



Rapport de Stage recherche Semestre 10

M2 GPEx Minerve

Interprétabilité Mécaniste pour l'Extraction d'Information dans les systèmes de RAG

par Marine DELVALLEZ

Jury : Thi-Bich-Hanh DAO
Anaïs HALFTERMEYER

Tutrice de stage : Anaïs HALFTERMEYER

2 Février - 28 Août
Année universitaire 2025-2026

Table des matières

1	Introduction	3
1.1	Événements ayant jalonné ce stage	3
1.2	Contexte et problématiques traités dans ce mémoire	4
1.2.1	Contexte	4
1.2.2	Problématiques qui ont guidé ce travail	4
2	État de l’art	6
2.1	Systèmes de Génération et Grands Modèles de Langue Augmentés par Extraction (RAG et RALLM)	6
2.1.1	Besoin et Définition	6
2.1.2	L’architecture des modèles RA-LLM	7
2.1.3	Architectures aperçues au cours du stage	8
2.1.4	Architecture et Modèles utilisées au cours du stage	9
2.2	Interprétabilité pour les systèmes de RAG	9
2.2.1	Trustworthy AI	10
2.2.2	Explicabilité pour le RAG	10
2.2.3	Interprétabilité Mécaniste	10
2.2.4	Activation Patching	11
3	Objectifs du stage et préliminaires	14
3.1	Objectifs du stage	14
3.2	Challenge RAG@EvalLLM	14
3.3	Construction d’un dataset adapté	15
3.4	Adaptation du modèle RAG4HistoricalNewspapers	15
4	Contribution méthodologique	17
4.1	Construction des perturbations	17
4.1.1	Identification des mots cibles	17
4.1.2	Perturbations de type replace basées sur les synonymes	18
4.1.3	Perturbations de type replace plus agressives	19
4.1.4	Perturbations de type full_replace	21
4.2	Compléments pour l’estimation de la pertinence d’une perturbation	21
4.3	Interprétation des résultats de l’Activation Patching	23
4.3.1	Complément à la visualisation de la sensibilité du modèle à la perturbation	23
4.3.2	Critère d’identification des nœuds sensibles	24
4.3.3	Matrice des nœuds sensibles	25
5	Résultats expérimentaux	26
5.1	Résultats et Analyse	26
5.1.1	Application à la cartographie du vocabulaire de la défense français dans mE5	26
5.1.2	Préconisations d’utilisation	26
5.2	Perspectives d’évolution de la méthode proposée	27
6	Conclusions et Perspectives	28
6.1	Conclusion	28
6.2	Perspectives générales	28

A	Lieu du stage, outils et ressources utilisés	31
A.1	Laboratoire d'Informatique Fondamentale d'Orléans	31
A.2	Outils utilisés durant le stage	31
A.3	Financement	32
B	Dépôt Git des outils développés durant le stage	33

Chapitre 1

Introduction

1.1 Événements ayant jalonné ce stage

Ce mémoire restitue les travaux et réflexions que j’ai portés à l’occasion de mon stage de fin de master ARIAS, parcours Minerve GPEx au sein de l’équipe Contraintes et Apprentissage du Laboratoire d’Informatique Fondamentale d’Orléans (LIFO) au cours duquel j’ai été encadré par Mme Halftermeyer. Au cours de ces six mois de stage et en parallèle de mon travail au sein du LIFO, j’ai pu suivre et participer aux événements scientifiques et de dissémination suivants :

Le colloque pluridisciplinaire **Sociétés en transition à l’ère de l’intelligence artificielle** (TransIA) portait sur les impacts de l’Intelligence Artificielle sur la société et les différents pans de la recherche. Il regroupait des présentations issues des différentes disciplines allant des Sciences de l’éducation au droit privé et droit public en passant par l’étude de l’impact social et environnemental de l’IA. Il m’a permis d’aborder la recherche au sujet de l’intelligence artificielle sous l’angle de son impact et d’avoir un regard plus ouvert sur les autres disciplines de recherche et leur application à l’IA.

La journée du Groupe de Recherche **TAL-I** regroupait des présentations et une session poster portant sur l’interprétabilité dans le contexte du Traitement Automatique des Langues Naturelles. Cette journée organisée à Paris m’a permis de compléter ma découverte du paysage de l’explicabilité pour le TALN effectuée durant mon immersion au semestre précédent et d’échanger avec d’autres chercheurs sur mon projet individuel ainsi qu’à propos de leur travaux. C’est à l’occasion d’un de ces échanges que j’ai choisi d’orienter mon sujet de stage sur l’utilisation d’outils d’explicabilité post-hoc.

J’ai aussi eu l’occasion de suivre en distanciel les présentations de la **Journée Frugalité en TAL et ML** qui portaient sur les enjeux et les impacts environnementaux et sociaux du développement du TAL et du ML et les solutions actuellement étudiées pour les prendre en compte et les réduire. Ces présentations m’ont permise de prendre conscience de l’importance de ses enjeux et mieux comprendre quels sont les leviers pour les prendre en compte dans des travaux de recherche en IA.

Le **Second Symposium on Mental Health and AI** regroupait des présentations de travaux à la croisée entre l’intelligence artificielle, les sciences cognitives et la psychologie. J’ai pu suivre certaines présentations en lien avec le projet collaboratif que nous avons présenté avec Eva Troupillon au semestre précédent dans le cadre du module Défi Appel à Proposition de Projet.

À l’occasion des **journées de l’équipe CA**, j’ai pu découvrir les travaux en cours au sein de l’équipe. J’y ai moi-même présenté les premières contributions méthodologiques et les difficultés auxquelles je faisais face dans le cadre de mon stage. J’ai pu prendre conscience de l’importance des échanges et de la collaboration au sein de la recherche et de la capacité de ces échanges à faire prendre du recul sur ses travaux et permettre d’avancer. C’est à partir d’un échange avec Mme Dao à la suite de ma présentation que certaines propositions de conception de perturbation sont nées.

J’ai aussi pu participer à un **challenge RAG** organisé par l’atelier EvalLLM. La tâche à laquelle je me suis inscrite permettait de compléter le paysage de mon sujet, notamment par un dataset francophone spécialisé dans la défense. Malheureusement, le calendrier ne m’a pas permis de soumettre de résultat valable et je n’ai donc pas pu concourir jusqu’au bout.

Mon intégration au LIFO m’a aussi facilité l’accès aux **séminaires organisés régulièrement** le lundi au sein du laboratoire. J’ai pu suivre plusieurs présentations de sujet d’étude de collègues de chercheurs du laboratoire. Ces présentations sur divers sujets (Traitement de la Langue, Automatisation des outils de développement, Graph

Neural Networks appliqués,...) me permettent de compléter ma culture des sujets de recherche d'actualités au sein du laboratoire et de ses partenaires qu'ils ce soient connexes au domaine de mon stage ou non.

En dehors des événements internes au monde de la recherche et de l'informatique, j'ai aussi pu marrainer la **Place du Numérique** organisée à Orléans le 9 juin dernier. Cet événement vise à faire connaître et rendre accessible la diversité du monde du numérique dont la recherche en informatique fait partie. J'ai notamment joué le rôle de témoin dans le *tribunal des générations futures* questionnant l'influence de l'intelligence artificielle sur notre comportement.

1.2 Contexte et problématiques traités dans ce mémoire

1.2.1 Contexte

Au sein de la recherche en deep learning, le développement de l'intelligence artificielle connaît une explosion depuis plusieurs années. Des modèles de plus en plus gros et de plus en plus complexes voient le jour et repoussent leurs limites très régulièrement. Une des grande difficultés de ces modèles et de contenir une très grande quantité et diversité d'informations leur permettant non seulement d'aborder une grande diversité de sujet de façon assez poussée mais le tout en générant des contenu de très grande qualité que ce soit sous forme d'images, de bande sonore ou de texte.

Pour répondre à ce problème, [Lewis et al., 2020] propose de coupler des modèles capables de générer des contenus de grande qualité à des bases de données capable de fournir les connaissances nécessaires à la génération de contenu pertinent à l'aide d'un modèle d'extraction d'information. L'objectif d'une telle architecture, appelée **architecture RAG** (*Retrieval Augmented Generation*), est de libérer les modèles générateurs de la charge de la connaissance sans sacrifier le format de sortie pour un simple moteur de recherche. De plus, entretenir une base de connaissance sous la forme de texte ou d'embedding est plus aisé et et moins coûteux en calcul que de ré-entraîner régulièrement un modèle à plusieurs milliards de paramètres.

La croissante complexité des modèles profond engendre aussi un problème de confiance. En effet, les modèles profond suivent des structures élaborées rendant le raisonnement sous-jacent totalement opaque à l'utilisateur, qu'il soit spécialiste de ces machines ou non. Cependant, cette opacité est problématique dans certains cas d'usage. Lorsque l'impact d'une mesure est grande, il est central de pouvoir justifier son intérêt avant de l'appliquer. De façon plus générale, il est inenvisageable de confier à un outil dont on n'est pas capable de justifier la fiabilité des tâches sensibles. C'est pour permettre l'intégration de l'intelligence artificielle au sein de la médecine, de la défense, de la justice, de l'économie ou encore de la géo-politique qu'il est nécessaire de mettre en place des outils capable de rendre de la transparence aux modèles du deep learning et que le domaine de l'**explicabilité de l'intelligence artificielle** s'est développé.

Au cours de ce stage, nous nous sommes intéressés à l'application d'une méthode d'explicabilité aux modèles extracteurs pour les architectures RAG.

