Vers des LLMs fiables pour des tâches recherche d'information

Hanane Djeddal, Pierre Erbacher, Laure Soulier, Karen Pinel-Sauvagnat, Sophia Katrenko, Lynda Tamine

Vers des LLMs fiables pour des tâches recherche d'information

Hanane Djeddal[1], Pièrre Erbacher[2], Laure Soulier[2], Karen Pinel-Sauvagnat[1], Sophia Katrenko[3] and Lynda Tamine[1]

[1]*Université Paul Sabatier, IRIT, Toulouse, France*
[2]*Sorbonne Université, CNRS, ISIR, F-75005 Paris, France*
[3]*ECOVADIS*

Abstract

Les grands modèles de langage (LLMs) ont émergé comme un outil largement utilisé pour la recherche d'information, mais il reste difficile pour l'utilisateur de détecter si leur réponse est fiable ou hallucinée. Dans ce travail préliminaire, notre objectif est de permettre aux LLMs de répondre à une requête de façon attribuable, améliorant ainsi leur factualité et leur vérifiabilité. Les travaux existants reposent principalement sur les moteurs de recherche commerciaux et l'évaluation humaine, ce qui rend difficile la reproduction et la comparaison des différentes approches de modélisation. Nous proposons un modèle multi-phases et ouvert qui permet de restituer des documents supports et générer des réponses avec citations. De plus, nous explorons différents scénarios pour évaluer notre modèle et ses divers modules de manière automatique.

Keywords

Recherche d'information, Grands modèles de langage, QA Attributable, Génération augmentée par la recherche d'information

1. Introduction

Le potentiel des grands modèles de langage (LLMs) est de plus en plus reconnu dans les scénarios de recherche d'information. Leurs réponses cohérentes et accessibles permettent un accès direct et facile à l'information, et ils deviennent l'outil de recherche principal de nombreux utilisateurs. Cependant, ils génèrent parfois du texte contenant des informations non-factuelles, aussi nommées hallucinations [1]. Cela a conduit à l'émergence d'un nouveau paradigme de génération défini dans Bonhet et al. [2] comme *attributed Question Answering* , ou dans [3] comme *self-supported QA* : l'idée est que les LLMs doivent fournir des citations vers un ou plusieurs passages support pour tout texte qu'ils génèrent. L'incorporation de citations présente plusieurs avantages : (1) les utilisateurs peuvent facilement vérifier la source des informations générées avec les citations fournies ; (2) les LLMs peuvent produire du texte qui suit fidèlement les passages cités, ce qui pourrait améliorer la factualité et réduire les hallucinations.

Conférence en Recherche d'Information et Applications (CORIA) 2024, 3 – 4 avril, La Rochelle, France
hanane.djeddal@irit.fr (H. Djeddal); pierre.erbacher@isir.upmc.fr (P. Erbacher); laure.soulier@isir.upmc.fr (L. Soulier); karen.sauvagnat@irit.fr (K. Pinel-Sauvagnat); skatrenko@ecovadis.com (S. Katrenko); Lynda.Lechani@irit.fr (L. Tamine)

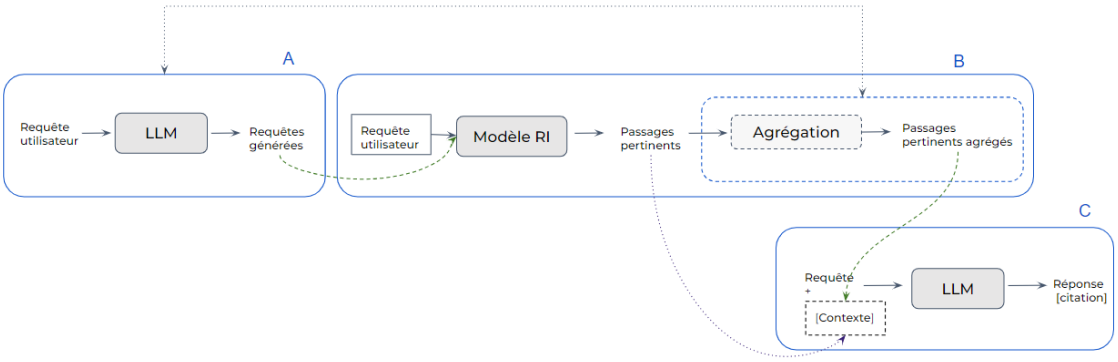

Figure 1: Vue globale du pipeline . Cadre A: Module de génération de requêtes. Cadre B: Module de recherche d'information. Cadre C: Module de génération de réponse.

