RAG pour Question Answering (HyDe vs RAG-Fusion)

Mohamed Ali Lajnef[1] , **Mathis Pernias**[2] , **Mathis Nguyen**[3]
Polytechnique Montréal

Abstract

Ce papier compare deux approches avancées de la génération augmentée par récupération (RAG) avec HyDE, qui s'appuie sur un document hypothétique généré à partir de la question, et RAG Fusion, qui combine les résultats de plusieurs reformulations de la requête initiale. En prolongeant le modèle RAG initial, ces méthodes visent à optimiser la phase de récupération pour renforcer la pertinence des réponses générées. Notre étude cherche à évaluer leur impact en termes de précision, de généralisation et d'efficacité.

1 Introduction

Les systèmes QA sont de plus en plus répandus dans les applications d'intelligence artificielle, qu'il s'agisse d'agents conversationnels ou de moteurs de réponse spécialisés. Les grands modèles de langage (LLMs) jouent un rôle central dans cette évolution, car ils permettent de générer des réponses en langage naturel de manière fluide à partir d'une requête utilisateur. Tirant leur puissance de vastes corpus d'entraînement, les LLMs restent limités par leur dépendance à des données figées, ce qui entraîne plusieurs contraintes bien connues. En particulier, les LLMs ne disposent d'aucun mécanisme d'accès à des sources d'information actualisées ou spécialisées au moment de l'inférence. Ils peuvent ainsi générer des réponses obsolètes ou inexactes, sans qu'il soit possible de remonter facilement à la source des informations utilisées.

La génération augmentée par récupération (RAG) a été introduite pour répondre à ces limites. Elle repose sur un principe simple : interroger dynamiquement une source de connaissance externe afin de fournir au modèle de langage un contexte pertinent avant la génération. En combinant un module de récupération d'information avec un modèle génératif, cette approche permet d'améliorer la qualité et la pertinence des réponses générées.

Le modèle RAG original [Lewis *et al.*, 2020] procède en deux étapes : (1) récupération des documents les plus pertinents pour une requête donnée via un index dense, puis (2) génération d'une réponse par un modèle de langage conditionné sur ces documents.

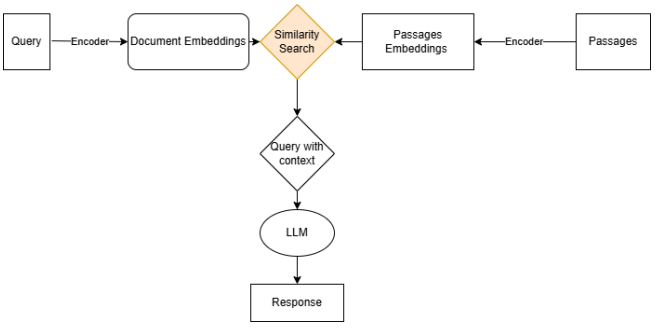

Figure 1: Pipeline du RAG

Plusieurs variantes du cadre RAG ont récemment été proposées pour améliorer spécifiquement la qualité de la phase de récupération.

Nous comparons ici expérimentalement deux variantes de RAG (HyDE et RAG Fusion) dans le cadre de systèmes de question-réponse.

2 Approche théorique

- **HyDE** (Hypothetical Document Embeddings) [Gao *et al.*, 2022] génère un texte hypothétique à partir de la question. Considéré comme une réponse idéale, il est ensuite utilisé comme nouvelle requête pour interroger la base de connaissances.

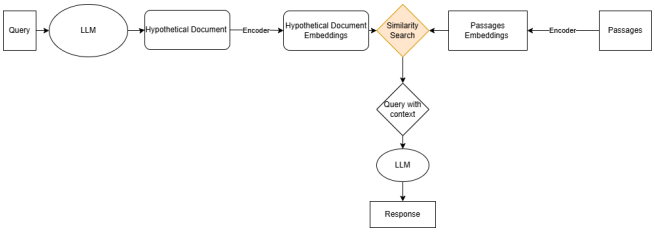

Figure 2: Pipeline de la méthode HyDE

Cette méthode cherche à mieux identifier les passages pertinents en s'appuyant sur le fait qu'une réponse est souvent plus proche sur le plan sémantique des documents visés que la question elle-même.

