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

Évaluer la récupération dans les systèmes RAG pour la question-réponse financière sur longs documents

Amine Kobeissi [1, 2] Philippe Langlais[1, 2]
(1) Université De Montréal
(2) RALI
amine.kobeissi@umontreal.ca, felipe@iro.umontreal.ca

RÉSUMÉ

La génération augmentée par récupération (RAG) est de plus en plus utilisée pour le question-réponse financier sur de longs documents financiers, mais sa fiabilité dépend de la capacité à récupérer précisément le contexte justifiant la réponse. Nous étudions un mode d'échec fréquent où le bon document est trouvé, mais la page ou le segment contenant l'information est manqué, poussant le générateur à extrapoler à partir d'un contexte incomplet. Nous évaluons la récupération à plusieurs granularités (document, page, segment) et proposons une analyse par oracles fournissant des bornes supérieures empiriques sur la récupération et la génération. Sur 150 questions de FinanceBench, nous comparons des stratégies denses, clairsemées, hybrides et hiérarchiques, avec reformulation de requêtes et reranking. Enfin, nous introduisons un scoreur de pages adapté au domaine, entraîné pour la pertinence au niveau page, qui améliore le rappel des pages et la qualité des segments récupérés.

ABSTRACT

Evaluating Retrieval in RAG Systems for Financial Question Answering over Long Documents

Retrieval-augmented generation (RAG) is increasingly used for financial question answering over long financial documents, but reliability depends on retrieving the exact evidence supporting an answer in high-stakes settings. We study a common failure mode where the correct document is retrieved but the answer-bearing page or chunk is missed, pushing the generator to infer from incomplete context. We evaluate retrieval at document, page, and chunk granularities and use oracle settings to estimate empirical upper bounds on retrieval and generation. On a 150-question subset of FinanceBench, we compare dense, sparse, hybrid, and hierarchical retrieval with query reformulation and reranking. Finally, we introduce a domain fine-tuned page scorer trained for page-level relevance, yielding gains in page recall and improving the quality of retrieved chunks.

MOTS-CLÉS : Génération augmentée par la recherche ; Traitement automatique des langues ; Recherche d'information ; Question-réponse financière ; Grands modèles de langage.

KEYWORDS: Retrieval-Augmented Generation ; Natural Language Processing ; Information Retrieval ; Financial Question Answering ; Large Language Models.

1 Introduction

Les grands modèles de langue (LLM) produisent souvent des réponses fluides et exactes, mais des tâches spécialisées comme le question-réponse financier exigent un ancrage vérifiable dans de longs documents sources. La génération augmentée par récupération (RAG) répond à ce besoin en

Actes des 33ème Conférence sur le Traitement Automatique des Langues Naturelles.
Volume 1 : articles scientifiques originaux, pages 259–272.

sélectionnant des passages dans un corpus qui conditionnent la réponse.

Un système RAG peut récupérer le bon document tout en manquant la page ou le segment (chunk) qui contient la réponse, en particulier dans des rapports longs avec des gabarits répétés, des tableaux denses et des sections similaires d'une année à l'autre. Lorsque les segments corrects ne sont pas récupérés, le générateur peut alors produire des réponses plausibles mais erronées. De plus, les métriques au niveau document peuvent masquer les cas où le bon document est récupéré mais où le contexte porteur de la réponse est manqué, et les métriques de génération (exactitude des réponses, ROUGE-L) ne distinguent pas directement les échecs de récupération des échecs de génération.

Nous étudions ce mode d'échec en question-réponse financier sur des documents de la *U.S. Securities and Exchange Commission* (SEC) à l'aide de FinanceBench (Islam *et al.*, 2023). On observe que les techniques standards de RAG restent nettement en dessous des réglages oracles, révélant un écart de récupération. Nous introduisons un scoreur de pages affiné sur le domaine qui classe les pages avant la récupération de segments, réduisant l'écart de récupération observé.

2 Travaux connexes

La génération augmentée par récupération combine la recherche de passages et la génération conditionnelle afin d'améliorer la factualité et la traçabilité en ancrant les réponses dans du texte récupéré (Lewis *et al.*, 2020). Dans les contextes de documents longs, la récupération devient le principal goulot d'étranglement, car de nombreux passages sont thématiquement similaires alors que seule une petite fraction contient le contexte porteur de la réponse (Izacard & Grave, 2021). Cela motive des évaluations qui mesurent non seulement si le bon document source est trouvé, mais aussi si la récupération localise la page et la zone (segment) optimales nécessaires pour une réponse vérifiable.

Le question-réponse financier est difficile car les documents sont souvent très longs, semi-structurés et sensibles aux valeurs numériques. Des jeux de données tels que FinQA (Chen *et al.*, 2021), TAT-QA (Zhu *et al.*, 2021) et ConvFinQA (Chen *et al.*, 2022) mettent l'accent sur le raisonnement numérique, souvent avec des calculs multi-étapes fondés sur des tableaux et du texte narratif. FinanceBench ajoute des annotations de référence dans des documents SEC et permet une évaluation fondée sur les preuves à grande échelle (Islam *et al.*, 2023). Des suites plus récentes orientées récupération, comme SEC-QA (Lai *et al.*, 2024), soulignent également des flux de travail où les utilisateurs ont besoin à la fois de la réponse et du contexte récupéré qui la justifie.