Plusieurs systèmes commerciaux, tels que Bing Chat [1] et perplexity.ai [2] ont adopté ce paradigme et répondent aux requêtes des utilisateurs en langage naturel avec des références à des pages Web [1]. Les auteurs de [3] et [4] ont proposé des systèmes similaires, mais ils expérimentent principalement avec des moteurs de recherche commerciaux et des modèles propriétaires, ce qui rend leurs résultats difficiles à reproduire et à évaluer [1].

Les systèmes de type RAG ("Retrievial Augmented Generation") [5], [6] [7] proposent des LLMs augmentés par la recherche d'information qui renforcent la mémoire paramétrique des LMs avec une mémoire non paramétrique en incorporant des passages support à la fois pendant l'entraînement et l'inférence, mais ils ne garantissent pas la fidélité aux passages récupérés ni ne fournissent explicitement de citations.

Le travail préliminaire présenté dans cet article vise à concevoir un modèle capable de générer des réponses fiables et factuelles à des requêtes utilisateur. Notre modèle est caractérisé par ce qui suit :

- *Une approche de génération augmentée par la recherche d'information*, plus connue avec l'expression anglissime "*Retrieval Augmented Generation RAG*" [5], [7]: pour répondre à l'objectif de la factualité, la génération de la réponse est contrainte par des passages supports (mentionné aussi dans cet article comme *documents supports* ou *le contexte*) sélectionnées en entrée. La sélection de ces passages est issue de l'évaluation de la requête initiale ainsi que d'un ensemble de requêtes générées par un LLM ; cette génération de requêtes est motivée par l'objectif de rappeler des documents qui couvrent au mieux les différents aspects de la requête utilisateur.
- *La génération jointe de la réponse et des citations* : nous entraînons le modèle à générer des réponses tout en citant des documents sources permettant d'attribuer ses réponses.

Dans ce qui suit, nous présentons le protocole expérimental suivi pour atteindre ces objectifs ainsi que les résultats préliminaires.

[1]https://www.bing.com/new
[2]https://www.perplexity.ai/

Table 1
Exemple de différentes sorties du pipeline

Requête utilisateur	Requêtes générées (Module A)	Passages support (Module B)	Réponse finale avec citation (Module C)
How old is the oldest pyramid?	1. List of ancient Egyptian pyramids and their construction dates 2. Pyramids of Giza: age and history 3. Pyramid of Djoser: age and significance 4. Pyramid of Djineh: oldest known pyramid and its historical context	[1] In April 2010, archaeologists discovered the 2,700-year-old tomb of a dignitary within Mound 11 that is the oldest pyramidal tomb yet discovered in Mesoamerica. [2] The earliest known Egyptian pyramids are found at Saqqara, northwest of Memphis. The earliest among these is the Pyramid of Djoser, which was built 2630–2610 BC during the Third Dynasty. This pyramid and its surrounding complex.... [3] At the main temple of Tlatelolco, archeologists recently discovered a pyramid within the visible temple....	The oldest pyramid is the Pyramid of Djoser, which was built around 2630-2610 BC during the Third Dynasty in Saqqara, Egypt. This pyramid and its surrounding complex are generally considered to be the world's oldest monumental structures constructed of dressed masonry. [2]

2. Aperçu du modèle

La figure 1 donne une vue globale du modèle de génération que nous proposons. Il est composé de trois modules :

- **Génération de requêtes** : ce module (cadre A de la figure 1) prend en entrée la requête de l'utilisateur et génère en sortie une liste de requêtes similaires permettant de couvrir d'autres aspects de son sujet.
- **Recherche d'information** : l'objectif de ce module (cadre B de la figure 1) est de retrouver les passages supports dans le corpus en utilisant la requête de l'utilisateur ainsi que les requêtes générées par le module A. Une fois que nous disposons d'une liste ordonnée de documents pour chaque requête, nous testons différentes méthodes d'agrégation afin d'obtenir une liste finale de documents supports.
- **Génération de réponse** : ce module (cadre C de la figure 1) prend en entrée la requête de l'utilisateur ainsi que les passages supports et génère une réponse avec citations.