- **RAG Fusion** [Rackauck, 2024] génère plusieurs reformulations d'une même question, puis utilise chacune d'elles de manière indépendante pour récupérer des documents à partir du corpus. Les documents récupérés pour chaque reformulation sont ensuite fusionnés avant la phase de génération de réponse.

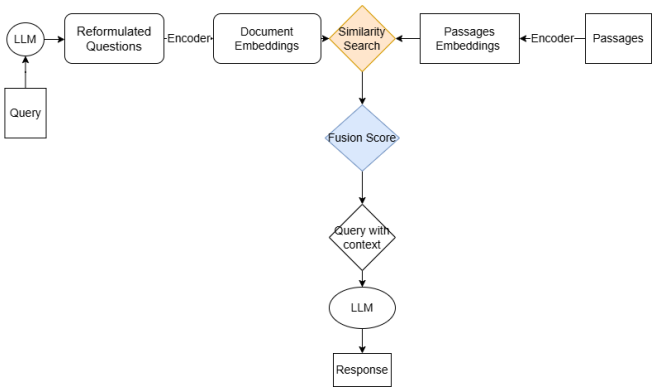

Figure 3: Pipeline de la méthode RAG Fusion

Cette stratégie vise à maximiser la couverture sémantique en explorant différentes perspectives de la requête initiale, réduisant ainsi le biais introduit par une formulation unique.

La fusion des résultats utilise la méthode *Reciprocal Rank Fusion* (RRF), qui combine les classements des documents issus de différentes requêtes. Pour un document donné d, son score fusionné $S(d)$ est calculé comme suit :

$$S(d) = \sum_{q \in Q} \frac{1}{k + \text{rank}_q(d)} \tag{1}$$

où :

- Q est l'ensemble des requêtes reformulées,
- $\text{rank}_q(d)$ est le rang du document d dans les résultats de la requête q (ou ∞ si d n'est pas présent),
- k est une constante de régularisation (typiquement $k = 60$) pour atténuer l'impact des documents moins bien classés.

Cette approche favorise les documents apparaissant fréquemment et en bonne position dans les différents classements, prenant ainsi en compte les indices de pertinence tout en atténuant les bruits spécifiques liés à certaines reformulations.

3 Expérimentation

3.1 Dataset

Pour cette étude, nous avons utilisé le jeu de donnée RAG Mini BioASQ. Le jeu de données RAG Mini BioASQ est une version réduite du corpus BioASQ, pensée pour des expériences rapides de génération augmentée par la recherche (RAG) dans le domaine biomédical. Il regroupe environ 2 700 exemples, chacun composé d'une question scientifique précise, d'une réponse courte, et d'identifiants de passages pertinents tirés d'articles biomédicaux. Ces passages proviennent d'un corpus plus large d'environ 13300 documents issus de la base PubMed, incluant des études de cas, des descriptions de mécanismes biologiques, ou encore des résultats de recherche sur différents modèles expérimentaux. Ce jeu de données associe donc les questions et les réponses à leurs extraits de texte originaux, ce qui permet de tester la capacité des modèles à retrouver l'information dans les documents et à formuler des réponses précises.

Question	**Réponse**	**Passages pertinents (ID)**
La protéine Papilin est-elle sécrétée ?	Oui, Papilin est une protéine sécrétée.	[2178,19297,150941]

Table 1: Exemple d'entrée du jeu de données RAG Mini BioASQ

3.2 Approche RAG Simple

Pour la phase de récupération, nous avons choisi d'utiliser le modèle *BGE-base*, reconnu pour ses performances en récupération de passages dans des configurations RAG. De plus, nous utilisons l'indexeur *FAISS*, qui permet une recherche rapide de passages à partir d'embeddings, particulièrement utilisé pour ce genre de tâches de retrieval.