La récupération pour les documents repose généralement sur le découpage en segments, des méthodes clairsemées, denses ou hybrides, la reformulation de requêtes et le réordonnancement (reranking). Des travaux récents ont insisté sur l'importance d'un prétraitement et de stratégies de récupération adaptés au domaine financier. Kim *et al.* (2025) proposent un *pipeline* en trois phases combinant expansion de requêtes, prétraitement du corpus, récupération hybride avec plongement de mots affinés, et un réordonnanceur entraîné par DPO avec sélection de documents. Évaluée sur plusieurs jeux de données de QA financier, leur méthode améliore nettement la précision de récupération. Wang *et al.* (2025) introduisent FinSage, un cadre RAG multi-aspect qui unifie un prétraitement multi-modal, une récupération multi-chemins et un reranker spécialisé au domaine, obtenant de bons résultats de récupération et des gains d'exactitude sur FinanceBench. Bien que ces systèmes obtiennent de bonnes performances de bout en bout, ils ne décomposent pas explicitement les échecs de récupération entre découverte du document et récupération intra-document.

Notre travail complète ces approches en introduisant une analyse par oracles qui quantifie la marge de progression due à une récupération imparfaite du document et/ou des pages ou segments dans le bon document. Nous montrons que, même si des méthodes comme HyDE et le reranking améliorent la découverte du document, des écarts substantiels au niveau page et segment persistent. Pour y remédier, nous introduisons un scoreur de pages affiné sur le domaine.

Entreprise	Amcor
Document	`AMCOR_2020_10K`
Type de question	Métriques générées
Raisonnement	Extraction d'information
Question	What is Amcor's year end FY2020 net AR (in USD millions) ? Address the question by adopting the perspective of a financial analyst who can only use the details shown within the balance sheet.
Réponse	$1616.00
Page de référence	49
Texte de référence	Amcor plc and Subsidiaries Consolidated Balance Sheet (in millions) As of June 30, 2020 2019 Assets Current assets : Cash and cash equivalents $ 742.6 $ 601.6 Trade receivables, net **1,615.9** 1,864.3 Inventories...

TABLE 1 – Exemple issu de FinanceBench. Le texte de référence est le passage extrait de la page de référence contenant l'information porteuse de la réponse.

3 Données et protocole expérimental

3.1 Jeu de données

Nous utilisons le sous-ensemble en accès libre de FinanceBench (Islam *et al.*, 2023), composé de 150 paires question-réponse sur 84 documents PDF uniques. Bien que les auteurs proposent un corpus complet plus large, nos demandes d'accès n'ont pas abouti, ce qui limite nos expériences à ce sous-ensemble public. Chaque exemple comprend une question, une réponse de référence et des annotations de vérité identifiant le document source et les pages de référence. La Table 1 montre un exemple simplifié de ce jeu de données. FinanceBench comprend trois types de questions. Les questions *domain relevant* exigent souvent de localiser des énoncés narratifs, peuvent référencer plusieurs périodes, et correspondent à des questions génériques pertinentes pour l'analyse financière d'entreprises cotées. Les questions *metrics generated* (comme dans la Table 1) requièrent une récupération très précise vers le bon énoncé ou tableau afin d'extraire des valeurs de base pour des calculs ultérieurs ; elles demandent du calcul et du raisonnement sur des grandeurs financières. Les questions *novel generated* visent davantage le raisonnement et sont très spécifiques à l'entreprise, au secteur et au rapport considéré ; elles ne sont pas conçues pour être purement extractives. Des exemples représentatifs pour les types *domain relevant* et *novel generated* sont fournis en Annexe A.

FinanceBench couvre quatre types de documents SEC, résumés dans la Table 2. La difficulté de récupération varie selon le type : les rapports annuels (10-K) sont longs et très structurés, tandis que les transcriptions d'appels de résultats sont conversationnelles et moins prévisibles.

Le corpus résultant, dont les statistiques sont résumées dans la Table 3, couvre 12 013 pages et 15 171

Type	Documents	Questions (%)	Description
10-K	64	112 (74.7%)	Rapport annuel : facteurs de risque, MD&A, états financiers audités et notes.
10-Q	8	15 (10.0%)	Rapport trimestriel : structure similaire au 10-K, non audité, plus court.
EC	6	14 (9.3%)	Transcription d'appel de résultats : format conversationnel, avec des éléments de preuve souvent répartis dans les questions des analystes et les réponses de la direction.
8-K	6	9 (6.0%)	Rapport courant : divulgation d'événements importants, souvent plus bref et centré sur un événement précis.
Total	**84**	**150 (100%)**	

TABLE 2 – Descriptions des types de documents dans le corpus FinanceBench (84 documents, 150 questions).

segments, avec une forte variabilité de longueur entre documents (143 ± 99 pages en moyenne). Cette hétérogénéité, combinée à la sur-représentation des 10-K dans les données d'entraînement, motivera l'analyse des performances par type de document en Section 5.

Statistique	Valeur
Questions	150
Documents	84
Pages totales	12 013
Pages par document	143.0 ± 98.7
Segments totaux	15 171
Tokens par requête	40 ± 24
Tokens par réponse	19.5 ± 24.1

TABLE 3 – Statistiques du corpus FinanceBench. Les segments sont construits avec une fenêtre de 1024 tokens et un chevauchement de 128 tokens.