1.2.2 Problématiques qui ont guidé ce travail

Un objectif initial de ce stage était de défricher et construire un sujet de thèse s'intégrant à un projet de l'association *Info Jeunes Centre Val de Loire*¹. Cette association sensibilise et porte assistance aux jeunes de 15 à 30 ans pour notamment leur faciliter la connaissance et l'accès aux multiples dispositifs d'aide s'adressant à eux. La diversité, le nombre et les multiples critères d'accès à ces dispositifs représente une grosse charge de travail pour maintenir leur maîtrise des dispositifs qui changent souvent. Leur projet vise à concevoir un moteur de recherche de dispositifs permettant, étant donné un profil de jeune et sa ou ses problématiques, de trouver l'ensemble des aides qui lui sont accessible en justifiant la pertinence et donnant les éventuelles modifications administratives nécessaires pour y être éligible. Cet outil s'adresserait aux assistants employés par l'association dans un premier temps et pourrait être étendu sous la forme d'un chatbot afin de la rendre accessible aux jeunes directement.

Le stage visait donc à défricher la bibliographie et proposer un premier prototype de méthode de cartographie des modèles d'extraction d'information à l'aide des outils trouvés. Cette cartographie sera utilisée par la suite comme support pour expliquer le comportement de l'extracteur ainsi que pour améliorer ce dernier.

Au cours de l'exploration bibliographique, les méthodes d'interprétation mécaniste [Somvanshi et al., 2026], l'activation patching [Chen et al., 2024] et l'outil MechIR [Parry et al., 2025] ont été identifiés. L'interprétabilité mécaniste vise à "ouvrir" les modèles et observer le flux des données lors de leur utilisation afin de fournir des

1. <https://infojeunescvdl.fr/>

explications de leur fonctionnement fondées sur la "mécanique interne". L'activation patching est une méthode d'interprétation mécaniste et MechIR propose une implémentation de cette dernière.

Cependant, la prise en main et l'exploitation de MechIR donnant du fil à retordre, l'objectif du stage à du être revu. Il fallait d'abord concevoir une méthodologie d'usage de l'outil afin de, dans un premier temps, localiser le vocabulaire de spécialité dans des modèles d'extraction. Ces premiers résultats serviront de support pour la construction du sujet de thèse.

Chapitre 2

État de l'art

2.1 Systèmes de Génération et Grands Modèles de Langue Augmentés par Extraction (RAG et RALLM)

2.1.1 Besoin et Définition

Les modèles d'apprentissage profond tels que les réseaux de neurones profonds, réseaux convolutionnels ou les réseaux basés sur le mécanisme d'attention constituent leur connaissances à partir des données utilisées pour les entraîner. Cependant, leurs connaissances et leur capacité d'extrapolation sont limitées. Cela rend ces modèles vulnérables à l'obsolescence des résultats et aux hallucinations [Huang et al., 2025]. Pour traiter ces problématiques, [Lewis et al., 2020] propose d'associer une base de donnée et un modèle d'extraction des données de cette base à un modèle génératif. La base de données permet d'entretenir plus facilement les connaissances du modèle obtenu afin de limiter l'obsolescence des résultats et/ou d'agrandir la base de connaissances selon le besoin ou la tâche visée. Les systèmes combinant un modèle d'extraction (*retrieval* en anglais) et un modèle génératif sont appelées systèmes RAG (*Retrieval Augmented Generation*). La question ou requête de l'utilisateur est fournie au modèle d'extraction qui récupère les documents ou extraits de documents pertinent au regard de la question. Ces documents ainsi que la question sont alors donnés au générateur qui produit le résultat final. Ce processus est schématisé dans le Figure 2.1

Les systèmes RAG peuvent être appliqués à de nombreuses tâches générant différents types de données (génération d'image, génération et/ou complétion de signaux, génération de texte, traduction,...). Lorsque le modèle génératif est un Grand Modèle de Langue (LLM), on parle de systèmes RA-LLM [Fan et al., 2024]. Ces modèles sont utilisées pour grande diversité de tâches du traitement automatique des langues. On retrouve notamment les différentes variantes de réponse aux questions (questions à forte intensité de raisonnement, question sur domaine ouvert¹,...), les tâches d'analyse de texte (vérification de fait, liage d'entité²), des tâches de génération ou de complétion (traduction, prédiction de citation, dialogue, résumé de texte, complétion de texte à trous³,...) ou encore les tâches d'extraction (extraction d'information, extraction d'argument, ...)

1. Plus connus sous le nom de *Reasoning-Intensive Question Answering*, *Open-Domain QA*, ...

2. Fact Checking, Entity Linking, ...

3. Slot Filling

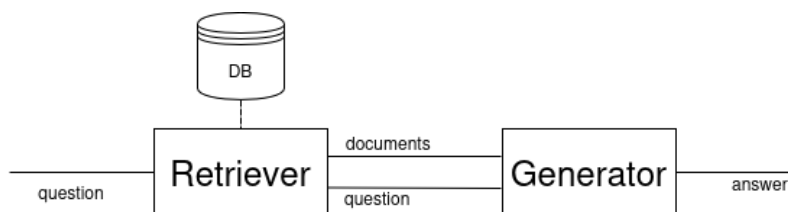


FIGURE 2.1 – Schema de l'architecture RAG proposée par [Lewis et al., 2020]

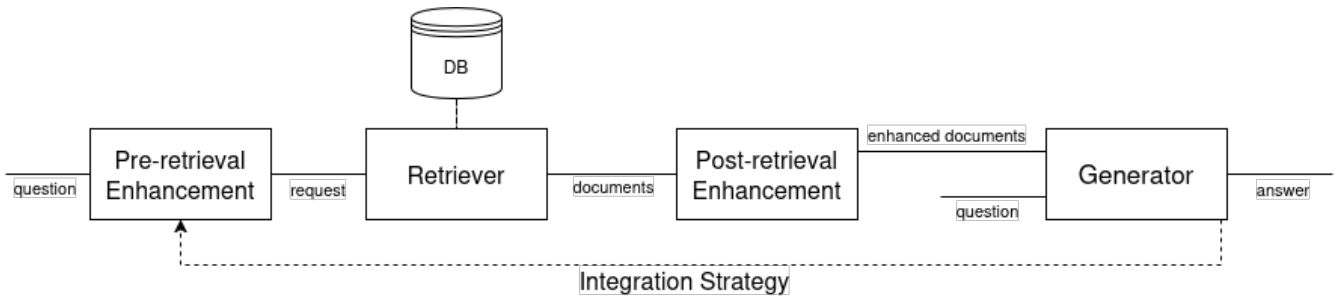


FIGURE 2.2 – Architecture généralisée des systèmes RA-LLM proposée par [Fan et al., 2024]

2.1.2 L’architecture des modèles RA-LLM

L’intuition de compléter un modèle génératif avec une base de donnée comme proposé par [Lewis et al., 2020] a été reprise à plusieurs reprises. Des stratégies et éléments complémentaires ont été ajoutés à ces systèmes pour faciliter la symbiose entre les deux composants principaux. [Fan et al., 2024] propose une nouvelle généralisation de cette famille de modèle illustré par la Figure 2.2

Extracteur On peut distinguer les extracteurs (aussi appelés retriever) en deux groupes. Les *Sparse retrievers* tels que TF-IDF ou BM25 [Sparck Jones, 1972, Lù, 2024] exploitent des statistiques sur les documents pour identifier les plus pertinents au regard de la question. Les *Dense retrievers* projettent les documents et les questions dans un espace latent au sein duquel les documents pertinents sont identifiés. On retrouve deux principaux fonctionnements pour ces extracteurs. Les *bi-encodeurs* projettent les documents et les questions séparément alors que les *cross-encodeurs* les projettent sous forme de paires (question, document) et fournissent directement le score de pertinence. Un hyper-paramètre important pour les extracteurs est la granularité d’extraction. Il correspond à la taille des extraits répertoriés dans la base de donnée. Les échelles les plus courantes sont celles du document, du passage et du token. On parle de *chunk retrieval* pour les deux premières et de *token retrieval* pour la troisième. On parle d’*entity retrieval* lorsque la base est construite sous l’angle de la connaissance et non celui du langage.

Augmentation pré-extraction Pour permettre une meilleure exploitation des informations disponibles dans la question, certains travaux proposent différentes stratégies de complétion et/ou reformulation de la question sous forme d’une requête. Le framework Rewrite-Retrieve-Read [Ma et al., 2023] propose de reformuler la question à l’aide d’un autre LLM pour la désambigüiser. L’approche de Query2Doc [Wang et al., 2023] consiste en générer des pseudo-documents qui pourraient correspondre aux documents pertinents à extraire et de les ajouter à la question avant de donner le tout à l’extracteur. L’objectif est de fournir des indices supplémentaires pour assurer une meilleure pertinence des documents extraits.

Augmentation post-extraction Pour que les documents extraits soient exploités le mieux possible, plusieurs travaux s’intéressent à des techniques d’augmentation de cet ensemble d’extraits. Cela peut passer par le classement des documents du plus pertinent au moins pertinent, le ré-ajustement du classement lorsqu’il existe, la combinaison des résultats de plusieurs extracteurs, la reformulation et/ou la synthèse des documents extraits.

Générateur [Fan et al., 2024] distingue les générateurs en deux groupes en fonction de l’accessibilité des paramètres : *parameter accessible* pour les modèles locaux modifiables pour lesquels on dispose des paramètres et *parameter inaccessible* pour les modèles dont les paramètres ne sont pas publics, qui sont généralement accessibles par API.

Intégration Une fois les documents récupérés et augmentés, il faut les fournir au générateur. [Fan et al., 2024] distingue trois stratégies d’intégration. L’*intégration en entrée* consiste à combiner les documents et la question (généralement à l’aide d’une template de prompt) avant de la fournir au générateur. L’*intégration en sortie* consiste à prendre en compte le ou les documents à la fin du processus de génération à l’aide d’un processus de fusion. L’*intégration dans les couches intermédiaires* injecte les documents encodés dans les couches intermédiaires du générateur.

