Actes de CORIA-TALN-RJCRI-RECITAL 2025
Marseille, France, 30 juin au 4 juillet 2025

Évaluation automatique du retour à la source dans un contexte historique long et bruité. Application aux débats parlementaires de la Troisième République française

Aurélien Pellet[1, 2] Julien Perez[1] Marie Puren[1, 3]
(1) LRE, 13-14 rue Voltaire, 94270 Le Kremlin-Bicêtre, France
(2) EPITECH, 24 rue Pasteur, 94270 Le Kremlin-Bicêtre, France
(3) CJM, 65 rue de Richelieu, 75002 Paris, France
aurelien.pellet@epita.fr, julien.perez@epita.fr, marie.puren@epita.fr

RÉSUMÉ

Dans le contexte de l'utilisation croissante des LLM, le besoin d'un retour efficace et automatique aux sources devient essentiel, en particulier pour les documents historiques. La capacité des LLM à identifier les sources pertinentes ne constitue plus seulement un maillon dans une chaîne où l'objectif final est la génération de réponses ; elle représente un enjeu fondamental de l'analyse, justifiant une évaluation à part entière. Quelles stratégies, quels modèles et quels paramètres offrent aux historiens les meilleures capacités d'exploration d'un corpus vaste et bruité ? Cet article propose une première tentative d'évaluation du retriever dans un cadre de RAG appliqué aux débats parlementaires de la Troisième République.

ABSTRACT

Automatic Evaluation of Source Tracing in a Long and Noisy Historical Context : Application to Parliamentary Debates of the French Third Republic.

In the context of the growing use of LLMs, the need for effective source retrieval is becoming essential, particularly for historical documents. The ability of LLMs to identify relevant sources is no longer just a link in a larger chain where the ultimate goal is answer generation ; it is a fundamental aspect of analysis that warrants dedicated evaluation. Which strategies, models, and parameters provide historians with the best means to explore a large and noisy corpus ? This paper presents a first attempt at evaluating the retriever in a RAG framework applied to the parliamentary debates of the French Third Republic.

MOTS-CLÉS : Humanités numériques, LLM, RAG, segmentation, retour à la source, documents historiques.

KEYWORDS: Digital humanities, LLM, RAG, chunking, information retrieval, historical documentsh.

ARTICLE : **Accepté à** Nom de la Conférence / Revue.

Actes de l'atelier Évaluation des modèles génératifs (LLM) et challenge 2025 (EvalLLM), pages 138–150.

1 Introduction

1.1 Le corpus des débats parlementaires

La fin des années 2000 a vu le début de la numérisation des transcriptions des débats parlementaires de la Troisième République (Alix, 2008) [1]. Hormis quelques rares éclipses au dix-neuvième siècle, ces transcriptions ont continué à rendre publiques les discussions entre députés et, à partir des années 1840, se sont attachées à retranscrire les débats dans le détail, tout en essayant de rendre le naturel des discussions (Coniez, 2008; Gardey, 2010). Cette nouvelle abondance de données pour l'histoire parlementaire (Ihalainen, 2021; Blaxill, 2022) permet d'étudier les débats dans la longue durée, et ainsi d'observer l'évolution des questions politiques et sociales abordées par les assemblées (comme dans (Baker *et al.*, 2017)).

Bien que ces données soient aujourd'hui facilement accessibles, ce corpus reste encore sous-exploité. Cette nouvelle abondance de sources peut avoir un effet paralysant pour les historiens, vu l'énormité du corpus à analyser. Comme l'explique Hannu Salmi, "If the nature, quality, and extent of the source material available for research have changed, it is natural that the research toolbox must change in tandem" (Salmi, 2021). L'accès à de grandes quantités de sources oblige ainsi à adapter les méthodes de travail utilisées par les historiens à l'exploration de ces données, et à développer de nouvelles méthodes en informatique pour répondre aux demandes des utilisateurs finaux, et aux défis posés par ce type de documents.

1.2 Grands modèles de langue et histoire

Si l'analyse de données textuelles n'est pas nouvelle en histoire (Dumont *et al.*, 2023; Prost, 1988), les récents progrès en apprentissage automatique et en traitement du langage naturel (Clavert & Muller, 2024) permettent désormais d'envisager l'analyse des discours parlementaires sur la longue durée. L'avènement des grands modèles de langage (LLMs) permet aujourd'hui d'extraire des données de ces documents et de les analyser à grande échelle. Les LLMs ont démontré des performances impressionnantes sur une large gamme de tâches en langage naturel, ainsi qu'une grande capacité de compréhension linguistique généraliste. Toutefois, les LLMs présentent plusieurs limites bien identifiées (hallucinations, obsolescence des informations, ou encore raisonnements non transparents et non traçables), ce qui remet en question leur utilisation dans les travaux historiques, fondés sur la vérifiabilité des sources.

La Génération Augmentée par Récupération (Retrieval-Augmented Generation ou RAG) (Lewis *et al.*, 2020) propose une solution innovante à ces problèmes (Minaee *et al.*, 2024). En combinant les forces des modèles de génération traditionnels avec la capacité de retrouver des informations pertinentes dans de vastes corpus textuels, ces systèmes peuvent produire des réponses plus précises à une grande variété de requêtes en langage naturel (Yahya *et al.*, 2012; Sun *et al.*, 2018; Petroni *et al.*, 2019). Ces systèmes utilisent un module de recherche (ou *retrieval*) pour identifier des passages de texte pertinents, qui sont ensuite utilisés pour conditionner un modèle de génération. Cela permet au modèle de produire des réponses éclairées par un large éventail d'informations pertinentes, au lieu d'être limité aux connaissances explicitement encodées dans ses paramètres.