3.2 Mise en place de la tâche

La Figure 1 illustre le *pipeline* RAG étudié. Soit $\mathcal{D} = \{d_1, \ldots, d_D\}$ un ensemble de documents PDF. Chaque document $d \in \mathcal{D}$ contient n pages $\{p_1^d, \ldots, p_n^d\}$, extraites avec le prétraitement de FinanceBench [1]. Nous construisons un corpus de segments $\mathcal{C}$ en découpant les pages en fragments de 1024 tokens avec un chevauchement de 128 tokens [2]. Chaque segment est un triplet (c, d, p), où c est le texte du segment, d l'identifiant du document et p le numéro de page. Pour une requête q, un récupérateur produit une liste ordonnée de segments, $\mathcal{R}_k(q) = \mathrm{Top}_{k\ (c,d,p)\in\mathcal{C}}\ s(q,c) = \{(c_i, d_i, p_i)\}_{i=1}^{k}$, où s est une fonction de score. Le générateur produit une réponse $\hat{y} = G_\theta(q, \mathcal{R}_k(q))$ (gabarit en Annexe B). FinanceBench fournit, pour chaque requête, un document $d^\star$ et des pages $P^\star$ de référence.

1. https://github.com/patronus-ai/financebench
2. Conformément à Yepes et al. (Yepes *et al.*, 2024), des segments plus longs améliorent la RAG financière.

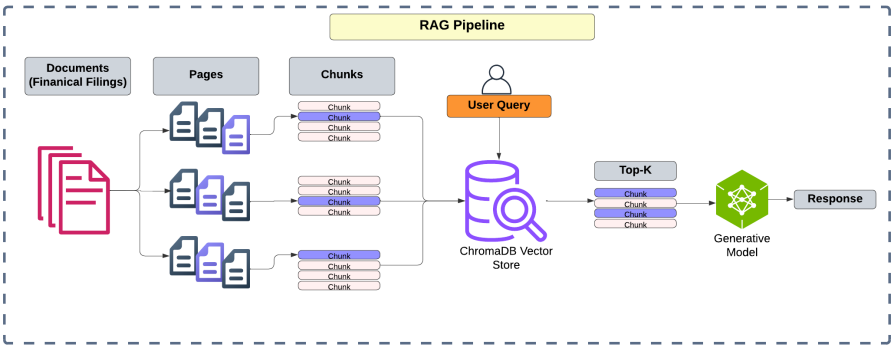

FIGURE 1 – Vue d'ensemble du pipeline RAG. Pour une requête, les pages et segments en bleu représentent le contexte de référence contenant la réponse.

3.3 Conditions de récupération *oracle*

Nous définissons trois réglages de récupération qui diffèrent par la restriction de l'espace de recherche. (i) **Récupération standard** : on récupère sur l'espace de recherche complet. (ii) **Oracle document** : les candidats sont restreints au document de référence, $\mathcal{C}^{\text{doc}}(d^\star) = \{(c, d, p) \in \mathcal{C} \mid d = d^\star\}$. (iii) **Oracle page** : les candidats sont restreints au document et aux pages de référence, $\mathcal{C}^{\text{page}}(d^\star, P^\star) = \{(c, d, p) \in \mathcal{C} \mid d = d^\star,\ p \in P^\star\}$, ce qui réduit presque entièrement le bruit.

Ces réglages fournissent des bornes supérieures empiriques. Le réglage *Oracle document* quantifie la marge de progression due à une découverte imparfaite des pages et des segments. Le réglage *Oracle page* quantifie la marge de progression due à une récupération imparfaite des segments lorsque le document et les pages de référence sont connus. L'évaluation de ces deux bornes aide à isoler l'origine des erreurs et fournit une référence commune pour comparer les systèmes. L'écart à l'oracle normalise les différences de *pipeline* et contextualise les scores bruts.

3.4 Métriques d'évaluation

Pour chaque question q, FinanceBench fournit un document de référence $d^\star(q)$ et un ensemble de pages de référence $P^\star(q)$. Soit la liste des k meilleurs segments récupérés $\mathcal{R}_k(q) = \{(c_i, d_i, p_i)\}_{i=1}^k$, où d_i et p_i désignent respectivement l'identifiant du document et l'index de page du segment c_i. Nous mesurons d'abord le rappel au niveau document. Chaque question ayant un seul document de référence, le rappel à k équivaut à un *hit* : $\text{DocRec@}k(q) = \mathbb{I}\left[d^\star(q) \in \{d_i\}_{i=1}^k\right]$. Nous mesurons ensuite le rappel au niveau page à k, où il peut y avoir plusieurs pages de référence, $\text{PageRec@}k(q) = \left|P^\star(q) \cap \{p_i \mid d_i = d^\star(q)\}_{i=1}^k\right| / |P^\star(q)|$.

Enfin, comme proxy au niveau segment, nous calculons la similarité maximale ROUGE-L (Lin, 2004) et BLEU (Papineni *et al.*, 2002) entre un segment récupéré, c_i, et les passages de référence concaténée en un seul texte, $e^\star(q)$. Pour évaluer le générateur, nous utilisons ROUGE-L entre la sortie générée et la réponse de référence, ainsi qu'un *numeric match* pour les questions de type métriques. Pour ce dernier, étant donnés une réponse de référence $y^\star$ et une prédiction $\hat{y}$, nous extrayons des ensembles de nombres $R = (y^\star)$ et $P = (\hat{y})$ via une expression régulière après suppression des virgules et symboles monétaires. Nous comptons un exemple comme correct s'il existe $r \in R$ et $p \in P$ tels que r et p soient proches selon `numpy.isclose`, en utilisant $\texttt{atol} = 0.03$ et $\texttt{rtol} = 0.03$.

Nous rapportons des moyennes macro sur les 150 questions et utilisons $k = 5$ dans toute l'évaluation, sauf pour le *numeric match* qui n'est rapporté que sur les 50 questions de type métriques.

