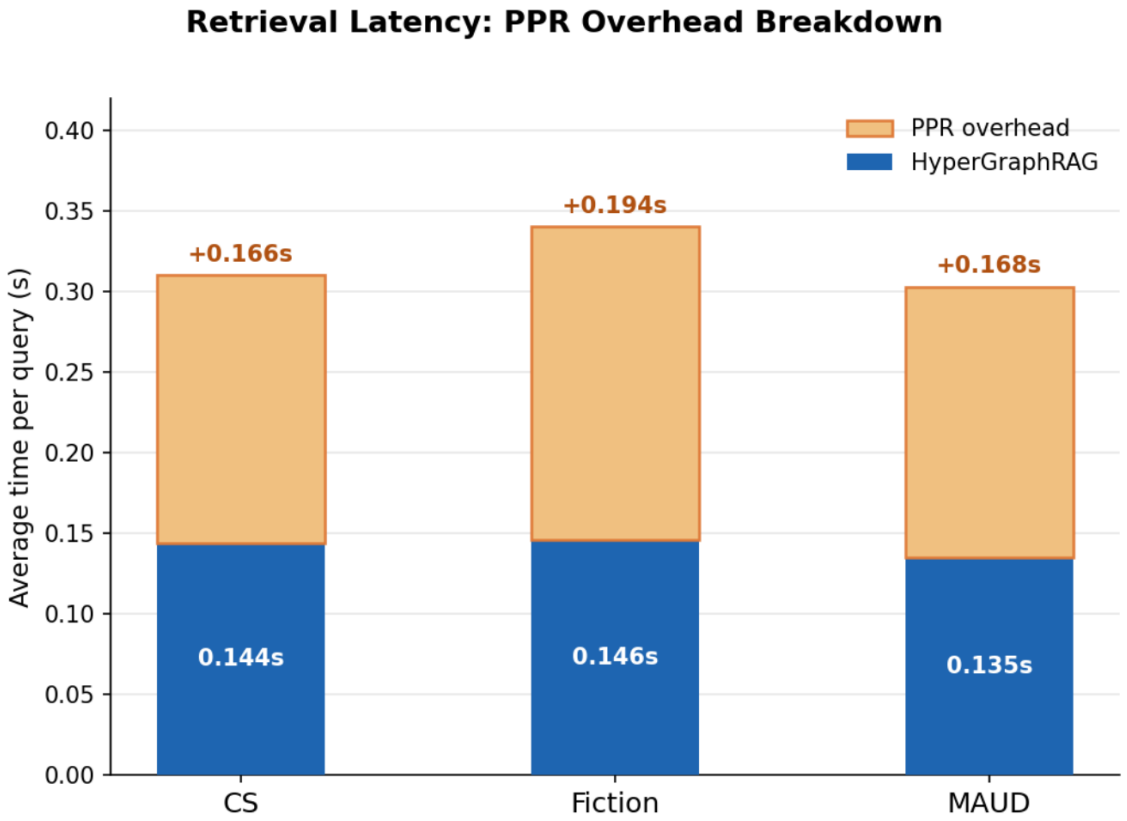


Latence de la récupération

Surcoût minime du PPR

Un surcoût de PPR **computationnel négligeable :**

- Latence de base (sans PPR) : entre 0,135 sec et 0,146 sec par requête.
- **Overhead du PPR** : Ajoute **moins de 0,2 seconde** par requête.
- **Exécution ciblée**: PPR opère uniquement sur un **sous-graphe** centré sur les nœuds pertinents de la requête.



04

Conclusion & Perspectives

Bilan et directions futures

Ce que nous retenons

Hypergraphe fiabilisé

L'extraction par auto-cohérence améliore la structure du graphe et facilite le raisonnement multi-sauts.

Graphe triparti + PPR

L'intégration du PPR exploite la topologie du graphe, augmentant la qualité des réponses.

Flexibilité coût/qualité

L'extraction optimisée est paradoxalement moins chère/plus rapide, et le surcoût de récupération par PPR est marginale.

Viabilité à grande échelle

L'hypergraphe fournit un contexte si qualitatif qu'un modèle compact (ex: GPT-4o mini) suffit pour garantir une bonne performance.

Perspectives

🚶 **Diffusion native sur hypergraphes :** Explorer des marches aléatoires nativement conçues pour les hypergraphes pour mieux exploiter les relations n-aires.

✂️ **Pruning contextuel :** Intégrer un mécanisme d'élagage pour réduire le bruit contextuel des entités et hyperarêtes.

📝 **Condensation des hyperarêtes :** Repenser les hyperarêtes comme des résumés d'informations atomiques (plutôt que des phrases *verbatim*) pour réduire le volume textuel et les coûts.

Merci

Questions & Discussion

Code & données disponibles :

github.com/qlik-oss/HyperGraphRAG