Pour la génération de réponses, nous avons opté pour *Mistral-7B-Instruct*, un modèle de taille intermédiaire adapté aux tâches instructionnelles. Il combine des performances solides avec une accessibilité en open source et une intégration simple, ce qui en a fait un choix pertinent dans le cadre de notre pipeline.

Pour les expérimentations, nous avons exploré la possibilité d'affiner l'encodeur, en adoptant la méthode de l'apprentissage contrastif. L'objectif était d'adapter un modèle de représentation des textes afin qu'il apprenne à rapprocher dans l'espace vectoriel les couples (requête, passage pertinent), tout en éloignant les passages non pertinents. Cette idée dérive de l'approche Dense Passage Retrieval for open domain question answering" [Karpukhin *et al.*, 2020].

$$L(q_i, p_i^+, p_{i,1}^-, \ldots, p_{i,n}^-) = -\log \frac{e^{\text{sim}(q_i, p_i^+)}}{e^{\text{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(q_i, p_{i,j}^-)}} \tag{2}$$

Figure 4: Fonction de perte constrastive rapprochant des paires positives (*requête* q_i et *passage pertinent* p_i^+) et éloignant des paires négatives (*requête* q_i et n *passages non pertinents* $p_{i,j}^-$) telle que décrite dans DPR

Pour ce faire, nous avons constitué des triplets (requête, passage positif, passage négatif). Les passages positifs sont ceux qui répondent effectivement à la question, tandis que les passages négatifs utilisés sont des Hard Negatives : ce sont des passages qui apparaissent proches de la requête dans l'espace sémantique, mais qui ne répondent pas à la question. Ce choix rend l'apprentissage plus difficile mais également plus efficace, car il force le modèle à apprendre des distinctions fines entre des passages lexicalement ou contextuelle-

ment proches. Cette approche visait à améliorer la capacité du modèle à encoder les requêtes et les documents de manière à faciliter le matching lors de la recherche de passages pertinents.

Par ailleurs, il faut noter que le modèle de génération n'a pas été utilisé notre évaluation initiale. En effet, comme les deux approches comparées n'intervenaient qu'au niveau de la phase de récupération et que le jeu de données fournissait directement les passages pertinents associés à chaque question, il était possible d'évaluer la qualité du retrieval à l'aide de métriques classiques telles que la précision@k, le rappel@k et le score F1@k. Ici, k désigne le nombre de documents les mieux classés considérés pour le calcul de ces métriques : par exemple, la précision@3 mesure la proportion de passages pertinents parmi les trois premiers résultats retournés par le système.

Avec cette approche RAG simple, nous avons obtenu les résultats suivants :

k	Precision@k	Recall@k	F1@k
1	0.870000	0.326167	0.467800
2	0.679000	0.495167	0.562867
3	0.582667	0.629333	0.594533
4	0.518500	0.738833	0.599119
5	0.455600	0.805167	0.572635

Table 2: Métriques de récupération en fonction de k (RAG Simple)

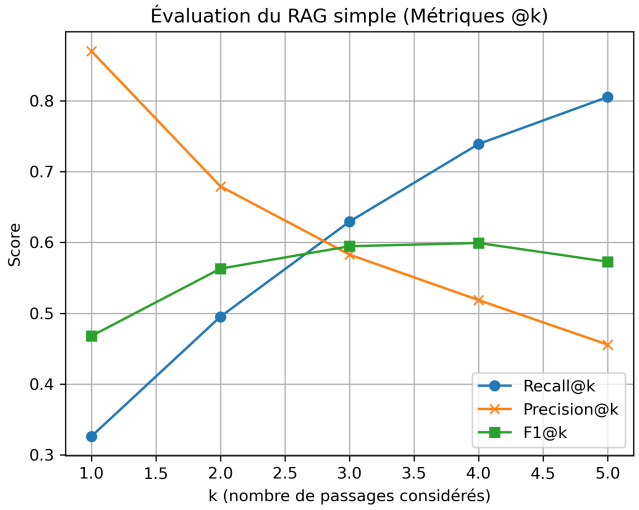

Figure 5: Precision, Recall et F1 Score en fonction du nombre de passages (RAG Simple)