Un autre élément à prendre en compte dans l’intégration des documents au cours de la génération est la fréquence et donc le nombre d’extractions effectués pour une question. Le recours à la génération peut se faire avec une fréquence

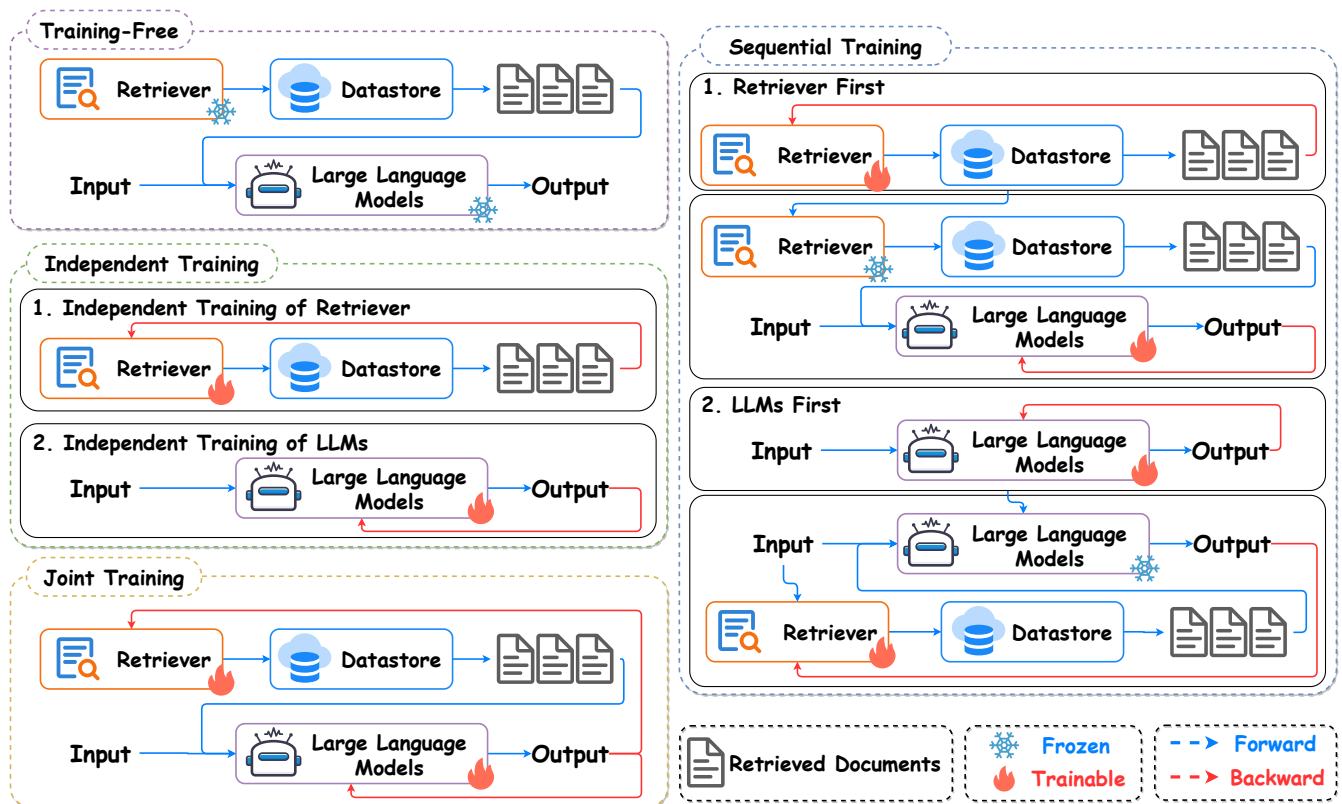


FIGURE 2.3 – Illustration des approches d'entraînement des systèmes RAG par [Fan et al., 2024]

fixée (une seule fois, tout les n token,...) ou être déclenchée au cours de la génération (par un modèle annexe, par la génération d'un token spécifique, en fonction d'un logit,...).

Entraînement Lors de la conception d'un système RAG, on assemble des modèles pré-entraînés. Une fois le système assemblé, il est possible d'affiner les composants pour les faire correspondre à la tâche visée et/ou augmenter leur symbiose. [Fan et al., 2024] distingue quatre approches d'entraînement complémentaire. La première consiste en ne pas ré-entraîner le modèle. On parle de système *training-free*. La seconde approche est l'*independent training*. L'extracteur et le générateur sont entraînés chacun de leur côté de façon totalement indépendante en dehors de la pipeline du système RAG. Le *sequential training* entraîne les deux composants l'un après l'autre mais le second est entraîné à l'intérieur de la pipeline du système RAG. Il n'y a pas d'ordre fixe pour le séquentiel training ; il peut être *Retriever-first* ou *LLMs-first*. Le *joint training* consiste à entraîner les deux modèles simultanément. Le schéma de la figure 2.3 issu de [Fan et al., 2024] illustre ces quatre approches.

2.1.3 Architectures aperçues au cours du stage

Au delà des tentatives de perfectionnement des systèmes RAG par la conception de nouvelles approches d'amélioration des entrées et des extraits et de nouvelles techniques d'intégration, cette architecture est aussi concernée par les problématiques de la gestion des données multimodales ou à flux continu, la résolution de tâches nécessitant un raisonnement long ou poussé et les enjeux d'interprétabilité et d'impact environnemental.

[Chen et al., 2025] propose un système RAG avec une base de données multi-modales où chaque entrée est un couple (image, texte). Les paires extraites à partir de la requête, elle aussi bi-modale, sont ré-ordonnées avant d'être fournies comme contexte au générateur.

Pour traiter les tâches à forte intensité de raisonnement (*reasoning intensive tasks*), [Xu et al., 2024] propose un système RAG combiné à un modèle de décomposition de raisonnement (*Chain-of-Query*). La requête de l'utilisateur est décomposé en sous-requêtes résolues une à une. Elles sont en suite vérifiées à l'aide des documents extraits. Le raisonnement est vérifié, complété et éventuellement repris en fonction du score de crédibilité obtenu. Une trace complète du raisonnement est restituée en sortie.

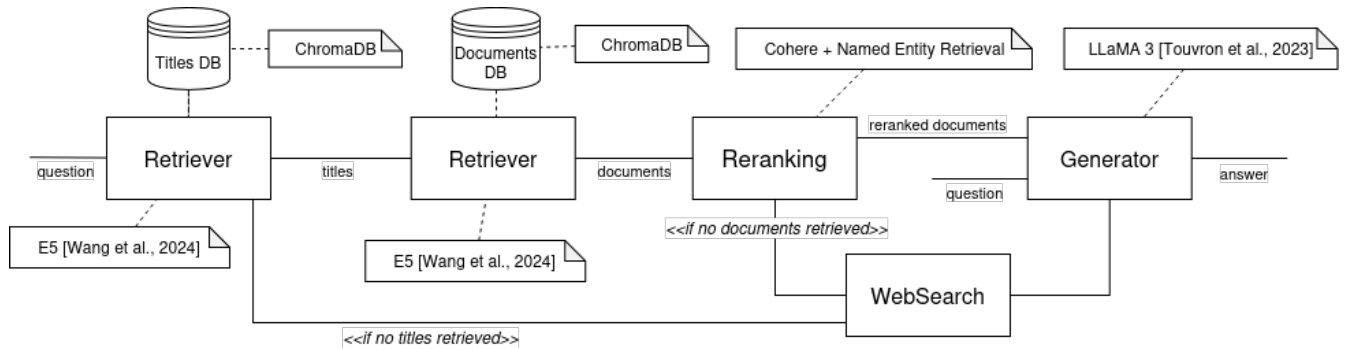


FIGURE 2.4 – Architecture de RAG4HistoricalNewspapers [Tran et al., 2024]

Les systèmes RAG sont aussi intéressants à intégrer à une architecture agentique en tant que source d'information. [Liu et al., 2025] présente un framework Hiérarchique Multi-agent Multi-modal composé d'un agent de décomposition des requêtes utilisateur qui découpe la requête entrée en sous tâches, d'un agent RAG pour des données structurées, non-structurées et sous forme de graphes à partir duquel les réponses aux sous questions sont obtenues et d'un agent de prise de décision qui juge la qualité de chaque entrée, les assemble pour obtenir la sortie finale.

[Zhu, 2025] s'intéresse au cas particulier des bases de données dynamiques. Lorsque le flux d'apport de donnée est important, le besoin de restructuration de la base de donnée est régulier or le coût calculatoire et énergétique associé est dissuasif. [Zhu, 2025] propose StreamingRAG, une architecture composée d'une base de données dynamique qui maintient un ensemble de documents (appelés prototypes) représentatifs du flux de données en l'agrémentant et la nettoyant au cours du temps. La base de données ne contient pas tous les éléments, ce qui est avantageux en terme de coût de mémoire, de calcul, d'énergie et de latence, tout en restant représentative du flux de document y compris lorsque celui-ci évolue.

2.1.4 Architecture et Modèles utilisées au cours du stage

Dans la suite, nous utilisons une version simplifiée de l'architecture RAG4HistoricalNewspaper conçue par [Tran et al., 2024]. Cette architecture est conçue pour faciliter l'exploration d'archives de journaux historiques. Comme illustré dans le Figure 2.4, un premier extracteur récupère les articles pertinents au regard de leur titre. Ces informations sont utilisées pour orienter l'extraction des extraits d'articles. Dans le cas où aucun titre ou document ne sont extraits, des ressources sont récupérées via une recherche Web. La première extraction peut être vue comme une amélioration de la requête pour l'extraction des extraits d'articles. Les deux extracteurs utilisent le modèle mE5-small de [Wang et al., 2024] et ChromaDB pour la gestion des données. Après l'extraction, les articles obtenus sont reclassés. Le score utilisé est issu d'une combinaison linéaire entre le score de Cohere accessible depuis l'API [Cohere, 2024] et le TF-IDF des titres concaténés avec les étiquettes des entités nommées de la question. Seuls les trois meilleurs documents au dessus d'un certain seuil sont conservés. Ce reclassement est une technique d'amélioration des ressources extraites. Pour finir, les documents et la question initiale sont fournis à travers un prompt template à LLaMa3 [Touvron et al., 2023] qui est utilisé comme générateur.