4 Approches comparées

Nous évaluons un ensemble de stratégies de récupération visant à améliorer la découverte au niveau document, page et segment. Toutes les stratégies partagent la même mise en place de la tâche, décrite en Section 3.2. Pour les méthodes denses, nous utilisons un magasin vectoriel commun, construit une seule fois par configuration. Les segments sont stockés dans une collection unique `ChromaDB`,[3] et la récupération utilise `similarity_search`.

Récupération dense La récupération dense projette une requête et chaque segment dans un espace vectoriel partagé (Karpukhin *et al.*, 2020), puis classe les segments par similarité. Soit $E(\cdot)$ le modèle de plongement et c un segment. Nous calculons $E(q)$ et récupérons les k meilleurs segments par similarité cosinus. Toutes les variantes denses ne diffèrent que par le choix du *backbone* de plongement, en partageant le même magasin vectoriel, la même segmentation et les mêmes métadonnées.

Récupération clairsemée La récupération clairsemée excelle dans l'appariement de mots-clés, ce qui peut être avantageux en QA financier. Nous testons BM25 (Robertson & Zaragoza, 2009) comme base clairsemée, et SPLADE (Formal *et al.*, 2021) comme récupérateur clairsemé appris produisant des vecteurs pondérés en termes. BM25 est implémenté via `rank-bm25`[4]. La tokenisation utilise un prétraitement qui préserve les symboles monétaires, pourcentages et nombres financiers. Le récupérateur retourne les k segments aux scores BM25 les plus élevés. Pour SPLADE, nous utilisons le checkpoint `naver/splade-cocondenser-ensembledistil`. En raison de contraintes architecturales, nous tronquons les entrées à 512 tokens et conservons les $N = 256$ termes les plus pondérés pour l'efficacité, ce qui peut affecter les résultats SPLADE car le modèle voit moins de contexte par segment que les méthodes à 1024 tokens.

Fusion hybride Nous fusionnons des listes classées issues de méthodes denses et clairsemées avec la *Reciprocal Rank Fusion* (RRF) (Cormack *et al.*, 2009). Soit $\text{rank}_r(c)$ le rang du segment c sous le récupérateur r. RRF assigne : $s_{\text{RRF}}(c) = \sum_{r \in \mathcal{M}} \frac{1}{k_{\text{rrf}} + \text{rank}_r(c)}$, où $k_{\text{rrf}} = 60$ et les méthodes sont pondérées de manière uniforme. Pour fusionner de façon robuste des sorties de différents systèmes, les segments sont alignés via des hachages SHA1 sur le texte et les métadonnées (source et page).

Reformulation de requêtes, HyDE et Multi-HyDE Pour réduire le décalage lexical, nous testons une expansion déterministe d'acronymes/expressions temporelles (CAPEX $\rightarrow$ *capital expenditure*, FY18 $\rightarrow$ *fiscal year 2018*), utilisée uniquement pour la récupération. Nous évaluons ensuite HyDE et Multi-HyDE (Gao *et al.*, 2023; George *et al.*, 2025), HyDE génère un passage hypothétique h avec Qwen-2.5-7B-Instruct (température $T = 0.7$, 200 tokens), puis effectue la récupération dense en encodant h avec l'encodeur de segments $E_{\text{chunk}}(\cdot)$, i.e., en utilisant $E_{\text{chunk}}(h)$ plutôt que $E_{\text{chunk}}(q)$. Multi-HyDE génère 4 passages $\{h_i\}_{i=1}^{4}$ et utilise la requête moyenne $\bar{h} = \frac{1}{4} \sum_{i=1}^{4} E_{\text{chunk}}(h_i)$.

Récupération hiérarchique La récupération hiérarchique traite les contextes longs en récupérant d'abord à une granularité grossière puis en raffinant à une granularité plus fine. Des travaux récents

3. `ChromaDB` : https://github.com/chroma-core/chroma
4. `rank-bm25` : https://github.com/dorianbrown/rank_bm25

ciblent la récupération fine dans les documents financiers sous la forme d'une récupération de documents suivie d'une récupération de passages, avec raffinement supplémentaire (Izacard & Grave, 2021; Choe *et al.*, 2025; Li *et al.*, 2025). Nous implémentons une hiérarchie simple *Parent-Child* : des segments enfants (fins) sont indexés pour la récupération, chacun mappé de façon déterministe vers un segment parent (plus large) utilisé comme contexte de génération. Après récupération des enfants, nous dédoublonnons par identifiant de parent et retournons jusqu'à k parents uniques, scorés par le meilleur score de similarité parmi leurs enfants associés.

Reranking par cross-encodeur Dans des documents longs, le reranking aide à départager des régions quasi-duplicata et à améliorer l'ordre des contextes récupérés après une première étape à fort rappel, ce qui est important lorsque plusieurs sections contiennent un langage similaire d'une année à l'autre ou entre segments d'activité (Li *et al.*, 2025). Nous ajoutons un reranker de seconde étape avec `BAAI/bge-reranker-v2-m3`. Pour chaque requête, nous récupérons d'abord $N = 20$ candidats via un récupérateur de base, puis nous rerankons avec le cross-encodeur et conservons les k meilleurs.

4.1 Notre récupérateur de pages/segments

Nous proposons une méthode hiérarchique qui identifie d'abord les pages pertinentes, puis récupère des segments uniquement sur ces pages, réduisant le bruit tout en préservant une sélection fine. Pour ce faire, nous affinons un bi-encodeur afin d'identifier des pages optimales, avec des plongements de pages pré-calculés pour une inférence efficace. La méthode comporte deux étapes : (i) classer les pages du corpus et retourner les P meilleures pages ; (ii) exécuter une récupération au niveau segment en ne considérant que les segments dont les métadonnées correspondent aux pages sélectionnées.