1. Les transcriptions sont publiées dans le *Journal Officiel de la République française*. Pour la période qui nous intéresse, on les trouvera sur Gallica, la bibliothèque numérique de la Bibliothèque nationale de France, pour la Chambre des députés et le Sénat.

1.3 Le RAG : quels défis avec les corpus historiques ?

Un RAG naïf (Gao *et al.*, 2023) (ou RAG basique) procède de la manière suivante : dans une première phase, on utilise un modèle d'*embedding* pour indexer des documents. Une requête est ensuite soumise par un utilisateur, à partir de laquelle et à l'aide du modèle d'*embedding* on va chercher les k-documents les plus similaires à la requête, où k est un paramètre à fixer. Il suffit ensuite de concaténer dans un prompt la requête de l'utilisateur et les documents récupérés, et d'envoyer le tout à un LLM pour lui demander de répondre à la question posée. On peut également affiner l'architecture de base d'un RAG en ajoutant plusieurs composantes, c'est le cas de selfRAG (Asai *et al.*, 2023) qui, en fonction de la requête, décide par lui-même de passer par une phase de *retrieving*.

Utiliser un RAG pour interroger un corpus de sources historiques implique d'interroger des documents qui peuvent être particulièrement longs. Typiquement, dans une seule séance parlementaire, plusieurs débats se succèdent et certains d'entre eux s'étirent dans le temps, voire sont à l'ordre du jour de plusieurs séances ; les orateurs interviennent à tour de rôle à la tribune, parfois plusieurs fois, au sein d'un même débat, et peuvent donc traiter à plusieurs reprises d'un même sujet et répondre à une ou des intervention(s) précédente(s). Le RAG doit donc être capable de récupérer le fragment de texte pertinent pour répondre à la question historique posée, tout en minimisant la fenêtre de contexte nécessaire. Le défi consiste donc à repérer les segments du corpus pertinents, tout en exigeant un haut degré de précision dans la réponse obtenue.

1.4 Rechercher les informations pertinentes : le *retriever*

La phase de *retrieval* constitue le premier bloc majeur d'un RAG. Les différents sous-documents de notre corpus sont d'abord indexés à l'aide d'*embeddings*. Les méthodes classiques de récupération de documents utilisaient des représentations *sparse* du texte, comme TF-IDF ou BM25 ; mais ces méthodes restent assez limitées, notamment pour capturer les notions de similarité sémantique lorsque les mots utilisés diffèrent. Les représentations avec des vecteurs denses, apprises sur des documents (Karpukhin *et al.*, 2020), se sont révélées très efficaces pour des tâches comme le *retrieval* ou le *clustering*. Elles reposent sur l'entraînement d'une représentation sémantique à partir de paires de questions/réponses. Les modèles les plus récents, en multilingue, permettent d'encoder des textes de plus en plus longs (Zhang *et al.*, 2024). D'autres méthodes, comme SPLADE (Formal *et al.*, 2021), proposent une approche basée sur des *embeddings* plus *sparse*, qui sont notamment plus efficaces pour la recherche par mots-clés, comme on le retrouvait dans les méthodes classiques évoquées. Des *benchmarks* classiques permettent de comparer et d'identifier les modèles à l'état de l'art (Muennighoff *et al.*, 2023).

La phase d'évaluation du *retriever* vise à mesurer la capacité d'un modèle d'*embedding* à récupérer tous les documents pertinents en fonction d'une requête. Dans ce contexte, plusieurs paramètres peuvent être évalués pour analyser leur influence sur le *retriever*. Des travaux ont également été développés pour étudier l'impact des stratégies de *chunking* (**?**), notamment en comparant différentes tailles, différents niveaux de découpage, mais aussi différentes approches, comme le découpage sémantique. Les corpus sur lesquels elles s'appliquent représentent rarement les caractéristiques des documents historiques longs et bruités. Enfin, des travaux cherchent à évaluer l'impact d'un *retriever* dans le contexte global d'un RAG (Alinejad *et al.*, 2024).

1.5 Objectifs

L'objectif premier du RAG consiste à maximiser la probabilité de prédire la bonne réponse tout en minimisant la taille de l'entrée. Dans ce contexte, le *retriever* apparaît comme une étape importante, mais moins cruciale à l'échelle de l'objectif final. Elle reste toutefois essentielle dans le contexte de la recherche historique : optimiser cette phase de *retrieval* consiste *in fine* à optimiser les capacités de retour à la source des modèles de langue. Nous proposons ici d'étudier nos premiers résultats qui concentrent essentiellement sur ces aspects du *retrieval*.

Comme objets de comparaison, nous cherchons à comprendre comment différentes stratégies de découpage d'un débat permettent d'optimiser le retour à la source. L'enjeu est de trouver un équilibre entre, d'une part, un découpage fin, à haute précision, qui réduit la taille des segments et facilite le retour précis à la source, et d'autre part, des méthodes de découpage plus grossières, qui offrent davantage de contexte, augmentent les chances de récupération des documents pertinents, mais au prix d'un traitement plus volumineux et d'une complexité accrue pour l'historien. C'est cet arbitrage entre performance et frugalité que nous explorons à travers ces premières évaluations.