On constate que la phase de récupération est déjà particulièrement efficace avec l'approche *RAG* simple. La précision est élevée à $k = 1$, ce qui suggère que le passage le plus pertinent est souvent correctement identifié. Le rappel progresse fortement avec k, atteignant plus de 80 % à $k = 5$, indiquant que les passages pertinents figurent généralement dans les premiers résultats. Pour $k = 3$, le score F1 atteint une valeur approximative de 0,60, ce qui montre également un bon équilibre entre précision et rappel.

3.3 HyDE

Nous avons choisi d'utiliser *BioBERT* comme modèle génératif pour produire les documents hypothétiques à partir des questions. Ce modèle pré-entraîné sur des textes biomédicaux nous paraissait bien aligné avec le style et le contenu du corpus, pouvant donc favoriser la génération de documents hypothétiques plus proches des passages réellement pertinents.

Il est important de préciser que l'encodeur utilisé ici n'a pas été affiné selon la méthode DPR, comme mentionné précédemment. En effet, dans le cadre de HyDE, le retrieval s'appuie sur la mesure de la similarité entre réponses hypothétiques et passages du corpus, et non entre questions et réponses conformément à l'objectif notre affinage initial.

Avec cette l'approche HyDE, nous avons obtenu les résultats suivants :

k	Precision@k	Recall@k	F1@k
1	0.644000	0.241667	0.346467
2	0.507000	0.369333	0.419800
3	0.426000	0.460167	0.434552
4	0.382000	0.545667	0.441619
5	0.338000	0.599667	0.425206

Table 3: Métriques de récupération en fonction de k (HyDE)

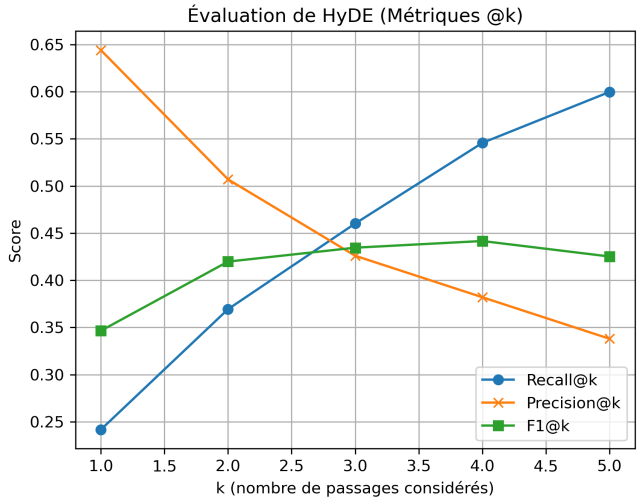

Figure 6: Précision, Recall et Score F1 en fonction du nombre de passages (HyDE)

Les résultats obtenus avec HyDE sont significativement inférieurs à ceux du RAG simple. La précision chute rapidement avec l'augmentation de k, et reste globalement basse, indiquant que les documents récupérés sont beaucoup moins pertinents. Bien que le rappel progresse comme attendu avec k, celui-ci reste également faible tout comme le score F1

Des tentatives d'amélioration ont notamment été explorées. -Nous avons expérimenté un affinage de l'encodeur sur les

réponses générées par HyDE, afin d'améliorer la correspondance entre les documents générés et les passages réellement présents dans le corpus. Toutefois, cela n'a pas conduit à une amélioration des performances de récupération, les scores de précision, de rappel et de F1 restant relativement similaires à ceux présentés ci-dessus.

-Nous avons aussi essayé d'utiliser un meilleur encodeur *sentence-transformers/all-mpnet-base-v2* et un meilleur générateur *EleutherAI/gpt-neo-2.7B* nous avons obtenu une légère amélioration mais tout de même négligable.