Le tableau 1 montre un exemple de génération avec les sorties intermédiaires de chacun des module cités plus haut.

3. Protocole Expérimental

3.1. Collections de test

Nous utilisons deux collections de test pour évaluer les différentes composantes du pipeline, à savoir (1) la pertinence des documents/passages récupérés (cadre B) ; (2) la qualité de la réponse générée et (3) la qualité des citations incluses dans la réponse générée :

- le corpus MIRACL [8] est utilisé pour le module de RI

- le corpus HAGRID [9] est utilisé pour évaluer la génération de réponses avec citations. Dans cette collection de test, les réponses aux requêtes sont générées avec un LLM (GPT-3.5 turbo) et des annotateurs humains évaluent les réponses du LLM selon deux critères: l'informativité et l'*attribuabilité*.

3.2. Métriques d'évaluation

Nous évaluons notre système à plusieurs niveaux : en évaluant la réponse générée finale mais aussi en considérant les sorties intermédiaires des différents modules. La qualité des requêtes générées est indirectement évaluée par son impact sur les étapes suivantes du pipeline, notamment la pertinence des passages restitués. Cette dernière est évaluée en utilisant les métriques de la RI classiques: précision/rappel @k et nDCG @k.
Finalement, nous évaluons différents aspects de la réponse finale :

- **Exactitude** : il s'agit de savoir si la réponse retournée est correcte par rapport à la question posée. Pour cela, nous la comparons à la réponse de référence fournie dans la collection de test en utilisant des métriques classiques d'évaluation de texte telles que ROUGE [10] et BERTScore [11].
- **Factualité** : pour mesurer à quel point la réponse est fidèle au contexte fourni, nous utilisons la métrique AUTOAIS définie dans [12] qui utilise un modèle de reconnaissance d'implications textuelles (RTE) pour mesurer un score d'implication entre la réponse et les passages supports.
- **Citation** : de manière similaire à AUTOAIS, nous suivons l'approche de T. Gao et al. [1], qui se base sur les modèles RTE et proposent un cadre d'évaluation de réponses avec citation en mesurant un rappel et précision associés aux citations.

3.3. Configurations du modèle

3.3.1. Génération de requêtes

Nous utilisons l'implémentions huggingface[3] du modèle Zephyr [13] pour la génération des requêtes. Nous testons deux configurations différentes :

- *Prompting à zéro exemple (zero-shot)* : dans cette configuration nous donnons pour instruction au modèle de générer des requêtes similaires à la requête d'entrée. Dans ce qui suit, les modèles avec la notation **'AZS'** (module A ZeroShot) indiquent que les requêtes utilisées proviennent de ce scénario.
- *Prompting avec quelques exemples (few-shot)* : cette configuration est identique à celle d'avant, mais nous rajoutons quelques exemples au prompt de ce que nous attendons du modèle. Pour ce scénario, nous utilisons la notation **'AFS'** (module A FewShot).

Dans les deux configurations, nous précisons au modèle de générer un maximum de 4 requêtes.

[3] https://huggingface.co/

3.3.2. Recherche d'information

Nous utilisons une combinaison de BM25 [14] avec MonoT5 [15] pour la restitution des passages pertinents.

- Module B **(Mod-B)** : dans cette configuration, seulement la requête utilisateur est utilisée pour récupérer les passages support.
- Module B et Module A : les requêtes issues de module A, en plus de la requête utilisateur sont utilisées. Pour chaque requête, nous récupérons la liste ordonnée des documents pertinents. Pour agréger ces listes de documents, nous testons différentes méthodes :
  - Re-ordonnancement **(rerank)** : les documents sont réordonnés avec MonoT5 en les comparant à la requête utilisateur.
  - Vote : Nous appliquons une fonction de vote pour sélectionner les documents pertinents à partir des listes initiales. Il s'agit de compter le nombre de fois un document est retourné par une sous-requêtes dans ses top 10 documents pertinents.