L'extracteur utilisé dans RAG4HistoricalNewspapers utilise l'encodeur mE5-small. La famille de modèles E5 proposée par [Wang et al., 2022] d'embedding de textes. Leur entraînement est fait par contrastive learning à partir du dataset CCPairs conçu à l'occasion pour les versions anglophones et un échantillon de datasets multilingues pour les versions multilingues [Wang et al., 2024].

2.2 Interprétabilité pour les systèmes de RAG

Les systèmes RAG et RA-LLM comme une grande partie des modèles profonds actuels sont opaques. Il est difficile voire impossible de suivre leur raisonnement lors de leur utilisation. Dans certains contextes, cette opacité est problématique car les décisions prises peuvent avoir des conséquences importantes. Cette problématique illustre le besoin d'une Intelligence artificielle de confiance.

2.2.1 Trustworthy AI

La définition d’IA de confiance (Trustworthy AI) n’est pas fixée et change souvent en fonction des contextes d’utilisations de cette expression. Dans le cas du Machine Learning, on désigne l’IA de confiance comme l’ensemble des systèmes d’IA dans lesquels il est légitime (dans le sens humainement understandable) d’avoir confiance. [Ni et al., 2025] rappelle les composantes de l’IA de confiance proposés par Le NIST (National Institute of Standards and Technology (USA)) :

Fiabilité Le comportement d’un IA de confiance doit être systématiquement adapté, l’incertitude sur son comportement doit être quantifiable et quantifié et ses capacité de généralisation doivent être robustes.

Intimité Une IA de confiance doit prévenir des risques pour les données des utilisateurs ainsi que pour les données sensibles contenues dans le modèle (issues de l’apprentissage ou de la base de donnée)

Explicabilité Le processus de décision d’une IA de confiance doit être transparent et compréhensible pour l’utilisateur.

Justesse Une IA de confiance doit minimiser les biais.

Justifiabilité Une IA de confiance doit pouvoir justifier son bon comportement et notamment sa capacité à respecter les réglementations

Sécurité Une IA de confiance doit détecter et prévenir les usages malicieux et cherchant à nuire.

Dans la suite, on s’intéresse aux différentes stratégies pour améliorer l’explicabilité des systèmes RAG et RALLM.

2.2.2 Explicabilité pour le RAG

L’explicabilité de l’intelligence artificielle, vise à rendre humainement compréhensible les raisons pour lesquelles un modèle ou système génère une certaine sortie. [Garouani et al., 2024, Bell et al., 2022].

Dans le contexte de l’explicabilité des systèmes RAG, [Ni et al., 2025] propose de distinguer l’explication du modèle d’extraction et l’explication du modèle génératif. De façon plus générale, une approche naïve d’explication des systèmes RAG est de construire des explications pour chaque composant du modèle (extraction, génération, améliorations, ...). La difficulté intervient au moment de mettre en perspective les explications fournies entre elles. [Ni et al., 2025] souligne l’absence de méthode d’explication pour l’extraction d’information spécifique contexte du RAG ni d’étude de l’application des méthodes génériques à ce cas. La même question se pose pour l’explication des modèle génératifs : les méthodes d’explications classiques sont-elles capable de prendre en compte la présence des autres composants ?

Au delà de l’objectif de transparence de l’explicabilité, [Ni et al., 2025] pointe la possible utilisation d’explication d’un composant comme indication supplémentaire pour améliorer les performances d’un autre composant.

2.2.3 Interprétabilité Mécaniste

L’explicabilité de l’intelligence artificielle est constituée de deux branches. Certains modèles contiennent leur explication. On parle de modèle *explicable by-design*. C’est le cas des premiers modèles de machine learning tels que les arbres de décision mais aussi de modèles plus récents comme les Sparse Auto-Encodeurs. Les modèles explicables by-design s’opposent à des modèles plus opaques tels que les réseaux de neurones profonds. Le processus de décision de ces modèles n’est pas directement accessible. Les méthodes d’*explicabilité post-hoc* permettent de construire des explications à partir de l’étude de leur comportement.

[Somvanshi et al., 2026] présente un sous-paradigme de l’explicabilité : l’**interprétabilité mécaniste**. L’interprétabilité mécaniste vise à identifier les parties ou sous-parties d’un modèle responsables des différents comportements du modèle. Les méthodes d’interprétabilité mécaniste vont explorer la sensibilité des paramètres du modèle sur des entrées choisies afin de construire une cartographie du modèle faisant la correspondance entre ses compétences et connaissances et ses composants ou groupe de composants. on retrouve plusieurs approches au sein des méthodes d’interprétabilité mécaniste.

Les méthodes de **manual circuit tracing** cherchent à identifier les sous-graphes de modèles neuronaux responsables d’un certain comportement recherché. Pour cela, elles observent les paramètres et les flux de données lors de l’utilisation du modèle.

Les méthodes de **representation analysis** observent les représentations latentes des couches intermédiaires grâce à des outils d’embedding. De cette manière, on peut localiser la détection de phénomènes présents dans l’entrée et la mise en place de notions dans la construction de la sortie.

Les techniques de *feature decomposition* cherchent à distinguer les différentes compétences traitées par chaque partie du modèle. Pour cela, elles intègrent des Sparse Auto-Encodeurs (SAE) sur les parties du modèle visées puis déduisent, par l’observation de ces SAE, les compétences de chaque partie du modèle.

Les méthodes basées sur les *toy models and synthetic tasks* consistent à concevoir des environnements contrôlés favorisant la localisation de motifs dans le modèle associés un comportement visé. Ces méthodes supposent qu’une même compétence prendra la même forme dans tous les modèles disposant de cette compétence.

Les méthodes *intervention-based* observent l’impact de modifications du modèle sur son comportement pour en déduire la présence et la localisation des compétences du modèle. On retrouve trois approches parmi les techniques intervention-based : les techniques d’intervention par *ablation*, les techniques d’intervention par *activation patching* que nous détaillerons dans la prochaine sous partie et les techniques de *path patching*.

2.2.4 Activation Patching

Définition et fonctionnement de l’Activation Patching

L’activation patching est une méthode d’interprétabilité mécaniste intervention-based pour les modèles profond. Elle identifie les composants d’un modèle sensibles à une famille de perturbations des entrées minimales pour changer la sortie [Geiger et al., 2021]. Pour cela, on effectue trois passes du dataset par perturbation : une sur les entrées non perturbées, une sur les entrées perturbées pour lesquelles on sauvegarde toutes les activations intermédiaires et une sur les entrées perturbées mais dans une version du modèle avec une activation remplacée par l’activation équivalente de la passe sur les données non perturbées. On s’intéresse alors à combien ce remplacement rapproche la représentation du document perturbé de celle du document non perturbé.

[Chen et al., 2024] propose une adaptation de cette méthode pour les modèle d’extraction d’information. Les entrées sont alors des paires (question, document) et la sortie, un score de pertinence. La perturbation ne doit plus nécessairement réduire la performance du modèle sur les entrées. (On peut penser à une perturbation qui duplique les mots importants de la question si ils apparaissent dans le document.) Par conséquent, la troisième passe ne se fait plus nécessairement sur les entrées non perturbées mais sur la version des entrées ayant la meilleure performance. La version de l’activation patching de [Chen et al., 2024] est formalisée dans l’encart 2.2.4.

Activation Patching pour l’Extraction d’Information [Chen et al., 2024]

Soit $Q \times D \subset \mathcal{Q} \times \mathcal{D}$ l’ensemble des paires question, document. Soit $Q \times \tilde{D}$ le même ensemble mais avec les documents perturbés à la place

1. Faire une passe pour toutes les paires $Q \times D$
 - noter o_n^e la sortie de chaque nœud $n, \forall e \in Q \times D$
 - noter p_D la performance du modèle sur les données non perturbées
2. Faire une passe pour toutes les paires $Q \times \tilde{D}$
 - noter $o_n^{\tilde{e}}$ la sortie de chaque nœud $n, \forall \tilde{e} \in Q \times \tilde{D}$
 - noter $p_{\tilde{D}}$ la performance du modèle sur les données perturbées
3. On renomme $D, e, \tilde{D}$ et $\tilde{e}$ respectivement
 - $\tilde{D}, \hat{e}, \tilde{D}$ et $\tilde{e}$ si $p_D > p_{\tilde{D}}$
 - $\tilde{D}, \check{e}, \hat{D}$ et $\hat{e}$ sinon

Pour chaque nœud n
4. Faire une passe de $Q \times \tilde{D}$ en remplaçant $o_{i,j}^{\tilde{e}}$ par $o_n^{\hat{e}}$ pour chaque $\tilde{e}$. On note la performance $\bar{p}_n$
5. $P_n = \frac{\bar{p}_n - p_{\hat{D}}}{p_{\tilde{D}} - p_{\hat{D}}}$ exprime l’impact de la perturbation sur la performance du modèle pour le nœud n

L’activation patching fourni trois performances par nœud avec lesquelles on peut estimer la sensibilité de chaque nœud à la perturbation. Si la sensibilité est élevée en valeur absolue, le nœud est sensible à la perturbation. Si la sensibilité est positive, le patch effectué atténue la perturbation. Si la sensibilité est négative, le patch accentue la perturbation.