2 Méthodologie

2.1 Génération automatique de questions

Afin de pouvoir évaluer le *retriever*, nous utilisons un modèle de langue pour générer automatiquement et rapidement un grand nombre de questions. La génération automatique de questions permet en effet de constituer rapidement un jeu de données sur lequel évaluer les différentes composantes de notre système RAG. Plusieurs méthodes existent, allant de la génération de questions simples à la génération *multi-hop*, accompagnée de la création d'un graphe de connaissances (**?**Krishna *et al.*, 2024). Dans notre étude, nous nous concentrons sur la génération de questions *single-hop*, c'est-à-dire des questions qui nécessitent uniquement un segment du texte d'origine pour pouvoir y répondre. Le modèle utilisé est Qwen2.5 dans sa version 32B (Qwen, 2024). Il est confronté à différents segments d'un débat et chargé de générer une ou plusieurs questions, tout en renvoyant le passage précis du texte utilisé pour produire la question. De manière similaire à d'autres travaux (**?**), nous effectuons une phase de pré-filtrage afin de nous assurer que la source renvoyée par le LLM corresponde exactement à un extrait du texte. Par défaut, les questions générées sont souvent peu pertinentes ou incohérentes. Un travail collaboratif avec un historien nous a permis d'affiner le *prompt* afin que les questions produites aient le plus de sens possible. Le prompt utilisé est présenté dans Table 3 (Annexe A). Un total de 1486 questions a été généré. Pour chacune des questions générées, nous nous sommes assurés que la source renvoyée par le LLM corresponde parfaitement au texte d'origine. Nous illustrons quelques exemples de questions générées dans la Table 4 (Annexe A).

Cette approche initiale révèle un certain nombre de limites :

1. Un certain nombre de questions posées ne sont pas pertinentes du point de vue de l'historien.
2. L'ensemble des questions formulées ainsi que les évaluations se feront intra-séance ; la génération de questions portant sur plusieurs séances demandera un travail plus conséquent qui sera fait en collaboration avec les historiens.
3. Les questions ne portent que sur un unique segment d'une séance. À plus long terme, nous

chercherons à produire des questions *multi-hop* pour tester plus profondément les capacités du *retriever*, comme cela a été montré dans (Krishna *et al.*, 2024).

Malgré ces limitations, cette première génération de questions permet de produire un dataset cohérent, qui sert de première base à l'évaluation du *retriever*, et nous servira par la suite à reproduire les évaluations sur un jeu de données plus complexe et adapté à la recherche en histoire.

2.2 Les stratégies de *chunking* (ou segmnentation)

Méthode de chunking	**Explication**	**Détail**
S	Séparations par section (S) d'un débat (à l'aide d'une regex)	La regex utilisée identifie les débuts de blocs par une suite de majuscules et de caractères de ponctuation
S&PP	Séparations par Section (S) d'un débat et par prise de parole (PP)	Les prises de paroles commencent toutes par un "M." ou une forme de ce genre
S&PP&MaxN	Séparation par section d'un débat et par prise de parole. On s'arrête de découper dès qu'il y a moins de N caractères dans un bloc	

TABLE 1 – Description des stratégies de *chunking* étudiées. Pour chaque méthode, un chunking récursif est appliqué, et les frontières entre les différentes sections sont identifiées à l'aide d'expressions régulières.

Le niveau de granularité choisi dans le découpage de nos documents a un impact direct sur l'interrogation du corpus (**?**). Un découpage en gros blocs permet potentiellement de donner plus de contexte au LLM pour répondre à la question. Cependant, la source renvoyée est d'autant plus grande et ne facilite pas l'exploration de l'historien. Enfin, des *chunks* (ou segments) trop grands risquent de diminuer l'efficacité des méthodes d'*embedding* qui vont devoir encoder une plus grande quantité d'informations, et devenir potentiellement moins précises. Sans oublier que plus les *chunks* sont grands, plus la phase de génération finale sera gourmande en énergie.

Dans cette optique, nous cherchons à réaliser une première comparaison de différentes stratégies de *chunking*. En plus de les comparer les unes aux autres, nous proposons de les comparer à « égalité de tokens produits ».

La Table 1 présente une évaluation comparée de différentes stratégies de *chunking*. Pour chaque stratégie, la segmentation s'effectue de manière récursive.

— **S** : On commence par découper le débat en sections. Les sections sont généralement séparées par des titres en majuscule. Du fait des erreurs d'OCR, nous avons mis en place une expression régulière détaillée pour capter au mieux les différentes sections. Cette séparation par section est plutôt efficace et produit en moyenne des *chunks* de taille conséquente.

— **S&PP** : En poursuivant la segmentation récursivement, on découpe jusqu'à la prise de parole. Les changements d'orateurs sont caractérisés par un saut de ligne, débutant par « M. » (abbréviation de « Monsieur »). Là encore, nous utilisons une expression régulière qui réussit à capter le mieux possible les segments. Les *chunks* produits ici sont de bien plus petite taille ; un *chunk* peut être aussi court qu'une phrase ou quelques mots tant que cela constitue une prise de parole.