Pour chaque document d, nous extrayons le texte de chaque page p et appliquons une fonction de normalisation déterministe $N(\cdot)$ qui réduit les espaces et tronque les pages à une longueur maximale fixe de $M = 2000$ caractères. Le scoreur de pages appris remplace un encodeur pré-entraîné par un bi-encodeur affiné. Soit (d, p) le texte de la page ; nous calculons les plongements à partir de $N((d, p))$. Pour la fonction de score, nous entraînons un bi-encodeur E_{θ_p} initialisé depuis `BAAI/BGE-M3` (Chen *et al.*, 2025). Le score de pertinence au niveau page, $s_{\text{page}}(q, d, p)$, est la similarité cosinus entre le plongement appris de la page $E_{\theta_p}\big(N(d, p)\big)$ et celui de la requête $E_{\theta_p}(q)$. À l'inférence, les pages sont classées selon s_{page} et les P meilleures pages sont transmises à l'étape segment.

4.1.1 Entraînement

Pour chaque question q, FinanceBench fournit un document de référence $d^\star(q)$ et un ensemble d'indices de pages de référence $P^\star(q)$. Pour l'entraînement, nous construisons des positifs en associant chaque question à chacune des pages annotées dans son document de référence : $\mathcal{S} = \{(q, (d^\star(q), p^+)) \mid p^+ \in P^\star(q)\}$. Nous échantillonnons des mini-lots depuis la partie entraînement de $\mathcal{S}$ et appliquons un apprentissage contrastif pour affiner E_{θ_p} avec une perte de type *Multiple Negatives Ranking* (négatifs dans le lot) (Henderson *et al.*, 2017). Pour un mini-lot de taille B avec des paires positives $\{(q_i, (d_i^\star, p_i^+))\}_{i=1}^B$, la perte est :

$$\mathcal{L}_{\text{MNR}} = -\frac{1}{B}\sum_{i=1}^{B}\left[s_{\text{page}}(q_i, d_i^\star, p_i^+) - \log\sum_{j=1}^{B}\exp\left(s_{\text{page}}(q_i, d_j^\star, p_j^+)\right)\right].$$

Pour chaque question q_i, la page de référence associée est la cible positive, et toutes les autres pages du mini-lot jouent le rôle de négatifs. Nous entraînons [5] le scoreur de pages en utilisant des découpages au niveau document pour éviter toute fuite de données, afin qu'aucun document n'apparaisse à la fois dans l'entraînement et l'évaluation. Nous effectuons une validation croisée à 5 plis en partitionnant les questions selon leurs documents sources. Pour chaque pli, nous entraînons sur 80% et évaluons sur les 20% restants de documents. L'entraînement utilise un taux d'apprentissage de 2×10^{-5}, une taille de lot de 16 et 15 époques. Ces hyperparamètres suivent les pratiques standard d'affinage de modèles de type BGE (Chen *et al.*, 2025) et n'ont pas fait l'objet d'une optimisation systématique, ce qui constitue une piste d'amélioration. Les métriques finales sont agrégées sur l'ensemble des plis.

4.1.2 Inférence et intégration avec la récupération au niveau segment

Après l'entraînement, nous encodons chaque page du corpus avec E_{θ_p} et l'indexons avec des métadonnées (d, p). À l'inférence, étant donnée une requête q, nous récupérons les $P = 20$ (ablation dans Annexe C) meilleures pages selon s_{page}, filtrons les segments pour ne conserver que ceux issus des pages sélectionnées, puis appliquons une récupération au niveau segment afin de sélectionner les k segments finaux pour la génération. L'Algorithme 1 décrit la routine complète.

Algorithm 1 Récupération *page puis segment* avec scoreur de pages affiné

Require: Requête q, encodeur de pages affiné E_{θ_p} et de segments E_{chunk}, P pages, k segments
Require: Index de pages $\mathcal{I}_{\text{page}}$ stockant des entrées $\big((d,p),\, v_{d,p}^{\text{page}}\big)$
où $v_{d,p}^{\text{page}} = E_{\theta_p}\big(N((d,p))\big)$ est pré-calculé hors-ligne une seule fois
Require: Index de segments $\mathcal{I}_{\text{chunk}}$ stockant des entrées $\big(c,\, v_c^{\text{chunk}},\, d_c,\, p_c\big)$
où $v_c^{\text{chunk}} = E_{\text{chunk}}(c)$ est pré-calculé hors-ligne une seule fois
Étape 1 : récupération des pages
$e_q^{\text{page}} \leftarrow E_{\theta_p}(q)$
$\mathcal{P}_P(q) \leftarrow \textsc{Search}_{\cos}\big(\mathcal{I}_{\text{page}},\, e_q^{\text{page}},\, P\big)$ ▷ Recherche de plus proches voisins sur les $v_{d,p}^{\text{page}}$ stockés
Étape 2 : récupération des segments sur espace filtré
$e_q^{\text{chunk}} \leftarrow E_{\text{chunk}}(q)$
$\mathcal{C}_{\text{filt}} \leftarrow \{c \in \mathcal{I}_{\text{chunk}} \mid (d_c, p_c) \in \mathcal{P}_P(q)\}$
$\mathcal{R}_k(q) \leftarrow \textsc{Search}_{\cos}\big(\mathcal{C}_{\text{filt}},\, e_q^{\text{chunk}},\, k\big)$ ▷ Recherche sur les v_c^{chunk} stockés dans le pool filtré
return $\mathcal{R}_k(q)$

5 Résultats et analyse

Les performances de récupération agrégées sur les 150 questions sont proposées en Table 4. Les deux premières lignes correspondent aux performances oracles avec `BAAI/BGE-M3`. Lorsque l'espace de recherche est restreint (expérience *Oracle page*), les scores BLEU et ROUGE-L augmentent, indiquant un meilleur contexte récupéré. L'exactitude numérique augmente également sous *Oracle page* (voir Table 6), confirmant que la génération en aval s'améliore lorsque la bonne page est trouvée.