Implémentation de l’Activation Patching

[Parry et al., 2025] propose MechIR, une implémentation en python des outils permettant d’appliquer les méthodes d’interprétabilité mécaniste à des modèles PyTorch et des modèles issus de HuggingFace. MechIR permet

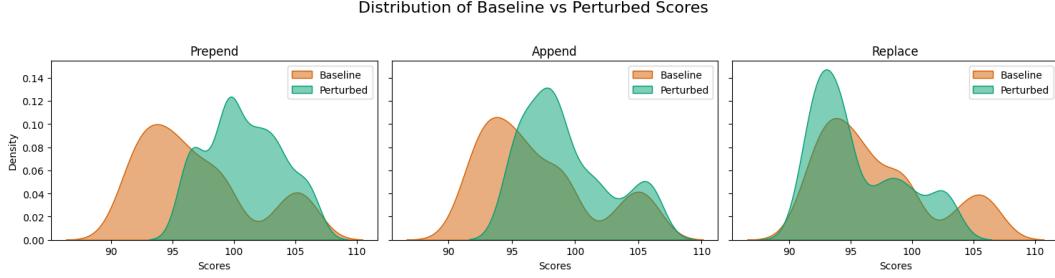


FIGURE 2.5 – Distribution des scores de pertinences pour trois perturbation sur les dataset Vaswani et le modèle TAS-B

notamment de mettre en place l’activation patching comme proposé par [Chen et al., 2024].

MechIR permet à l’heure actuelle d’appliquer trois formes de perturbation sur les documents :

- **append** : Ajoute un mot fourni en paramètre à la fin des documents,
- **prepend** : Ajoute un mot fourni en paramètre au début des documents,
- **replace** : Remplace chaque occurrence d’un mot par un autre. Les deux mots sont fourni en paramètre.

Pour qu’une perturbation soit pertinente, elle doit changer significativement la sortie du modèle (ie. perturbe significativement le score) sans dénaturer excessivement l’entrée. MechIR propose une visualisation préalable de la distribution des scores de pertinence attribué par le modèle sur les données et les données perturbées (voir Figure 2.5) pour estimer la capacité de la perturbation à impacter les scores. Une perturbation impacte significativement les sorties des paires question, document si les distributions des scores de pertinence pour les documents perturbés et non-perturbés sont significativement différentes.

Par exemple, la Figure 2.5 visualise les distributions des scores de pertinence pour trois perturbations (ajouter *microwave* au début du document, ajouter *microwave* à la fin du document et remplacer *microwave* par *toaster*). Le dataset utilisé est une partie du dataset Vaswani issu de la librairie irdataset [Hou et al., 2025], les extraits manipulés sont des abstracts d’articles de physique et le modèle analysé est TAS-B [Hofstätter et al., 2021]. L’axe des abscisses correspond à la valeur prise par les scores et l’axe des ordonnées à leur densité. Plus la densité est élevée, plus il y a de scores proche de la valeur en ordonnée. La distribution des scores de pertinence des documents vis à vis des question est représentée en orange et celle des scores pour les documents perturbés est en vert. La distribution des scores pour les documents non perturbés est identique pour les trois perturbations étant donné qu’on manipule le même dataset. Les perturbations de type **append** et **replace** impactent un peu les distribution des scores mais elles continuent à suivre la même tendance. La perturbation de type **prepend** impacte plus la distribution des scores de pertinence : les densité sont élevées pour les scores différent et le profil de la courbe est très différent pour les scores de documents perturbés et non perturbés.

Une fois la perturbation choisie, on peut appliquer l’activation patching pour chaque nœud du modèle choisi et visualiser les sensibilité à la perturbation de chacun d’entre eux (voir Figure 2.6). Chaque nœud du modèle est identifié à l’aide ses coordonnées (numéro de couche, position dans la couche). Le score du nœud correspond à la moyenne des sensibilité obtenu pour chaque document. Plus le nœuds à un score élevé en valeur absolue (ie. plus la couleur est foncée), plus la perturbation impacte son comportement. Cela signifie que la modification porte sur une information utilisée par le nœud dans le modèle. Si le score est positif (resp. négatif), le patching atténue (resp. accentue) la perturbation en moyenne.

Dans la Figure 2.6, on remarque principalement le nœud (1,6). Une interprétation possible est que le patching du nœud diminue fortement l’estimation de pertinence effectuée par le modèle sur les documents non perturbé.

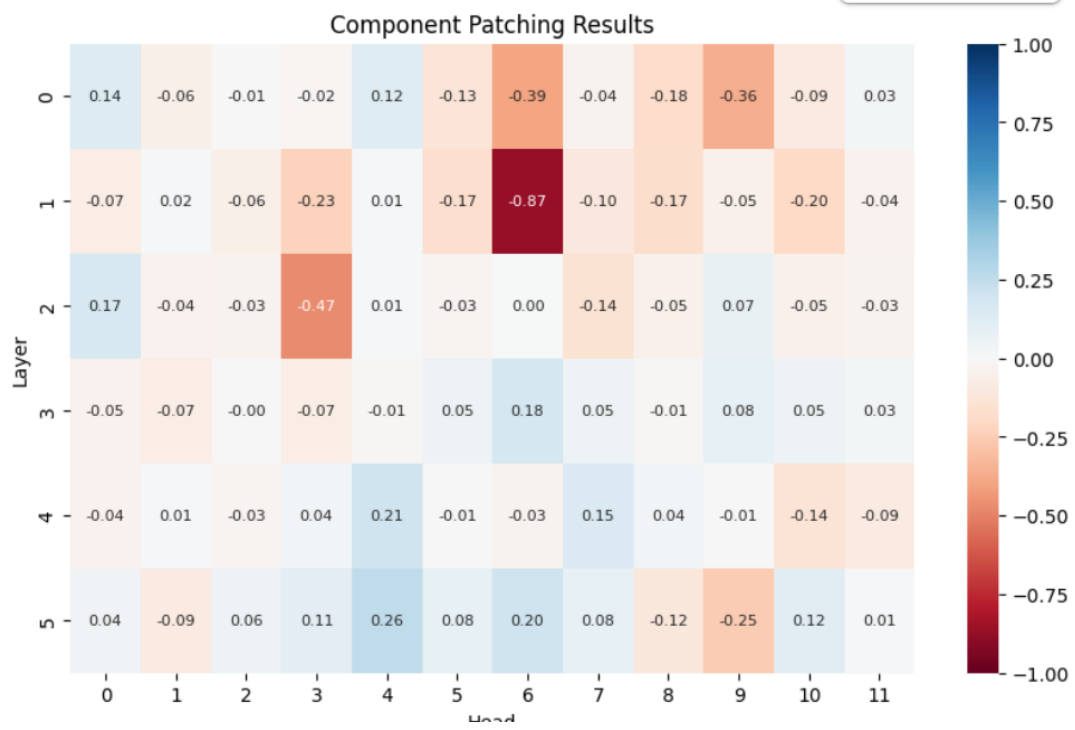


FIGURE 2.6 – Matrices de sensibilité des nœuds du modèle TAS-B pour la première perturbation de l'exemple

Chapitre 3

Objectifs du stage et préliminaires

3.1 Objectifs du stage

La langue de spécialité correspond à l'ensemble du vocabulaire et des tournures spécifiques à un domaine d'expertise [Cabr , 1998]. Si la fa on de s'exprimer ne change que de fa on limit e, certains mots ont parfois un sens diff rent (exemple : *n uds* en deep learning) et certaines fa ons de s'exprimer sont inhabituelles en dehors du domaine (ex : codes de r daction d'une preuve math matique). Chaque domaine dispose de sa langue de sp cialit  qui est incluse dans la langue g n rale et en intersection avec la langue commune.

Hypoth ses Consid rant que l'activation patching permet de localiser les composants et n uds d'un mod le sensibles   une perturbation, on s'int resse aux hypoth ses suivantes :

H1 : Dans les mod les de langue, une langue de sp cialit  repose sur certains composants du mod le.

H2 : Ces composants peuvent  tre cartographi s   l'aide de l'activation patching.

Objectifs Ce stage vise   appliquer l'activation patching   la localisation du vocabulaire sp cifique de la langue de sp cialit  (vocabulaire de sp cialit ) du domaine de la d fense. L'objectif est de proposer une m thodologie, un cadre de mise en pratique de l'activation patching pour la localisation du vocabulaire de sp cialit    l'aide de l'outil MechIR ainsi qu'une application au cas de la langue de sp cialit  de la d fense sur le mod le mE5-small.

Int gration dans l'explicabilit  post-hoc des mod les d'extraction d'information et des syst mes RAG

La possibilit  de localiser le vocabulaire de sp cialit  ouvre la voie   la localisation de concepts et/ou domaines pr cis dans un mod le de langue. Il devient alors possible de cartographier ces mod les. Ces cartographies pourraient   terme  tre utilis es pour expliquer le raisonnement, les  l ments pris en compte dans l'entr e et dans les documents extraits au sein d'un syst me RAG ou RA-LLM.

3.2 Challenge RAG@EvalLLM

Le sujet de l'extraction d'information et du RAG est d'actualit  au sein de la recherche en machine learning   l'international. Son application au fran ais et son  valuation anime les recherches en France.   l'occasion de l'atelier EvalLLM2026 qui portait sur l' valuation des mod les g n ratifs et du RAG, ont  t  organis s deux challenges. L'un d'entre eux portait sur le RAG. La premi re t che  tait une t che d'extraction d'information et la seconde t che portait sur l'attribution de sources (  partir d'une question, d'extraits de documents et d'une r ponse, associer   chaque fragment de la r ponse un document d'o  l'information est issue). Ce challenge mettant   disposition une grande quantit  de documents en fran ais sp cifiques   un domaine (la d fense) et un protocole d' valuation des r sultats, je me suis inscrite   la premi re t che pour donner un cadre   la construction du mod le et du protocole d'am lioration.

La t che d'extraction d'information demandait de r pondre   des requ tes sous la forme d'une ou plusieurs questions complexes ou d'une suite de mots cl s sans s parateur   l'aide d'une base de documents et fournir les pages des documents utilis s pour r pondre ainsi que leur rang de pertinence. La base de documents contenait en grande partie des PDF contenant parfois des images. Les documents avaient diverses origines, ne suivaient pas structure commune particuli re et  taient en majorit  en fran ais (quelques documents  taient en anglais ou

disponibles dans les deux langues). Le format des entrées et des sorties était fourni à l’avance. Les documents étaient accessibles dès l’inscription mais les questions n’étaient communiquées que durant les trois jours de test (à l’exception des cinq questions présentes dans l’exemple de format des données). Aucun dataset étiqueté n’était fourni.