— **S&PP&MaxN** : On reproduit le découpage récursif jusqu'à la prise de parole, mais on se fixe une borne supérieure. Le découpage s'arrête quand la taille du segment est en dessous de N caractères. Étant donné qu'une prise de parole peut être longue, on autorise le segmenteur à découper par saut de ligne, voire même par passage à la ligne. Cette méthode permet de créer des *chunks* dont la taille est plus contrôlable que dans les deux autres méthodes.

Toutes ces stratégies respectent la structure globale des documents. Par leur taille et le type d'information qu'elles regroupent, les *chunks* représentent différents rapports à la source d'origine et à son interprétation. Nous cherchons ensuite à évaluer comment ces stratégies de *chunking* peuvent impacter la capacité d'un modèle à récupérer l'information pertinente.

2.3 Méthodes d'évaluation

Il existe plusieurs méthodes d'évaluation du *retriever* (Alinejad *et al.*, 2024) (Alinejad *et al.*, 2024; **?**). Étant donné que, dans notre cas d'étude, chaque question ne contient qu'une seule bonne référence, nous nous proposons d'évaluer deux métriques qui sont basées sur de **l'exact matching** : le *Recall* et le nDCG.

Recall@K Le *Recall@K* mesure la capacité du *retriever* à récupérer le document contenant la réponse parmi les K documents les plus similaires à une question. Pour chaque question, il vaut 1 si la source correcte est dans les K premiers documents, 0 sinon. Le *Recall@K* global est la moyenne de ces valeurs sur toutes les questions, évaluée pour différentes valeurs de K afin d'analyser les performances du modèle.

nDCG@K Le *Normalized Discounted Cumulative Gain* (nDCG@K) évalue la pertinence et la position des documents récupérés. Contrairement au *Recall*, il pénalise les documents pertinents classés plus bas. Pour une question, le *DCG* somme les scores de pertinence (1 pour la source correcte, 0 sinon), dévalués par un facteur logarithmique selon leur position. Le *nDCG@K* est obtenu en divisant le *DCG* par le *DCG* idéal (source correcte en première position). Le *nDCG@K* global est la moyenne sur toutes les questions, reflétant la qualité du classement des résultats.

Nous pouvons ensuite appliquer ces mesures pour comparer et évaluer les différentes architectures de RAG et méthodes de *chunking*.

3 Résultats

3.1 Évaluation du retriever

La Table 2 illustre les premières évaluations obtenues sur quatre configurations de *chunking*. Le premier point à souligner est que, comme attendu, la stratégie **S** produit les meilleurs résultats. Cela s'explique par le fait qu'elle génère les *chunks* les plus grands. Cependant, cette approche se fait au détriment d'une taille de *chunks* très importante (11 000 en moyenne), ce qui ne permet pas un retour à la source optimal. La stratégie **S&PP&Max10000** affiche un bon compromis entre la taille des *chunks* et l'efficacité du *retrieval*. Elle obtient les deuxièmes meilleurs résultats pour presque toutes les métriques, avec une fraction du coût en *tokens* par rapport à **S**. À partir de trois documents récupérés, elle est comparable à **S** en termes de *recall*, mais reste moins performante selon le *nDCG*.

Stratégie de Chunking		Taille moyenne des chunks *(nombre de tokens)*	Recall@k					nDCG@k	
Granularité	**Taille limite** *(caractères)*		1	2	3	5	10	5	10
S&PP	1000	204 ± 66	0.54	0.66	0.73	0.79	0.87	0.68	0.70
	10000	1412 ± 940	0.58	0.72	0.78	0.86	0.93	0.73	0.74
S	None	10989 ± 14302	0.62	0.73	0.78	0.85	0.93	0.74	0.77
PP		808 ± 1267	0.50	0.65	0.71	0.79	0.87	0.66	0.68

TABLE 2 – Évaluation du *retriever* pour différentes stratégies de *chunking*. Pour chaque configuration, nous calculons le nombre moyen de tokens par chunk récupéré sur l'ensemble des questions. La stratégie de segmentation par section donne presque toujours les meilleurs résultats, mais elle est très coûteuse en nombre de tokens. La stratégie **S&PP&Max10000** offre un bon compromis entre coût à l'entrée et performance du retriever.

Un autre point intéressant concerne la comparaison des deux autres configurations. Bien qu'elle produise en moyenne des *chunks* quatre fois plus petits, **S&PP&Max1000** est plus performante que **PP**. Cela peut s'expliquer par le fait que **PP** génère des *chunks* avec une grande variabilité de taille. Certains segments de grande taille peuvent produire un *embedding* de moindre qualité, ce qui introduit de la confusion. En revanche, **S&PP&Max1000** produit des *chunks* de taille similaire et plus petits, facilitant la récupération par similarité sémantique. Pour des modèles de petite taille, la stratégie de découpage **S&PP&Max1000** semble donc être la meilleure solution.