3.3.3. Génération de réponse avec citation

Nous utilisons le même modèle Zephyr en zero-shot pour la génération de la réponse avec un prompt adapté. Pour les documents support fournis au modèle, nous testons différentes combinaisons:

- *Sans passages support* : le modèle génère la réponse en utilisant la requête utilisateur seulement, sans passages support. L'objectif est de tester la capacité du modèle à répondre à la question en se basant uniquement sur sa mémoire paramétrique.
- *Passages support - vérité terrain* : le modèle prend en entrée la requête utilisateur et les passages support de la vérité terrain. Ce sont les passages annotés comme pertinents dans la collection de test.
- *Passages support - Module B* : dans ce cas, les passages support sont issus du module de la recherche d'information (module B) mais en utilisant seulement la requête utilisateur pour les restituer (ligne 1 (Mod-B) dans la table 2).
- Passages support - Module A puis B : les passages support sont issus du module de la recherche d'information (module B) en utilisant la requête utilisateur ainsi que les requêtes générées par le module A (ligne 4 (Mod-AFS puis B-rerank) dans la table 2). Cela correspond à une exécution des différents modules du modèle.

4. Résultats Préliminaires

Des expérimentations sont en cours pour étudier et améliorer chaque module du modèle. Les tableaux 2 et 3 montrent les résultats préliminaires de l'évaluation du module de recherche d'information et le module de génération de texte respectivement.

Ces résultats préliminaires nous permettent d'identifier les améliorations possibles à apporter. Par exemple, dans le tableau 2, on remarque que l'ajout des requêtes générées n'implique pas forcément une amélioration dans la pertinence des documents restitués mais il a un impact

Table 2
Évaluation de la pertinence des passages support (Module B). En utilisant la requête utilisateur (**Mod-B**). En utilisant la requête utilisateur et les requêtes générées par le module A en zero-shot avec une agrégation rerank (**Mod-AZS puis B-rerank**) ou avec une fonction de vote **(Mod-AZS puis B-vote**). En utilisant des requêtes générées en few shot avec une agrégation rerank (**Mod-AFS puis B-rerank**)

Modèle	@1		@3		@5		@10		
	Précision	rappel	Précision	rappel	Précision	rappel	Précision	rappel	nDCG
Mod-B	**0.480447**	0.232883	**0.331006**	0.431305	**0.259497**	**0.531478**	**0.164525**	0.633107	**0.5321**
Mod-AZS puis B-rerank	0.47067	0.228192	0.329609	**0.436017**	0.255587	0.530081	0.162989	**0.639379**	0.5311
Mod-AZS puis B-vote	0.423184	0.201937	0.304469	0.398944	0.240503	0.489447	0.162151	0.615254	0.4988
Mod-AFS puis B-rerank	0.472067	**0.233077**	0.328212	0.433875	0.254749	0.527947	0.162709	0.636347	0.5301

Table 3
Évaluation de l'exactitude de la réponse généré (Module C). En variant les passages support: Sans passages support (**Sans**). En utilisant les passages support de la vérité terrain (**vérité terrain**). En utilisant les passages support récupérés par la requête utilisateur seulement (**sortie de Mod B**). En utilisant les passages support issus de la configuration *Mod-AFS-B-rerank* (**sortie de Mod AFS puis B rerank**). Meilleures performances en gras, et deuxième meilleures souligné.

Passage support	Modèle	Rouge-L			BertScore		
		Precision	Recall	F-score	Precision	Recall	F-score
Sans	Zephyr	<u>0.294756</u>	0.428566	0.284193	<u>0.875352</u>	0.888833	0.881262
Vérité terrain		**0.352949**	**0.603769**	**0.383775**	**0.886766**	**0.919257**	**0.902015**
sortie de Mod-B		0.281629	0.515091	0.301436	0.869459	0.899492	0.883532
sortie de Mod-AFS puis B-rerank		0.294583	<u>0.525681</u>	<u>0.314097</u>	0.870915	<u>0.901127</u>	<u>0.88509</u>

sur la génération finale dans le tableau 3, où la génération de la réponse finale à partir de la configuration *sortie de Mod-AFS puis B-rerank* est meilleure que la génération en utlisant la configuration *sortie de Mod-B*. Cela nous motive à examiner davantage la façon dont on agrège les documents et d'explorer différentes méthodes d'agrégation.