Je souhaitais présenter une version améliorée du modèle mE5 à l’aide de l’activation patching afin de mettre en jeu les hypothèses présentées précédemment. Malheureusement, la méthodologie n’étant pas au point, je n’ai pas eu le temps d’aller jusqu’au bout du challenge et de cet objectif.

3.3 Construction d’un dataset adapté

Les données fournies par le challenge étant brutes, un premier traitement a été effectué. Les PDFs ont été convertis au format Markdown à l’aide de PyMuPDF4LLM [Artifex Software, 2026] pour les stocker dans base de donnée. Ce format est lisible pour l’humain et est souvent utilisé en entrée et sortie des grands modèles de langue. Les images n’ont pas été conservées car nous ne nous intéressons pas au traitement de données multi-modales. Ce pourrait être une perspective d’évolution dans la conception du modèle. Les textes récupérés sont découpés en pages et en chunks plus petits pour les pages contenant plus de 500 tokens. Le découpage prend en compte la structure Markdown (séparation entre paragraphes, avant un nouveau titre,...)

La construction de la base de donnée a nécessité plusieurs itérations. Les différentes versions sont toutes au format CSV avec des champs d’identification de la question (identifiant) et de l’extrait (nom du fichier, identifiant, numéro de page, numéro de chunk dans le document), la question et l’extrait du document. Pour la base de donnée du modèle et son utilisation, les questions et les documents sont fournis dans deux fichiers CSV distincts pour toutes les versions. La base est mise sous forme de paires (produit cartésien) pour l’utilisation de MechIR et l’application de l’activation patching.

La première base de donnée est une base naïve avec toutes les questions et tous les extraits des documents fournis. Pour effectuer les manipulations avec MechIR, il faut manipuler le produit cartésien des questions et des extraits, soit une base de plus de 27 million d’entrées, surdimensionnée pour l’utilisation. De plus, du fait du produit cartésien, la majorité des documents ne sont pas pertinents pour la question. La base de donnée n’est pas équilibrée. Cela rend l’utilisation de MechIR et de l’activation patching moins accessible car plus difficile à interpréter. Dans un premier temps, la base de donnée a été échantillonnée pour ne garder qu’une petite partie mais la base de donnée obtenue n’est toujours pas équilibrée.

Pour résoudre le problème de l’équilibre, la seconde base de donnée est construite artificiellement à partir des extraits et des questions précédentes. Pour chacune des 26 questions, un extrait pertinent et un extrait non pertinent ont été choisis. La base de donnée obtenue est équilibrée, étiquetée et de taille très raisonnable. Pour autant, la diversité des mots utilisés et des sujets abordés rend difficile la construction d’une perturbation qui impacte la sortie sans dénaturer les documents.

La troisième base de donnée a été construite à partir d’une seule question. Ainsi, les extraits pertinents partagent un plus grand nombre de mots sur lesquels il est possible de s’appuyer pour construire des perturbations. Pour maintenir l’équilibre, les extraits sont pour moitié non pertinents. Le choix des extraits est fait à partir de trois PDFs : un pour les extraits pertinents et deux pour les extraits non pertinents.

Le troisième dataset est la version remaniée des données du challenge RAG@EvalLLM qui est utilisée pour concevoir les propositions méthodologiques du chapitre suivant. C’est sur ces données que les exemples présentés sont appliqués. La question retenue est la suivante :

Quelles sont les trois principales catégories de limitations techniques affectant les systèmes de drones, et comment chacune influence-t-elle l’efficacité opérationnelle de ces vecteurs ?

Les extraits sont issus de trois PDF portant respectivement sur les limitations matérielles du drone, le code de la défense concernant les matériel de guerre et les fausses informations dans le contexte des Jeux Olympiques et Paralympiques. Le dataset final contient 28 extraits dont 14 extraits pertinents vis à vis de la question, tous issus du premier PDF.

3.4 Adaptation du modèle RAG4HistoricalNewspapers

Pour répondre au challenge, nous avons besoin d’une architecture RAG complète fonctionnelle avec un extracteur sur lequel appliquer l’activation patching. Le stage ne portant pas sur la conception d’une architecture, nous avons repris et simplifié l’architecture proposée par [Tran et al., 2024] pour l’adapter à la première tâche du

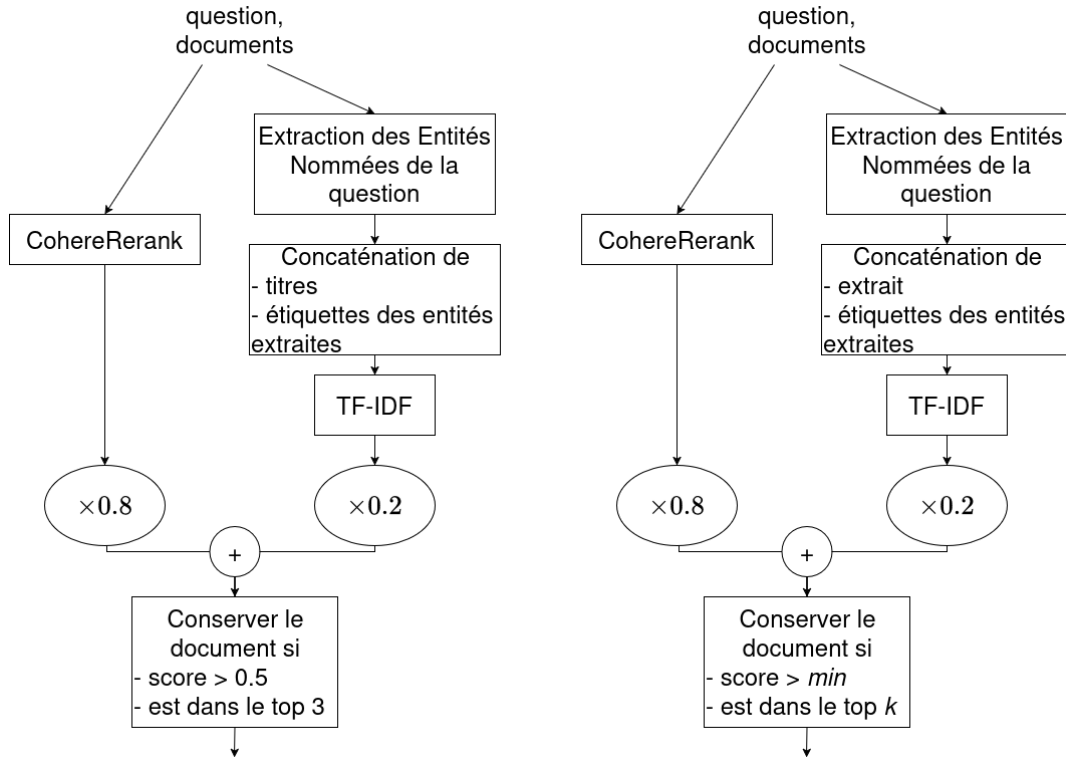


FIGURE 3.1 – Calcul du score de pertinence pour le reclassement des extraits de RAG4historicalNewspapers à gauche et de notre version simplifiée à droite

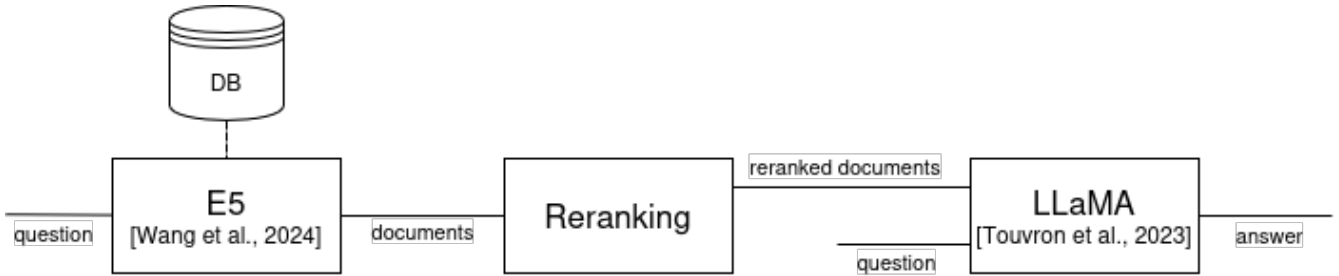


FIGURE 3.2 – Version simplifiée de RAG4HistoricalNewspapers utilisée pour le challenge

challenge RAG. Le choix de cette architecture est motivé par la disponibilité du code source et le fait qu'elle est déjà adapté aux documents en français.

Les données du challenge n'étant pas au format (titre, document), la première extraction a été supprimée. Pour la même raison, le reclassement des textes extraits a été revu : la composante du TF-IDF est appliquée sur les extraits au lieu des titres (voir Figure 3.1). Le module de récupération de document sur le web n'étant pas compatible avec le challenge, il a aussi été écarté.

L'architecture obtenue est illustrée par la Figure 3.2.

Chapitre 4

Contribution méthodologique

4.1 Construction des perturbations

Dans un premier temps, on se concentre sur les perturbations de type **replace**. Cela permet d'exploiter la multiplicité des sens (polysémie) de certains mots de spécialité en jouant sur leur sens de spécialité et leur sens commun ou issu d'une autre spécialité.

Pour construire des perturbations de type **replace** qui impactent le modèle et qui soient pertinentes à interpréter, il faut bien choisir le mot à remplacer et le mot remplaçant. On propose aussi un nouveau type de perturbation.