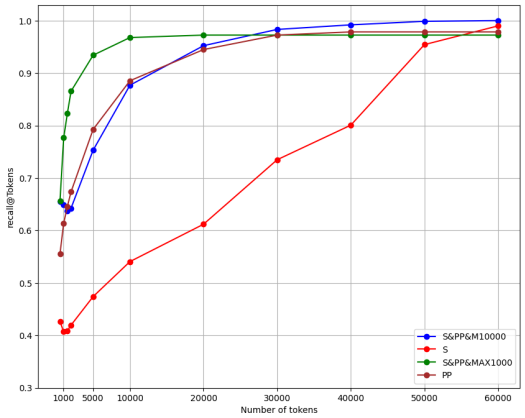


FIGURE 1 – Comparaison des stratégies de *chunking* en fonction du nombre maximal de tokens. Pour chaque stratégie, nous calculons le *recall* pour une limite donnée de tokens afin de les comparer plus efficacement. Seules les questions pour lesquelles au moins un document a pu être récupéré sont prises en compte pour chaque stratégie.

La Figure 1 présente l'évaluation du *recall@k* à nombre de *tokens* égal. Pour chaque stratégie de *chunking* et chaque seuil de nombre maximal de *tokens*, nous avons récupéré autant de documents que possible dans la limite de ce nombre de *tokens*. Le *recall* est ensuite évalué à ce seuil. Cette méthode permet une comparaison précise à un même nombre de *tokens*. Dans le graphique, l'hypothèse est que, si pour une question donnée aucun segment ne respecte la limite de contexte, alors cette question est ignorée pour la stratégie considérée. La performance plus faible de **S** s'explique par le fait que la source de la réponse à une question est souvent perdue dans des chunks de trop grande taille, ce qui dilue l'information sémantique dans l'*embedding*. Ainsi, seuls des chunks peu significatifs et de petite taille sont récupérés en priorité. Concernant les autres stratégies, à nombre égal de *tokens*, c'est la méthode **S&PP&Max1000** qui fournit le meilleur *recall*. Cela peut s'expliquer par le fait que, en

produisant des *chunks* de petite taille, l'*embedding* calculé est plus précis, permettant au *retriever* de récupérer plus facilement la source de la réponse à la question.

3.2 Comparaison de deux stratégies de RAG

Nous comparons à présent deux stratégies de RAG en fonction des différentes méthodes de *chunking*. Nous mettons en regard notre première architecture de RAG naïf avec une version augmentée par *reranking*. Le modèle utilisé pour le *reranking* est `bge-reranker-v2-m3` (Li *et al.*, 2023; Chen *et al.*, 2024). Pour chaque question, nous récupérons $k = 20$ documents et appliquons le *reranker* sur le couple (question, document). Pour chaque couple, nous obtenons un score de proximité qui nous permet ensuite de reclasser les documents. Nous pouvons alors recalculer le *recall*. Sur la Figure 2, les lignes pleines correspondent aux stratégies de *chunking* avec un RAG naïf, tandis que les lignes pointillées correspondent aux stratégies de RAG avec *reranking*. On constate clairement le gain apporté par le modèle de *reranking*.
Pour $k = 1$, on passe de 0.5 en *recall* pour **PP** à 0.79. Concernant les stratégies **PP** et **S&PP&Max1000**, nous avons vu qu'elles produisaient des résultats proches dans le cadre d'un RAG naïf ; l'approche par *reranking* permet d'observer des différences. **PP** obtient en moyenne deux points de performance supplémentaires à partir de $k = 3$ documents récupérés. La phase de *reranking* compense les imprécisions des méthodes d'*embedding*, rendant la stratégie de découpage par prise de parole plus intéressante malgré des *chunks* de plus grande taille. Concernant **S&PP&Max10000**, elle reste une stratégie intéressante, car elle offre un bon compromis entre taille et efficacité, avec un *recall* supérieur d'au moins cinq points sur les différentes évaluations.
La Table 5 (Annexe A) permet d'avoir une vision plus claire du coût en calcul des méthodes de *reranking*. Pour des stratégies comme **S&PP&Max10000** et **PP**, le temps d'inférence sur l'ensemble des questions est d'environ deux heures. Les performances du *retriever* sont effectivement bien meilleures, mais à un coût élevé. La méthode **S&PP&Max1000** semble offrir ici un compromis intéressant : elle produit des résultats presque toujours supérieurs à toutes les autres stratégies sans phase de *reranking*, tout en maintenant un temps d'inférence raisonnable, inférieur à 10 minutes.

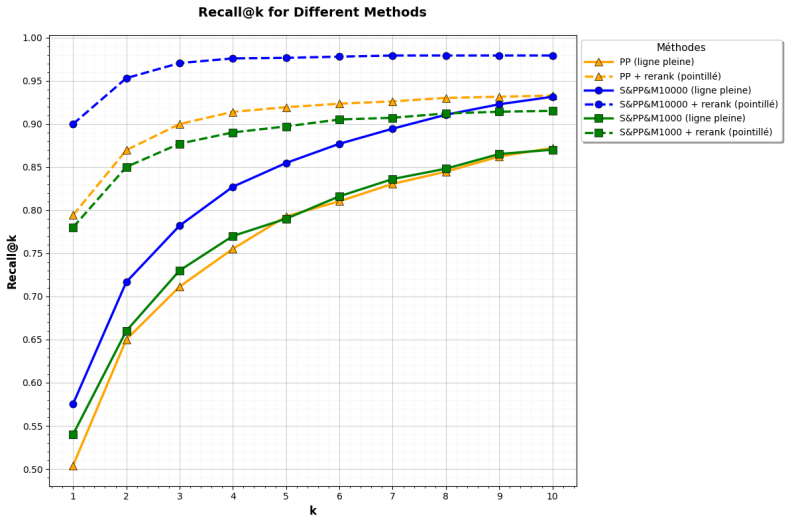


FIGURE 2 – Comparaison des résultats sur le retriever avec et sans reranking. En ligne pleine les résultats avec un RAG classique et en pointillé avec un RAG augmenté par reranking