5. Conclusion

Dans la suite de notre travail, nous améliorerons les différents modules : explorer différentes méthodes d'agrégation dans le module B, améliorer la qualité des requêtes générées, améliorer la génération de la réponse finale. En plus, nous allons pallier le manque de collections adaptées à l'évaluation de notre tâche en étendant des collections de test existantes.

References

[1] T. Gao, H. Yen, J. Yu, D. Chen, Enabling large language models to generate text with citations, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 6465–6488. URL: https://aclanthology.org/2023.emnlp-main.398. doi:10.18653/v1/2023.emnlp-main.398.

[2] B. Bohnet, V. Q. Tran, P. Verga, R. Aharoni, D. Andor, L. B. Soares, J. Eisenstein, K. Ganchev, J. Herzig, K. Hui, T. Kwiatkowski, J. Ma, J. Ni, T. Schuster, W. W. Cohen, M. Collins, D. Das, D. Metzler, S. Petrov, K. Webster, Attributed question answering: Evaluation and modeling for attributed large language models, CoRR abs/2212.08037 (2022). URL: https://doi.org/10.48550/arXiv.2212.08037. doi:10.48550/ARXIV.2212.08037. arXiv:2212.08037.

[3] J. Menick, M. Trebacz, V. Mikulik, J. Aslanides, H. F. Song, M. J. Chadwick, M. Glaese, S. Young, L. Campbell-Gillingham, G. Irving, N. McAleese, Teaching language models to support answers with verified quotes, CoRR abs/2203.11147 (2022). URL: https://doi.org/10.48550/arXiv.2203.11147. doi:10.48550/ARXIV.2203.11147. arXiv:2203.11147.

[4] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, J. Schulman, Webgpt: Browser-assisted question-answering with human feedback, CoRR abs/2112.09332 (2021). URL: https://arxiv.org/abs/2112.09332. arXiv:2112.09332.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., ????, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[6] D. Yu, C. Zhu, Y. Fang, W. Yu, S. Wang, Y. Xu, X. Ren, Y. Yang, M. Zeng, KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 4961–4974. URL: https://aclanthology.org/2022.acl-long.340. doi:10.18653/v1/2022.acl-long.340.

[7] G. Izacard, P. S. H. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, E. Grave, Atlas: Few-shot learning with retrieval augmented language models, J. Mach. Learn. Res. 24 (2023) 251:1–251:43. URL: http://jmlr.org/papers/v24/23-0037.html.

[8] X. Zhang, N. Thakur, O. Ogundepo, E. Kamalloo, D. Alfonso-Hermelo, X. Li, Q. Liu, M. Rezagholizadeh, J. Lin, Making a miracl: Multilingual information retrieval across a continuum of languages, 2022. arXiv:2210.09984.

[9] E. Kamalloo, A. Jafari, X. Zhang, N. Thakur, J. Lin, HAGRID: A human-llm collaborative dataset for generative information-seeking with attribution, CoRR abs/2307.16883 (2023). URL: https://doi.org/10.48550/arXiv.2307.16883. doi:10.48550/ARXIV.2307.16883. arXiv:2307.16883.

[10] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[11] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, 2020. arXiv:1904.09675.

[12] L. Gao, Z. Dai, P. Pasupat, A. Chen, A. T. Chaganty, Y. Fan, V. Zhao, N. Lao, H. Lee, D.-C. Juan, K. Guu, RARR: Researching and revising what language models say, using language models, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),

Association for Computational Linguistics, Toronto, Canada, 2023, pp. 16477–16508. URL: https://aclanthology.org/2023.acl-long.910. doi:10.18653/v1/2023.acl-long.910.

[13] L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, N. Sarrazin, O. Sanseviero, A. M. Rush, T. Wolf, Zephyr: Direct distillation of lm alignment, 2023. arXiv:2310.16944.

[14] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389. doi:10.1561/1500000019.

[15] R. Nogueira, Z. Jiang, R. Pradeep, J. Lin, Document ranking with a pretrained sequence-to-sequence model, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 708–718. URL: https://aclanthology.org/2020.findings-emnlp.63. doi:10.18653/v1/2020.findings-emnlp.63.