4.1.1 Identification des mots cibles

Pour qu'une perturbation de type **replace** impacte de façon pertinente le modèle, elle doit remplacer des mots bien choisis. On propose les critères suivants pour sélectionner des mots cibles pour impacter le modèle :

- Fréquence du mot dans le dataset
- L'IDF du mot dans le dataset
- La présence du mot dans la question ciblée

Les indicateurs de fréquence permettent de faciliter l'identification du vocabulaire de spécialité. On peut aussi s'aider d'un dictionnaire. Un intérêt particulier doit être porté aux mots ayant un sens spécifique ou différent dans le contexte de la spécialité visée. Ces mots permettent d'exploiter la polysémie lors de la construction de perturbations.

Mot	Fréquence	TF-IDF
matériel	16	0.063
système	11	0.054
limitation	6	0.050
drone	10	0.044
défense	10	0.044
prendre	6	0.043
compte	5	0.040
action	8	0.040
article	9	0.040
ministre	8	0.039
aérien	7	0.037
acteur	6	0.034
guerre	8	0.034
informationnel	5	0.031
information	6	0.030

TABLE 4.1 – Mots importants du troisième dataset d'exemple

La table 4.1 présente, après lemmatisation des documents, les statistiques du vocabulaire présent pour les 15 plus grands TF-IDF. Tous les mots ayant un fort TF-IDF ne sont pas intéressants à remplacer. On se concentre sur les mots appartenant au vocabulaire de la défense et disposant d'au moins un autre sens. On inclut aussi les mots

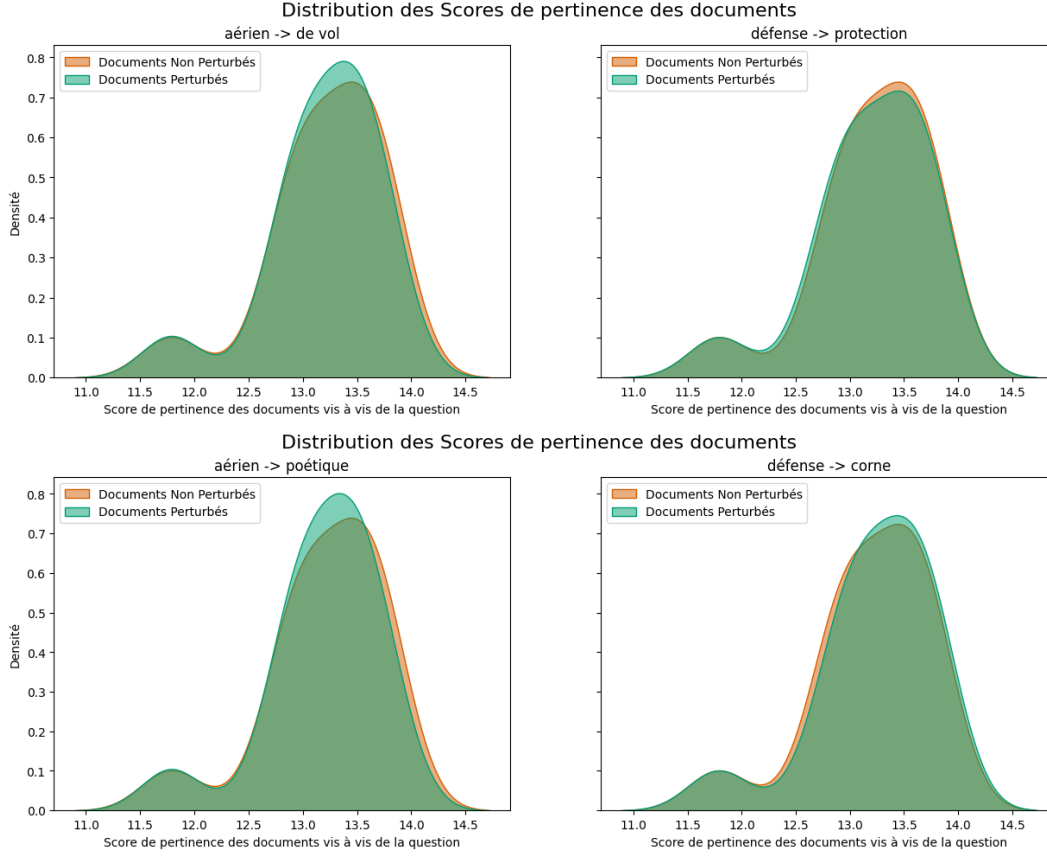


FIGURE 4.1 – Distribution des scores de pertinence pour les exemples de perturbation de niveau 1 et 2

porteurs de sens ou d’information de la question du dataset. En les modifiant, on s’attend à un fort impact sur la pertinence du document. On choisit de se concentrer sur les mots indiqués en gras ainsi que sur le mot *technique* car il est porteur de sens dans la question.

4.1.2 Perturbations de type replace basées sur les synonymes

Une fois le mot à remplacer choisi, on propose dans un premier temps d’exploiter la polysémie (les multiples sens) de ce mot. On distingue deux groupes de candidats : les synonymes (ou mot proche) du mot remplacé partageant le sens du vocabulaire de spécialité et les synonymes (ou mot proche) du mot remplacé ayant un autre sens. Ces groupes de mots remplaçants reflètent deux niveaux de perturbation. Les perturbations de niveau 1 remplacent le mot par un synonyme du mot remplacé ayant le sens du vocabulaire de spécialité ciblé. Par exemple, dans le domaine du droit, remplacer *défense* par *interdiction* est une perturbation de niveau 1. Les perturbations de niveau 2 remplacent le mot ciblé par un mot proche mais dans un autre sens que le domaine de spécialité visé. Dans l’exemple précédent, remplacer *défense* par *corne* est une perturbation de niveau 2.

Les seconde et troisième colonnes de la table 4.3 présentent les mots remplaçants retenus pour les mots ciblés choisis dans l’exemple de la sous-section précédente. Dans le contexte de la défense, *aérien* qualifie des véhicules ou objets se déplaçant en volant. L’expression *de vol* est un bon candidat pour construire une perturbation de niveau 1. Sur le même principe, le secteur de la défense vise à protéger un territoire et/ou une population. Remplacer *défense* par *protection* est une perturbation de niveau 1. À contrario, qualifier un mouvement ou une démarche d’aérien ou de poétique ne correspond pas à l’usage fait dans la défense. *poétique* est un mot remplaçant adapté pour une perturbation de niveau 2. De même, la défense d’un éléphant peut être comparée à une corne de rhinocéros mais le sens donné à *défense* dans ce contexte n’est pas le même. Remplacer *défense* par *corne* est une perturbation de niveau 2. Les distributions des scores de pertinence des documents pour ces quatre perturbations sont disponibles dans la figure 4.1.

Les perturbations de niveau 1 et 2 listés dans la table 4.3 n’impactent pas suffisamment le modèle. Cela nous a surpris car les résultats présentés par [Parry et al., 2025] exploitaient aussi la polysémie et donnaient des pertur-

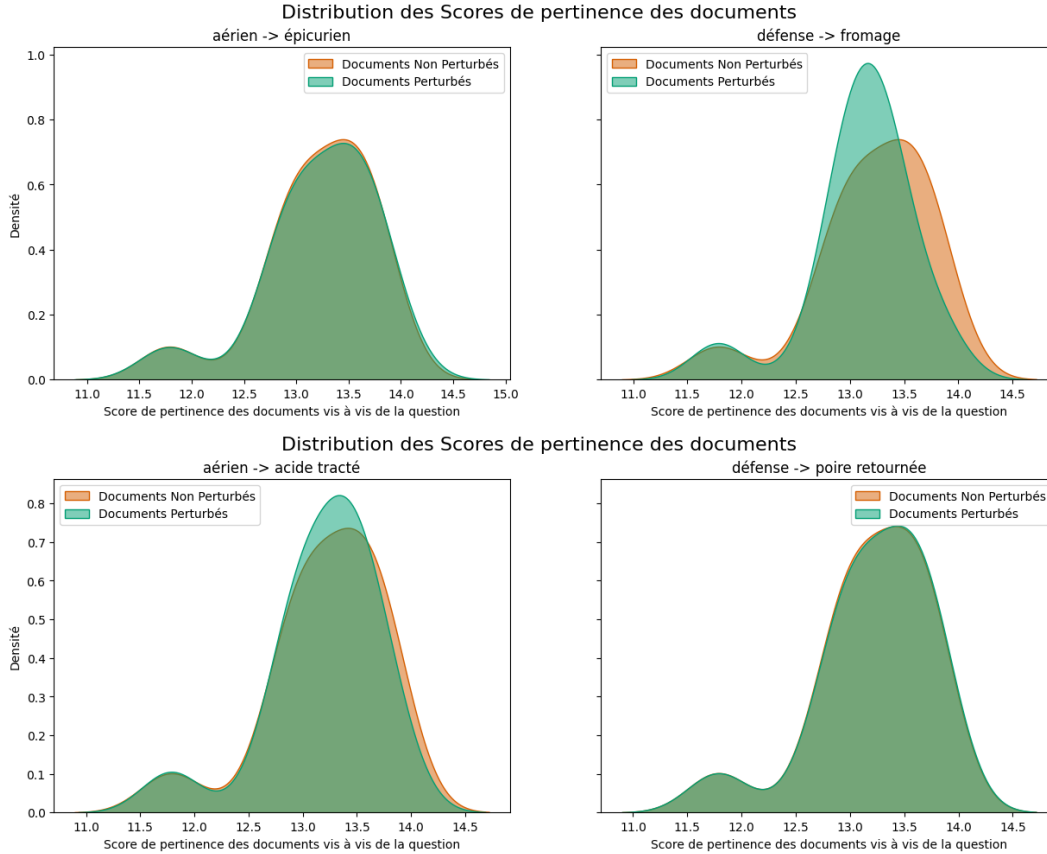


FIGURE 4.2 – Distribution des scores de pertinence pour les exemples de perturbation de niveau 3 et 4

bations plus impactantes. En prenant un peu de recul, les grands modèles de langage semblent pouvoir manipuler différents sens pour un mot. Cela était moins le cas pour les modèles peu profonds tels que Word2Vec qui tendent à regrouper tous les sens ou les écraser au profit du plus présent. Il est possible que les perturbations exploitant la polysémie (perturbations de niveau 2) laissent suffisamment d’indices pour que les grands modèles gèrent cette difficulté. Cela pourrait expliquer le faible impact des perturbations de niveau 2 sur mE5.