3.4 RAG Fusion

Nous avons choisi d'utiliser le modèle *T5 Paraphraser* pour générer les reformulations de questions dans le cadre de cette approche. Entraîné spécifiquement pour la paraphrase, ce modèle nous semblait suffisant et permet de produire des variantes lexicales et syntaxiques relativement diversifiées tout en conservant le sens de la question initiale.

Avec cette l'approche RAG-Fusion, nous avons obtenu les résultats suivants :

k	Precision@k	Recall@k	F1@k
1	0.862000	0.322000	0.462400
2	0.669000	0.485333	0.553000
3	0.576667	0.620167	0.587410
4	0.524500	0.743167	0.604833
5	0.475200	0.838333	0.597024

Table 4: Précision, Recall et Score F1 en fonction du nombre de passages (RAG Fusion)

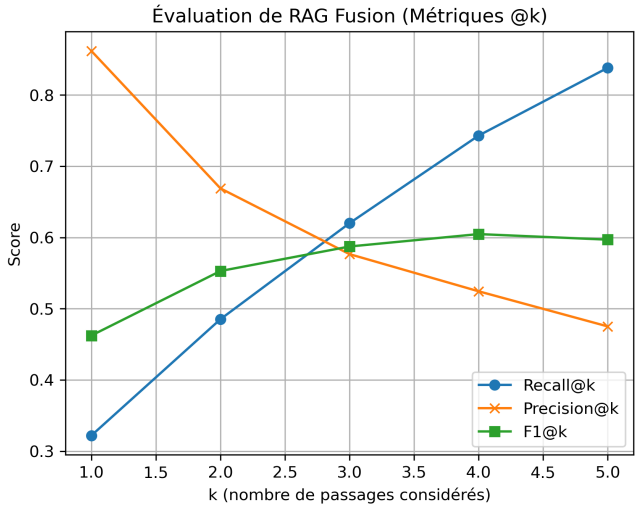

Figure 7: Précision, Recall et Score F1 en fonction du nombre de passages (RAG Fusion)

Les résultats obtenus avec RAG Fusion sont globalement comparables à ceux du RAG simple, bien que légèrement moins bons. La stratégie de fusion ne semble pas apporter de gain significatif par rapport à l'approche simple, malgré l'ajout de reformulations. Contrairement à HyDE, elle a toutefois le mérite de préserver la stabilité des performances sans réellement nuire à la qualité du retrieval.

RAG Fusion avec Reranker

Des tentatives d'amélioration ont aussi été explorées.

Nous avons remplacé la méthode de fusion RRF par un **reranker dense** entraîné spécifiquement sur des paires *(question, passage)*.

Le **reranker** utilisé repose sur un modèle de type *Cross-Encoder* comme présenté par [Nogueira *et al.*, 2019].Il prend en entrée la concaténation d'une question et d'un passage sous la forme suivante :

[CLS] question [SEP] passage [SEP]

Cette séquence est encodée conjointement par un modèle Transformer (type BERT), puis le vecteur associé au token `[CLS]` est projeté pour produire un score de pertinence unique.

Pour entraîner ce modèle, nous avons utilisé notre jeu de données présenté en subsection 3.1. Pour chaque question, nous avons construit des paires *(question, passage)* de la manière suivante :

- un label 1.0 est attribué aux paires *(question, passage pertinent)*, c'est-à-dire ceux explicitement associés à la question dans les annotations du jeu de données,
- un label 0.0 est attribué aux paires *(question, passage nonpertinent)*. Les passages non-pertinents sont sélectionnés aléatoirement parmi les autres passages du corpus. Afin de garantir un équilibre suffisant entre positifs et négatifs, nous avons échantillonné jusqu'à quatre passages non pertinents (*neg_ratio* = 4) pour chaque passage pertinent.

Le modèle est alors affiné (*fine-tuned*) en utilisant une **loss de régression** classique, la *Mean Squared Error* (MSE Loss), qui minimise l'écart entre le score prédit $f(q, p)$ et l'étiquette cible y. La fonction de perte est définie comme suit :

$$\mathcal{L}_{\text{MSE}} = (f(q, p) - y)^2$$

où :

- $f(q, p)$ est le score prédit pour la paire (*question, passage*),
- $y \in \{0.0, 1.0\}$ est l'étiquette cible.