Pour toutes les méthodes testées, le rappel au niveau page est nettement inférieur au rappel au niveau document, ce qui montre que les méthodes de base parviennent souvent à identifier le bon

5. Expériences effectuées sur un GPU RTX 3090 avec 24 Go de mémoire.

document, mais peinent à retrouver le contexte porteur de la réponse à l'intérieur du document. Conformément à des travaux antérieurs, la récupération dense surpasse les méthodes clairsemées sur FinanceBench (Kim *et al.*, 2025), probablement en raison d'un décalage lexical entre les requêtes et le langage formel des documents. Parmi les baselines, `BGE-M3` combiné à Multi-HyDE et au reranking est le meilleur (0.46 de rappel page), ce qui indique que la reformulation de requêtes et le scoring par cross-encodeur améliorent la récupération intra-document. Néanmoins, en rappel page, cette méthode reste à 0.14 de la borne *Oracle document* (0.60), ce qui met en évidence l'écart de récupération. Notre scoreur de pages appris atteint 0.55 de rappel page, dépasse les baselines et réduit l'écart vis-à-vis de l'oracle *document*. Nous obtenons également les meilleurs scores BLEU (0.33) et ROUGE-L (0.46) au niveau segment. Cela confirme que le filtrage au niveau page améliore la qualité de récupération par rapport à une recherche directe sur les segments.

Méthode	DocRec@5	PageRec@5	Max BLEU@5	Max ROUGE-L@5
Oracle document	1.00	0.60	0.25	0.42
Oracle page	1.00	1.00	0.40	0.59
Dense (BGE-M3)	0.88	0.34	0.26	0.35
BM25	0.32	0.07	0.04	0.12
SPLADE	0.50	0.16	0.18	0.29
Fusion hybride (BM25 + BGE-M3)	0.61	0.23	0.13	0.27
Parent-Child	0.31	0.13	0.12	0.22
Expansion de requête	0.68	0.27	0.08	0.24
HyDE	0.86	0.40	0.25	0.37
Multi-HyDE	0.85	0.42	0.27	0.39
BGE-M3 + ReRanker	0.87	0.41	0.19	0.34
BGE-M3 + Multi-HyDE + ReRanker	0.93	0.46	0.28	0.40
Scoreur de pages appris (notre méthode)	**0.95**	**0.55**	**0.33**	**0.46**

TABLE 4 – Performances de récupération agrégées sur 150 exemples FinanceBench à $k = 5$.

5.1 Types de questions et de documents

La Figure 2 montre que les performances varient fortement selon le type de question. Sur les questions *metrics*, notre scoreur atteint un rappel page de 0.81, supérieur à *Oracle document* (0.68), alors que les deux atteignent un rappel document parfait. Cela s'explique par le fait que les questions *metrics* ciblent des tableaux et énoncés numériques dans des sections structurellement prévisibles, ce qui permet aux plongements adaptés au domaine de récupérer efficacement la bonne page. À l'inverse, les questions *domain* et *novel* présentent des écarts plus importants vis-à-vis de leurs bornes oracles, car elles requièrent un contexte narratif plus difficile à capturer.

Comme indiqué dans la Table 5, les performances varient fortement selon le type de document. Sur les documents 10K, qui représentent 75% du jeu de données, le scoreur de pages appris atteint 0.62 de rappel page, dépassant à la fois la meilleure baseline (0.45) et *Oracle document* (0.56). Cette performance sur les 10K est probablement due au volume d'exemples de ce type dans les données d'entraînement et reflète la structure relativement constante des rapports annuels. En revanche, les performances sur les transcriptions d'appels de résultats (*Earnings Call*, EC) sont faibles (0.10), loin

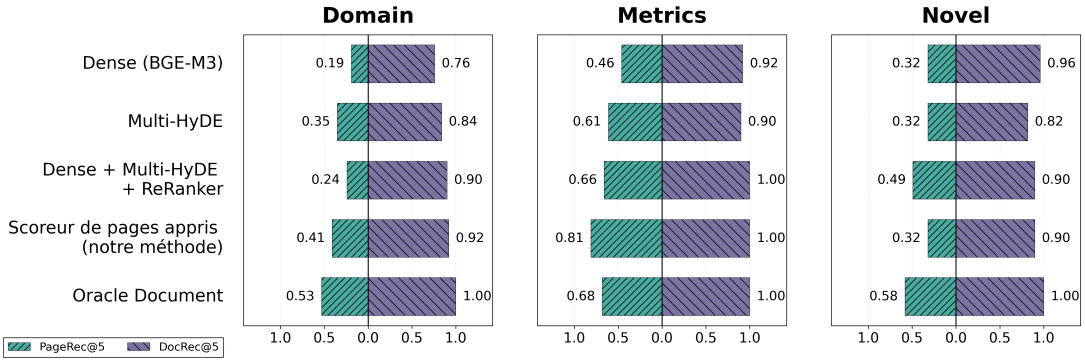

FIGURE 2 – Rappel au niveau document et page à $k = 5$ par type de question (50 questions par type).

de la borne oracle (0.64) et de la baseline (0.36). Les appels de résultats sont moins standardisés et les passages de référence peuvent apparaître dans des sections de questions-réponses conversationnelles, différentes des données d'entraînement. Les performances sur 10Q et 8K sont également inférieures à la baseline, probablement à cause d'une faible représentation de ces types de documents pendant l'entraînement.