4 Discussion et conclusion

Nous avons proposé une première approche pour une évaluation indépendante de la phase de *retrieval* d'un RAG, appliquée à l'analyse des débats parlementaires de la Troisième République. Malgré des limites liées à la génération des données d'évaluation, nous avons observé que le découpage et la segmentation des documents influencent fortement la phase de *retrieval*. Dans le cadre d'un RAG, trois aspects peuvent être évalués : la réponse finale du modèle, la récupération des documents et le coût en nombre de *tokens* d'entrée et de sortie. Nous avons montré comment différentes stratégies offrent des compromis entre le coût en *tokens* d'entrée et l'efficacité du *retrieval*. Enfin, nous avons constaté qu'une amélioration de l'architecture du RAG augmente significativement les performances, mais au prix d'une inférence plus longue. À l'avenir, nous chercherons à affiner et produire un jeu de données plus représentatif des besoins des historiens, intégrant des questions nécessitant un raisonnement avancé. Nous souhaitons également approfondir nos analyses des stratégies de découpage (en intégrant, par exemple, de la segmentation sémantique) et explorer d'autres architectures de RAG intégrant du raisonnement. Enfin, nous visons une évaluation *end-to-end* du RAG, incluant le coût en *tokens* d'entrée et de sortie, l'évaluation du *retriever*, ainsi que la validité de la réponse à une question. Ces premiers résultats serviront de base à ces futures explorations.

Références

ALINEJAD A., KUMAR K. & VAHDAT A. (2024). Evaluating the retrieval component in llm-based question answering systems. DOI : 10.48550/ARXIV.2406.06458.

ALIX Y. (2008). La numérisation concertée en sciences juridiques. *Bulletin des bibliothèques de France (BBF)*, **5**, 93–94.

ASAI A., WU Z., WANG Y., SIL A. & HAJISHIRZI H. (2023). Self-rag : Learning to retrieve, generate, and critique through self-reflection. DOI : 10.48550/ARXIV.2310.11511.

BAKER H., BREZINA V. & MCENERY T. (2017). Ireland in british parliamentary debates 1803–2005. plotting changes in discourse in a large volume of time-series corpus data. In *Exploring Future Paths for Historical Sociolinguistics*, p. 83–107. Amsterdam : John Benjamins Publishing Company.

BLAXILL L. (2022). Parliamentary corpora and research in political science and political history. In *Proceedings of The Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference*, Marseille.

CHEN J., XIAO S., ZHANG P., LUO K., LIAN D. & LIU Z. (2024). M3-embedding : Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In L.-W. KU, A. MARTINS & V. SRIKUMAR, Éds., *Findings of the Association for Computational Linguistics : ACL 2024*, p. 2318–2335, Bangkok, Thailand : Association for Computational Linguistics. DOI : 10.18653/v1/2024.findings-acl.137.

CLAVERT F. & MULLER C. (2024). L'histoire au temps des algorithmes. une réflexion prospective sur l'introduction de l'intelligence artificielle en histoire au 21e siècle. *20 & 21. Revue d'histoire*, **2**(1162), 13–26.

CONIEZ H. (2008). *Écrire la démocratie. De la publicité des débats parlementaires*. Paris : L'Harmattan.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

DUMONT L., JULIEN O. & LAMASSÉ S. (2023). *Histoires de mots. Saisir le passé grâce aux données textuelles*. Paris : Éditions de la Sorbonne.

FORMAL T., PIWOWARSKI B. & CLINCHANT S. (2021). Splade : Sparse lexical and expansion model for first stage ranking. SIGIR '21, p. 2288–2292, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3404835.3463098.

GAO Y., XIONG Y., GAO X., JIA K., PAN J., BI Y., DAI Y., SUN J., WANG M. & WANG H. (2023). Retrieval-augmented generation for large language models : A survey. DOI : 10.48550/ARXIV.2312.10997.

GARDEY D. (2010). Scriptes de la démocratie : les sténographes et rédacteurs des débats (1848–2005). *Sociologie du travail*, **52**(12), 195–211.

IHALAINEN P. (2021). Parliaments as meeting places for political concepts. https://intellectualhistory.net/blog/parliaments-as-meeting-places-for-political-concepts. Center for Intellectual History Blog.

KARPUKHIN V., OGUZ B., MIN S., LEWIS P., 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, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.550.

KRISHNA S., KRISHNA K., MOHANANEY A., SCHWARCZ S., STAMBLER A., UPADHYAY S. & FARUQUI M. (2024). Fact, fetch, and reason : A unified evaluation of retrieval-augmented generation. DOI : 10.48550/ARXIV.2409.12941.

LEWIS P., PEREZ E., PIKTUS A., PETRONI F., KARPUKHIN V., GOYAL N., KÜTTLER H., LEWIS M., TAU YIH W., ROCKTÄSCHEL T., RIEDEL S. & KIELA D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, p. 9459–9474.

LI C., LIU Z., XIAO S. & SHAO Y. (2023). Making large language models a better foundation for dense retrieval. DOI : 10.48550/ARXIV.2312.15503.