Cette approche pousse le modèle à attribuer un score proche de 1 aux passages pertinents et un score proche de 0 aux passages non pertinents, sans imposer directement un écart fixe entre eux comme le ferait une Margin Ranking Loss.

Cette approche produisait de meilleurs scores comme on peut le voir dans le tableau 5, mais l'amélioration observée s'expliquait essentiellement par la qualité du reranker lui-même, et non par la stratégie de fusion des reformulations. Le gain ne pouvait donc pas être attribué au mécanisme RAG Fusion en tant que tel.

Par ailleurs, nous avons aussi testé un autre modèle de reformulation de type T5, basé sur du *prompting* libre (non

spécialisé dans la paraphrase). Cependant, les résultats ont été détériorés en raison de la génération de variantes trop diversifiées, perdant parfois le sens de la requête initiale.

3.5 Récapitulatif des résultats

Le tableau suivant présente les F1-scores obtenus en fonction de k pour RAG Simple, HyDE, RAG Fusion et RAG-Fusion with Reranker.

Approche	F1@1	F1@2	F1@3	F1@4	F1@5
RAG Simple	0.4678	0.5629	0.5945	0.5991	0.5726
HyDE	0.3465	0.4198	0.4346	0.4416	0.4252
RAG Fusion	0.4624	0.5530	0.5874	0.6048	0.5970
RAG Fusion with Reranker	**0.508**	**0.577**	**0.602**	**0.618**	**0.613**

Table 5: F1-score pour chaque approche en fonction de k

3.6 Discussion des résultats

Malgré les promesses théoriques de HyDE et de RAG Fusion pour améliorer la qualité du retrieval dans les architectures RAG, nos résultats expérimentaux montrent que ces méthodes n'ont pas surpassé, et ont plutôt sous-performé par rapport à l'approche RAG simple. Ces résultats peuvent s'expliquer par plusieurs facteurs liés aux caractéristiques propres à chaque méthode, mais aussi à celles du jeu de données utilisé.

Dans le cas de HyDE, l'efficacité dépend fortement de la qualité des documents hypothétiques générés à partir des questions. Si la réponse simulée est trop générique, trop brève ou formulée dans un style qui diffère de celui des documents indexés, elle peut se révéler peu efficace pout la récupération. De plus, étant donné la technicité du jeu de données, le générateur utilisé (BioBERT) ne semblait pas disposer de connaissances suffisantes pour produire des réponses réellement factuelles, ce qui a aussi contribué à la dégradation des performances. Dans ce contexte, HyDE ajoute une couche de bruit inutile, ce qui explique cette chute des scores.

Du côté de RAG Fusion, l'approche repose sur la diversité sémantique entre les reformulations. Or, dans nos expérimentations, celles générées par T5 Paraphraser étaient parfois trop proches de la question initiale, ce qui limite l'intérêt de la fusion car les résultats fusionnés par RRF étaient en grande partie redondants. À l'inverse, des essais menés avec un modèle T5 plus libre via prompting ont conduit à des reformulations trop variées, et donc souvent trop éloignées de l'intention de la question. Ces résultats illustrent la difficulté à trouver un juste équilibre concernant la diversité des reformulations, sans quoi RAG Fusion ne parvient peut être pas à apporter une réelle valeur ajoutée.

3.7 Expérimentations sur un autre jeu de données

Les jeux de données disponibles publiquement pour l'évaluation des systèmes RAG sont très rares. Nous avons toutefois mené des expérimentations complémentaires sur le jeu de données RAG Mini Wikipedia, une version allégée d'un corpus Wikipédia, conçue pour tester des approches de RAG dans un contexte généraliste. Ce jeu propose des paires question-réponse accompagnées de passages issus d'articles encyclopédiques. Cependant, à la différence de RAG Mini BioASQ, il ne fournit pas les identifiants des passages considérés comme pertinents, ce qui empêche une évaluation de la phase de récupération en termes de précision ou de rappel. L'évaluation des performances a donc été menée exclusivement sur la qualité des réponses générées (avec le modèle *Mistral-7B-Instruct*), à l'aide du score BLEU.