	10K (n=112)		10Q (n=15)		8K (n=9)		EC (n=14)	
	D@5	P@5	D@5	P@5	D@5	P@5	D@5	P@5
BGE-M3 + Multi-HyDE + ReRanker	0.96	0.45	0.87	0.47	0.89	0.78	0.86	0.36
Scoreur de pages appris (notre méthode)	0.97	0.62	0.87	0.40	0.89	0.56	0.86	0.10
Oracle document	1.00	0.56	1.00	0.60	1.00	0.89	1.00	0.64

TABLE 5 – Rappel au niveau document (D@5) et page (P@5) à $k = 5$. EC = *Earnings Call*.

5.2 Résultats de génération

Malgré les améliorations de rappel page, il est important d'évaluer si la génération en aval bénéficie d'une meilleure récupération des preuves. La Table 6 montre la qualité des réponses avec Qwen-2.5-7B-Instruct. Notre scoreur de pages appris atteint 0.50 de numeric match, dépassant la meilleure baseline (0.38) et Oracle document (0.44), tout en améliorant ROUGE-L par rapport à la baseline la plus forte (0.15 contre 0.12). Cela est cohérent avec la Figure 2, où notre méthode dépasse le réglage Oracle document sur les questions metrics. Ces résultats suggèrent que le filtrage au niveau page cible efficacement les pages contenant les preuves numériques nécessaires à des calculs précis, tout en maintenant la qualité des réponses sur les questions extractives et de raisonnement.

Méthode	ROUGE-L (*NM*)	Méthode	ROUGE-L (*NM*)
BGE-M3 + Multi-HyDE + ReRanker	0.12 (0.38)	Oracle doc	0.13 (0.44)
Scoreur de pages appris (notre méthode)	0.15 (0.50)	Oracle page	0.16 (0.70)

TABLE 6 – Résultats de génération. Le *numeric match* (NM) suit la tolérance de FinanceBench.

6 Conclusion

Nous avons introduit un cadre basé sur des oracles pour décomposer les échecs de récupération en question-réponse financier, en séparant la découverte du document de la récupération au niveau page et segment au sein du bon document. Sur FinanceBench, et à travers des stratégies de récupération variées, nous constatons que si les méthodes actuelles améliorent le rappel document, des écarts au niveau page persistent même lorsque le bon document est récupéré. Pour adresser ce goulot d'étranglement, nous proposons un scoreur de pages qui classe les pages au sein d'un document puis restreint la récupération de segments aux pages les mieux classées. Sur ce sous-ensemble, notre approche atteint 55 % de rappel au niveau page, ce qui constitue le meilleur résultat parmi les méthodes non-oracles testées. Pour les questions de type *metrics generated*, elle surpasse même le réglage *Oracle document* en rappel page. Ces résultats suggèrent qu'un scoreur explicite au niveau des pages, affiné au domaine financier, constitue une piste prometteuse pour réduire l'écart de récupération intra-document, bien que sa généralisation reste à confirmer sur des corpus plus larges.

Notre étude se concentre sur les 150 questions en accès libre de FinanceBench, majoritairement issues de documents 10-K, ce qui constitue une limite importante. Les types de documents sous-représentés comme les transcriptions d'appels de résultats et les rapports 8-K ne permettent pas de tirer des conclusions fermes sur ces catégories. L'évaluation sur l'intégralité de FinanceBench ou d'autres benchmarks financiers reste une priorité pour valider la généralisation de l'approche. Le scoreur de pages requiert des annotations de référence au niveau page et s'appuie sur des négatifs aléatoires *in-batch*. Une stratégie de *hard negative mining* ciblant des pages thématiquement proches mais non pertinentes pourrait améliorer la discrimination. Notre segmentation fixe à 1024 tokens avec chevauchement n'a pas été comparée à des approches alternatives telles que la segmentation thématique ou sémantique, qui pourraient mieux préserver les frontières naturelles des tableaux financiers. L'évaluation de la génération repose sur ROUGE-L et l'appariement numérique, qui ne capturent qu'imparfaitement la correction factuelle en finance, et sur un seul LLM, ce qui limite les conclusions sur la robustesse de la génération.

Parmi les pistes prometteuses figurent la combinaison du scoreur de pages avec le reranking ou l'augmentation de requêtes, l'exploration de stratégies de segmentation adaptées aux structures tabulaires des documents financiers, l'utilisation de l'apprentissage par renforcement pour une récupération itérative, et l'extension de l'approche à des questions multi-documents nécessitant l'agrégation de contexte entre plusieurs rapports.

Références

CHEN J., XIAO S., ZHANG P., LUO K., LIAN D. & LIU Z. (2025). M3-embedding : Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

CHEN Z., CHEN W., SMILEY C., SHAH S., BOROVA I., LANGDON D., MOUSSA R., BEANE M. I., HUANG T.-H. K., ROUTLEDGE B. R. & WANG W. Y. (2021). Finqa : A dataset of numerical reasoning over financial data. *ArXiv*, **abs/2109.00122**.

CHEN Z., LI S., SMILEY C., MA Z., SHAH S. & WANG W. Y. (2022). Convfinqa : Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, p. 6279–6292.

CHOE J., KIM J. & JUNG W. (2025). Hierarchical retrieval with evidence curation for open-domain financial question answering on standardized documents. In *Findings of the Association for Computational Linguistics : ACL 2025*, p. 16663–16681 : Association for Computational Linguistics. DOI : 10.18653/v1/2025.findings-acl.855.