MINAEE S., MIKOLOV T., NIKZAD N., CHENAGHLU M., SOCHER R., AMATRIAIN X. & GAO J. (2024). Large language models : A survey. DOI : 10.48550/ARXIV.2402.06196.

MUENNIGHOFF N., TAZI N., MAGNE L. & REIMERS N. (2023). MTEB : Massive text embedding benchmark. In A. VLACHOS & I. AUGENSTEIN, Éds., *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, p. 2014–2037, Dubrovnik, Croatia : Association for Computational Linguistics. DOI : 10.18653/v1/2023.eacl-main.148.

PETRONI F., ROCKTÄSCHEL T., RIEDEL S., LEWIS P., BAKHTIN A., WU Y. & MILLER A. (2019). Language models as knowledge bases ? In K. INUI, J. JIANG, V. NG & X. WAN, Éds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 2463–2473, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1250.

PROST A. (1988). Les mots. In *Pour une histoire politique*, p. 255–285. Paris : Éditions du Seuil.

QWEN (2024). Qwen2.5 technical report. DOI : 10.48550/ARXIV.2412.15115.

SALMI H. (2021). *What is Digital History ?* Polity Press.

SUN H., DHINGRA B., ZAHEER M., MAZAITIS K., SALAKHUTDINOV R. & COHEN W. W. (2018). Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 4231–4242, Brussels : Association for Computational Linguistics.

YAHYA M., BERBERICH K., ELBASSUONI S., RAMANATH M., TRESP V. & WEIKUM G. (2012). Natural language questions for the web of data. In *Proceedings of the 2012 Joint Conference on*

Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), p. 379–390, Jeju Island, Korea : Association for Computational Linguistics.

ZHANG X., ZHANG Y., LONG D., XIE W., DAI Z., TANG J., LIN H., YANG B., XIE P., HUANG F., ZHANG M., LI W. & ZHANG M. (2024). mGTE : Generalized long-context text representation and reranking models for multilingual text retrieval. In F. DERNONCOURT, D. PREOŢIUC-PIETRO & A. SHIMORINA, Éds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing : Industry Track*, p. 1393–1412, Miami, Florida, US : Association for Computational Linguistics. DOI : 10.18653/v1/2024.emnlp-industry.103.

A Annexe

Génération de questions Le prompt complet utilisé pour générer des questions est présenté dans la Table 3. Dans la Table 4, nous illustrons quelques questions générées ainsi que la source identifiée par le LLM pour produire la réponse.

Évaluation du retriever Dans la Figure 3 (Annexe A), nous affinons l'évaluation des stratégies en prenant en compte d'autres configurations pour la mesure du *recall*. La Figure 3b présente les résultats dans la configuration suivante : si, pour une stratégie de chunking donnée et une limite donnée du nombre de tokens, aucun document ne peut être récupéré, alors nous considérons que le retriever échoue à récupérer le bon document pour cette question. Cela explique la diminution du *recall* pour des tailles de contexte réduites. Les résultats restent inchangés pour **S&PP&Max1000**, car cette stratégie produit des chunks dont la taille est bien inférieure à la plus petite limite évaluée dans le graphique (500 tokens). Dans la Table 3a, les résultats sont présentés en évaluant les différentes stratégies sur le même sous-ensemble de questions pour un seuil donné. Par exemple, au seuil de 1000 tokens, seules les questions pour lesquelles au moins un document a pu être récupéré (tout en respectant le seuil) pour chacune des stratégies sont prises en compte.

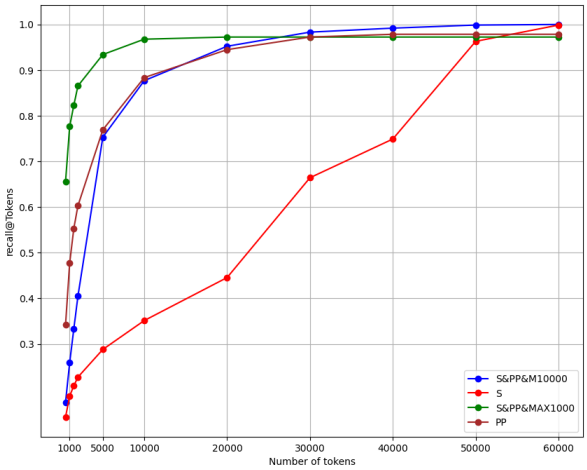
Évaluation des architectures de RAG Dans la Table 5, nous présentons les résultats du retriever en fonction de la stratégie de chunking et les comparons avec une architecture de RAG augmentée par reranking. Nous indiquons également le temps de calcul du retriever pour l'ensemble des 1487 questions.