Question	Réponse
How many long was Lincoln's formal education?	18 months

Table 6: Exemple d'entrée du jeu de données RAG Mini Wikipedia

En mettant en place des pipeline similaires et suite au passage à travers notre modèle génératif, nous avons obtenu les résultats suivants :

Méthode	BLEU-1	BLEU-2
Générateur seul	0.299	0.297
RAG Simple	0.580	0.571
HyDE	0.564	0.554
RAG Fusion	0.550	0.538

Table 7: Scores BLEU-1 et BLEU-2 obtenus pour chaque méthode.

Sur ce second jeu de données, les résultats montrent aussi que HyDE et RAG Fusion n'améliorent pas les scores obtenus par l'approche RAG simple. Toutefois, contrairement au cas précédent, HyDE obtient ici de meilleurs résultats que RAG Fusion, avec des scores *BLEU* légèrement supérieurs. Cela peut s'expliquer par la nature même des questions posées dans ce corpus : elles sont courtes, explicites et simples, ce qui favorise l'efficacité d'une génération hypothétique de réponse. À l'inverse, la diversité introduite par les reformulations de RAG Fusion semble moins utile, dans un contexte où la formulation initiale est déjà suffisante et claire. Ainsi, ces résultats illustrent que les performances des méthodes avancées dépendent fortement du type de données utilisé. Il demeure donc difficile, dans notre cadre expérimental, d'identifier et de trouver un jeu de données sur lequel HyDE ou RAG Fusion surpassent significativement l'approche RAG simple, et d'établir une comparaison fiable de leurs performances respectives.

3.8 Lien vers le dépôt GitHub

Notre travail est disponible sur GitHub à l'adresse suivante : https://github.com/mohamedali05/Projet-INF8225

4 Analyse Critique

Dans ce projet, nous avons choisi d'explorer des techniques avancées de génération augmentée par récupération (RAG), avec l'objectif de dépasser les performances standard du RAG simple. Nous nous sommes concentrés sur deux variantes aux logiques assez opposées : HyDE, qui consiste à générer un

document hypothétique à partir de la question afin de guider la récupération, et RAG Fusion, qui combine les résultats de plusieurs reformulations indépendantes de la même requête. Notre intention était de mieux comprendre ces approches en les implémentant, en les adaptant à des jeux de données existants et en évaluant leurs performances de manière empirique.

Cependant, ce travail nous a rapidement confrontés à des limites pratiques. D'abord, il existe très peu de jeux de données adaptés à l'évaluation de la phase de récupération, avec des passages pertinents clairement identifiés. Ensuite, les performances observées sur les deux méthodes ont souvent été décevantes et difficiles à interpréter. Il a ainsi été complexe de comparer objectivement les apports de chaque approche, faute de gains significatifs.

Malgré ce manque de résultats, nous avons choisi de creuser davantage, en affinant nos expérimentations. Cette démarche nous a amenés à nous interroger davantage sur les conditions de réussite de ces approches, au-delà de leur formulation théorique. Nous avons ainsi compris que des éléments comme la structure du corpus, la qualité de la génération, ou la cohérence entre les représentations sémantiques sont des facteurs déterminants dans l'efficacité réelle d'une méthode. Finalement, ce travail nous a appris que les approches RAG avancées ne sont pas toujours meilleures, mais qu'elles doivent être sélectionnées et ajustées en fonction du contexte.

References

[Gao *et al.*, 2022] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*, 2022.

[Karpukhin *et al.*, 2020] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[Lewis *et al.*, 2020] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, and Vladimir Karpukhin. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.

[Nogueira *et al.*, 2019] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019.

[Rackauck, 2024] Zackary Rackauck. Rag-fusion: A new take on retrieval-augmented generation. *arXiv preprint arXiv:2003.01897*, 2024.