CORMACK G. V., CLARKE C. L. & BUETTCHER S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, p. 758–759.

FORMAL T., PIWOWARSKI B. & CLINCHANT S. (2021). Splade : Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 2288–2292.

GAO L., MA X., LIN J. & CALLAN J. (2023). Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1762–1777.

GEORGE R., SRINIVASAN A. G., JOE J. K., MR H., KANT H., VIMALKANTH R., SURESH S. *et al.* (2025). Enhancing financial rag with agentic ai and multi-hyde : A novel approach to knowledge retrieval and hallucination reduction. In *Proceedings of The 10th Workshop on Financial Technology and Natural Language Processing*, p. 19–32.

HENDERSON M., AL-RFOU R., STROPE B., HSUAN SUNG Y., LUKACS L., GUO R., KUMAR S., MIKLOS B. & KURZWEIL R. (2017). Efficient natural language response suggestion for smart reply.

ISLAM P., KANNAPPAN A., KIELA D., QIAN R., SCHERRER N. & VIDGEN B. (2023). Financebench : A new benchmark for financial question answering.

IZACARD G. & GRAVE E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, p. 874–880, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.eacl-main.74.

KARPUKHIN V., OGUZ B., MIN S., LEWIS P. S., WU L., EDUNOV S., CHEN D. & YIH W.-T. (2020). Dense passage retrieval for open-domain question answering. In *"Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)"*, p. 6769–6781.

KIM S., SONG H., SEO H. & KIM H. (2025). Optimizing retrieval strategies for financial question answering documents in retrieval-augmented generation systems.

LAI V. D., KRUMDICK M., LOVERING C., REDDY V., SCHMIDT C. & TANNER C. (2024). Sec-qa : A systematic evaluation corpus for financial qa.

LEWIS P., PEREZ E., PIKTUS A., PETRONI F., KARPUKHIN V., GOYAL N., KÜTTLER H., LEWIS M., YIH W.-T., ROCKTÄSCHEL T. *et al.* (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, **33**, 9459–9474.

LI Y., WANG M., DE CARVALHO M., SABANIS S. & MA T. (2025). Fingear : Financial mapping-guided enhanced answer retrieval.

LIN C.-Y. (2004). ROUGE : A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, p. 74–81, Barcelona, Spain : Association for Computational Linguistics.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, p. 311–318.

ROBERTSON S. E. & ZARAGOZA H. (2009). The probabilistic relevance framework : Bm25 and beyond. *Found. Trends Inf. Retr.*, **3**, 333–389.

WANG X., CHI J., TAI Z., KWOK T. S. T., HE H., LI Z., HUA Y., LI M., LU P., WANG S. *et al.* (2025). Finsage : A multi-aspect rag system for financial filings question answering. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, p. 6144–6152.

YEPES A. J., YOU Y., MILCZEK J., LAVERDE S. & LI R. (2024). Financial report chunking for effective retrieval augmented generation.

ZHU F., LEI W., HUANG Y., WANG C., ZHANG S., LV J., FENG F. & CHUA T.-S. (2021). TAT-QA : A question answering benchmark on a hybrid of tabular and textual content in finance. In C. ZONG, F. XIA, W. LI & R. NAVIGLI, Éds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 3277–3287, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.254.

Annexe

A Exemples par type de question

Entreprise	Nike
Document	NIKE_2023_10K
Type de question	Domain relevant
Raisonnement	Raisonnement Numérique
Question	Among operating, investing, and financing activities, which brought in the most (or lost the least) cash flow for Nike in FY2023 ?
Réponse	Cash flow from operations was the highest in FY2023.
Page de référence	61
Texte de référence	NIKE, INC. CONSOLIDATED STATEMENTS OF CASH FLOWS. YEAR ENDED MAY 31 (Dollars in millions). 2023 2022 2021. Cash provided (used) by operations : Net income $5,070 $6,046 $5,727. ... **Cash provided (used) by operations 5,841 5,188 6,657.** ... **Cash provided (used) by investing activities 564 (1,524) (3,800).** ... **Cash provided (used) by financing activities (7,447) (4,836) (1,459).** ...

TABLE 7 – Exemple de question *domain relevant* issu de FinanceBench.

Entreprise	PepsiCo
Document	`PEPSICO_2023_8K`
Type de question	Novel generated
Raisonnement	—
Question	By how much did Pepsico increase its unsecured five year revolving credit agreement on May 26, 2023 ?
Réponse	$400,000,000 increase.
Page de référence	1
Texte de référence	Effective May 26, 2023, PepsiCo terminated the **$3,800,000,000 five year unsecured revolving credit agreement** (2022 Five Year Credit Agreement). ... On May 26, 2023, PepsiCo entered into a new **$4,200,000,000 five year unsecured revolving credit agreement** (2023 Five Year Credit Agreement) among PepsiCo, as borrower, the lenders party thereto, and Citibank, N.A., as administrative agent. ...

TABLE 8 – Exemple de question *novel generated* issu de FinanceBench.

B Gabarit de prompt du générateur

```
System: You are a financial analyst assistant. Your task is to provide
accurate, concise, and well-supported answers to questions based on
provided financial document segments.
Context: {retrieved_evidence}
Question: {query}
Answer:
```

C Performance du scoreur de pages appris selon P

P	DocRec@5	PageRec@5	Max BLEU@5	Max ROUGE-L@5
5	0.92	0.48	0.30	0.44
10	0.95	0.48	0.31	0.44
20	0.95	0.55	0.33	0.46

TABLE 9 – Ablation de la valeur de P pour le scoreur de pages appris