Template du prompt

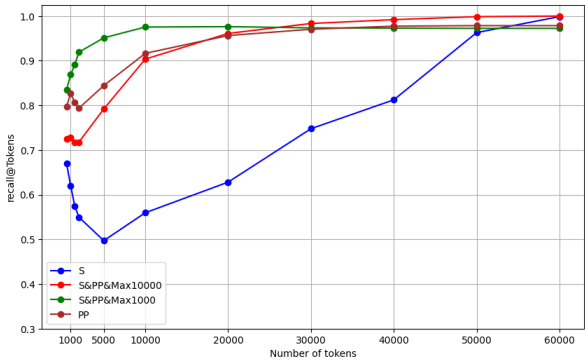
SYSTEM Tu es un assistant spécialisé dans la génération de questions basées sur des débats parlementaires de la Troisième République. Ta mission est de générer des questions factuelles précises et d'identifier les passages exacts qui permettent d'y répondre.
Instructions de génération :
1. Pour chaque texte fourni, génère des questions qui : - Ne peuvent être répondues que par des faits explicitement présents dans le texte - Incluent systématiquement la source de l'information (ex : "Selon le député X...") - Se concentrent sur des éléments factuels (qui, quand, où, combien) - Mentionnent tout le contexte nécessaire dans la question elle-même
2. Pour chaque question, tu dois : - Extraire le passage exact du texte qui contient la réponse - Conserver toutes les caractéristiques du texte original (ponctuation, orthographe, caractères spéciaux) - Ne faire aucune modification ou correction au passage extrait
3. Ne génère AUCUNE question si le texte est : - Une liste de députés - Un sommaire de séance - Une mention d'absences ou de congés - Une question écrite sans réponse (ex : "5383. - Question écrite, remise à la présidence de la Chambre, le 28 octobre 1915, par M. Raoul Méquillet")
4. Format de réponse obligatoire : - Si des questions sont possibles : ["question" : "Question complète avec contexte et source", "source" : "Passage exact du texte permettant de répondre" , "question" : "Autre question si pertinent", "source" : "Passage exact correspondant"] - Si aucune question n'est possible : "AUCUNE QUESTION"
Ne fais aucun commentaire supplémentaire et respecte strictement ce format de réponse.

USER Voici le texte : [TEXT]

TABLE 3 – Template de prompt pour la génération de questions sur les débats parlementaires



(a) Évaluation des stratégies de *chunking* en prenant en compte les questions pour lesquelles aucun document n'a pu être récupéré



(b) Évaluation des stratégies de *chunking* en évaluant à chaque seuil de *tokens* sur les mêmes questions. La forte inflexion pour la stratégie **S** s'explique

FIGURE 3 – Détail des évaluations des stratégies de *chunking* en prenant en compte différentes configurations pour l'évaluation du *recall@k*

Question	Source renvoyée
"Quel est le montant de la subvention offerte par le conseil général de la Haute-Garonne pour l'établissement de la ligne de Chaum au Pont-du-Roi ?"	Il est pris acte de l'offre faite par le conseil général de la Haute-Garonne, dans sa délibération du 7 avril 1880, de payer à l'État une subvention de 20,000 fr. par kilomètre pour l'établissement de la ligne de Chaum au Pont-du-Roi.
"Quel est le montant du droit perçu pour les envois de fonds au moyen de mandats de poste à destination du Japon ?"	« Art. 2. - Le droit à percevoir dans les bureaux de poste français pour les envois de fonds au moyen de mandats de poste à destination du Japon, est fixé à 0 fr. 10 par dix francs ; toute fraction de dix francs sera également passible d'un droit de 0 fr. 10. »
"Selon M. Cornil, combien de chirurgiens généraux existent dans l'armée anglaise sur 600 chirurgiens ?"	"Ainsi, dans l'armée anglaise, sur 600 chirurgiens, il y a 6 chirurgiens généraux dont les plus anciens ont le grade de major général,"
"Quelle est la position du comité consultatif d'hygiène publique sur la revaccination ?"	La revaccination, qui est le complément nécessaire de la vaccination pour assurer l'immunité contre la variole, doit être pratiquée dix ans au plus tard après une vaccination réussie, et répétée aussi souvent que possible, quand elle n'a pas été suivie de cicatrices caractéristiques.
"Selon M. Liouville, combien de personnes la variole a-t-elle fait mourir à Paris en 1880 ?"	Dansla dernière année, en 1880,lava*lf0l®afait,à Paris seulement, 2,258 vie.
Selon M. Clémenceau, quelle est la différence de prix entre la poudre de guerre et la poudre de commerce extérieur ?	et la preuve que ces deux sortes de poudres n'ont aucun rapport entre elles, c'est que la poudre de guerre se vend 2 fr., et la poudre de commerce extérieur 1 fr. 10.

TABLE 4 – Exemple de questions générées par un LLM avec la source renvoyée par le modèle

Stratégie de Chunking		**Taille moyenne des chunks** *(nombre de tokens)*	**Coût de calcul** *(temps)*	**Recall@k**			
Granularité	**Taille limite** *(caractères)*			1	2	5	10
S&PP	1000	204 ± 66	<1min	0.54	0.66	0.79	0.87
	10000	1412 ± 940	<1min	0.58	0.72	0.86	0.93
PP	None	808 ± 1267	<1min	0.50	0.65	0.79	0.87
S&PP+reranking	1000	204 ± 66	9min	0.78	0.85	0.89	0.92
	10000	1412 ± 940	1h47min	0.90	0.95	0.98	0.98
PP+reranking	None	808 ± 1267	2h25min	0.79	0.87	0.92	0.93

TABLE 5 – Évaluation du *retriever* pour différentes stratégies de *chunking* et pour deux architectures de RAG : sans et avec reranking. On affiche le temps de calcul du retriever pour chacune des configurations sur la totalité des questions générées. La stratégie **S&PP&Max1000** avec reranking, tout en restant dans un temps raisonnable apporte un gros gain en recall comparée à toute les autres stratégies de chunking sans reranking.