4.1.3 Perturbations de type replace plus agressives

On propose deux niveaux de perturbation supplémentaires effectuant des remplacements plus prononcés. Les perturbations de niveau 3 remplacent le mot ciblé par un mot n’ayant aucun lien avec les différents sens du mot ciblé.

On conserve juste la même nature entre le mot remplacé et le mot remplaçant (les noms sont remplacés par des noms, les adjectifs par des adjectifs,...). Ces perturbations ne donnent aucune piste sur le mot initialement présent ce qui rend la compréhension plus difficile. Les perturbations de niveau 4 sont encore plus agressives. Le mot est remplacé par un groupe de mot absurde (ex : luge pontificale). Ces perturbations masquent les mots cibles qui sont identifiés comme importants par un groupe de mot qui n’est que du bruit.

Les quatrième et cinquième colonnes de la table 4.3, donnent les perturbations de niveau 3 et 4 retenues pour les mots à remplacer choisis. Les mots remplaçants choisis pour les perturbations de niveau 3 restent de même nature (*épicurien* est un adjectif comme *aérien*,...) mais sont sémantiquement éloignés du mot remplacé. Les groupes nominaux remplaçant pour les perturbations de niveau 4 sont bien des concepts qui ne sont pas réels. Les distributions des pertinences sont visualisées dans la figure 4.2. On remarque d’ailleurs que les perturbations choisies n’impactent pas toujours le modèle autant qu’attendu.

La table 4.2 récapitule les 4 niveaux de perturbation et les phénomènes observés.

Niveau	Caractéristiques du mot remplaçant	Résultat attendu	Caractéristique de nœud ou composant visé dans le modèle
1	synonyme (ou mot proche) ayant le sens du vocabulaire de spécialité visé	Faible perturbation	Détection et prise en compte du mot ciblé spécifiquement
2	synonyme (ou mot proche) ayant un sens différent de celui lié au vocabulaire de spécialité visé	Perturbation marquée. Quelques nœuds sensibles lors de l'application de l'activation patching	Détection et prise en compte du sens du vocabulaire de spécialité
3	mot sans lien avec aucun sens du mot cible	Perturbation marquée. Quelques nœuds sensibles lors de l'application de l'activation patching	Prise en compte du sens de spécialité du mot ciblé.
4	groupe de mot absurde sans lien avec la mot cible	Perturbation très marquée. Quelques nœuds sensibles lors de l'application de l'activation patching	Prise en compte du sens de spécialité du mot ciblé.

TABLE 4.2 – Récapitulatif des niveaux de perturbation proposé

Mot remplacé	Niveau 1	Niveau 2	Niveau 3	Niveau 4
aérien	de vol	poétique	épicurien	acide tracté
article	sujet	déterminant	table	luge pontificale
défense	protection	corne	fromage	poire retournée
drone	vecteur	bourdon	gazon	ver de verre
limitation	contrainte	bornage	animation	plastique anarchique
matériel	outil	concret	post-it	porte sans poivre
système	dispositif	ensemble	panneau	esperluette en vent
technique	spécialisé	difficile	magique	feuille entubée

TABLE 4.3 – Perturbations proposées pour les mots à remplacer sélectionnés pour chaque niveau évoqué

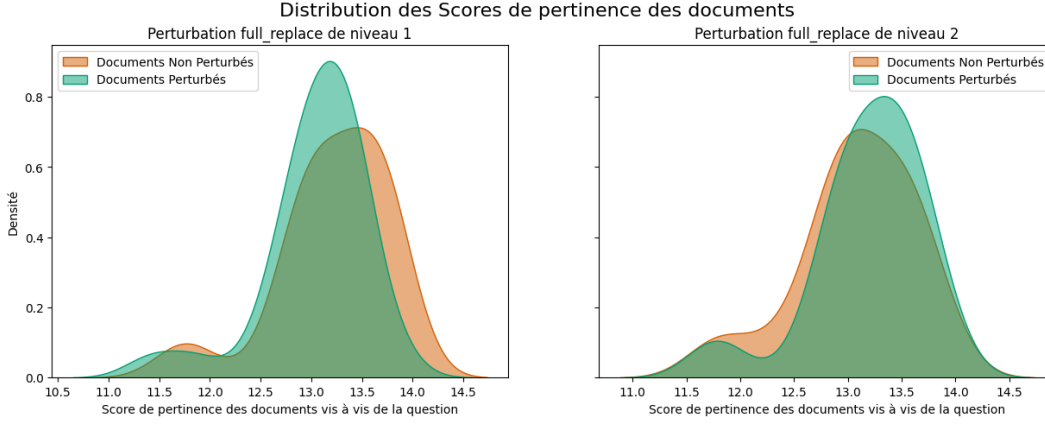


FIGURE 4.3 – Distribution des scores de pertinence pour les exemples de perturbation de type **full_replace** de niveau 1 et 2

4.1.4 Perturbations de type **full_replace**

Une autre piste explorée pour augmenter l’impact des perturbations basées sur les synonymes est de cumuler plusieurs remplacements en même temps. On propose le nouveau type de perturbation **full_replace** qui applique une liste de perturbations de type **replace** en une fois sur le document en guise de perturbation. On étend les quatre niveaux de perturbation aux perturbations de type **full_replace** en considérant le niveau des perturbations **replace** qu’elle inclus.

Au delà de l’augmentation de l’impact de la perturbation sur le modèle, les perturbation **full_replace** permettent de s’affranchir du bruit introduit par chaque mot individuellement et de se concentrer sur un ensemble de composants du modèle sensible au vocabulaire de spécialité de façon plus globale.

La Figure 4.3 représente les distributions des scores de pertinence pour deux perturbations de type **full_replace**. La première perturbation applique les 8 perturbation **replace** de niveau 1 listées dans la table 4.3. Par exemple, l’extrait

Certaines limitations sont plus particulièrement liées aux caractéristiques du vecteur aérien et n’affectent donc pas toutes les catégories de drones de la même manière.

devient

Certaines contraintes sont plus particulièrement liées aux caractéristiques du vecteur de vol et n’affectent donc pas toutes les catégories de vecteurs de la même manière.

lorsqu’on applique cette perturbation. Les mots *limitations*, *aérien* et *drones* ont tous les trois été remplacés. On peut considérer que cette perturbation est de niveau 1 car tous les remplacements effectués sont de niveau 1. La seconde perturbation regroupe les perturbation **replace** de niveau 2 proposées dans la table 4.3. C’est donc une perturbation de niveau 2. On constate que les perturbations de type **full_replace** impactent plus fortement la distribution des scores de pertinence que les perturbation **replace**.

Lorsqu’on y regarde de plus près, une autre force des perturbations **full_replace** est leur capacité à couvrir plus facilement l’ensemble du dataset. Pour qu’une perturbation impacte le modèle, elle doit modifier les documents. Si une perturbation **replace** modifie qu’une partie des documents alors elle aura moins d’impact sur le modèle. Les documents non modifiés auront un score ramené à zéro (la division par zéro induit un NaN qui est interprété comme un zéro dans le calcul de la moyenne des scores de sensibilité pour tout le dataset). Les perturbations **full_replace** touchent plus de document car elles modifient l’union des documents modifiés par chacune d’entre elles.

4.2 Compléments pour l’estimation de la pertinence d’une perturbation

On propose de compléter l’estimation de la pertinence d’une perturbation proposée par [Parry et al., 2025] à l’aide de deux autres visualisations qui mettent les perturbations en perspective vis à vis des données et entre elles.

La première visualisation complémentaire est la visualisation des scores de pertinence en séparant les documents pertinents et non pertinents vis à vis de la question. Cela permet de distinguer les effets de la perturbation sur chaque groupe de documents. On peut notamment d’identifier un impacts plus important des perturbations sur l’un

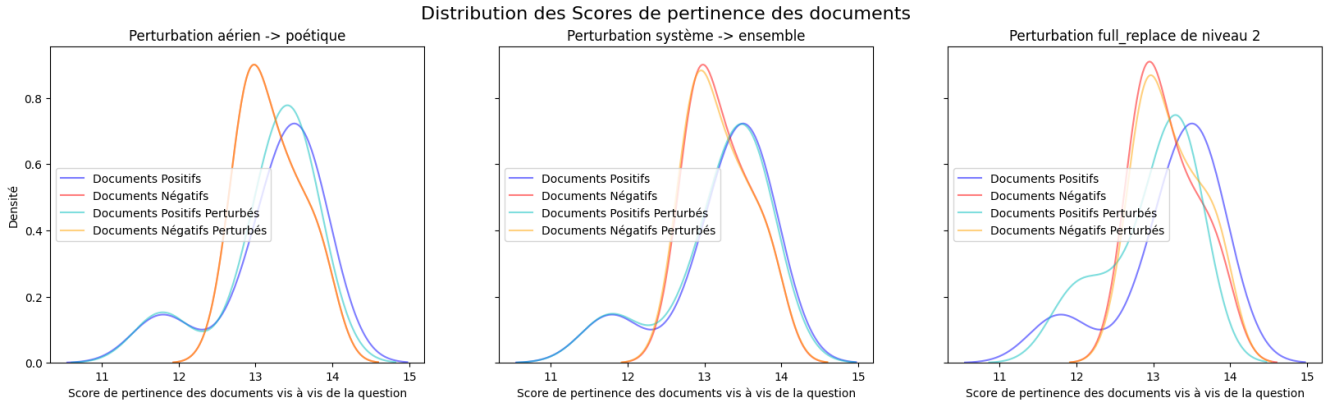


FIGURE 4.4 – Distribution des scores de perturbation pour les documents positifs/négatifs (non-)perturbés pour trois perturbations

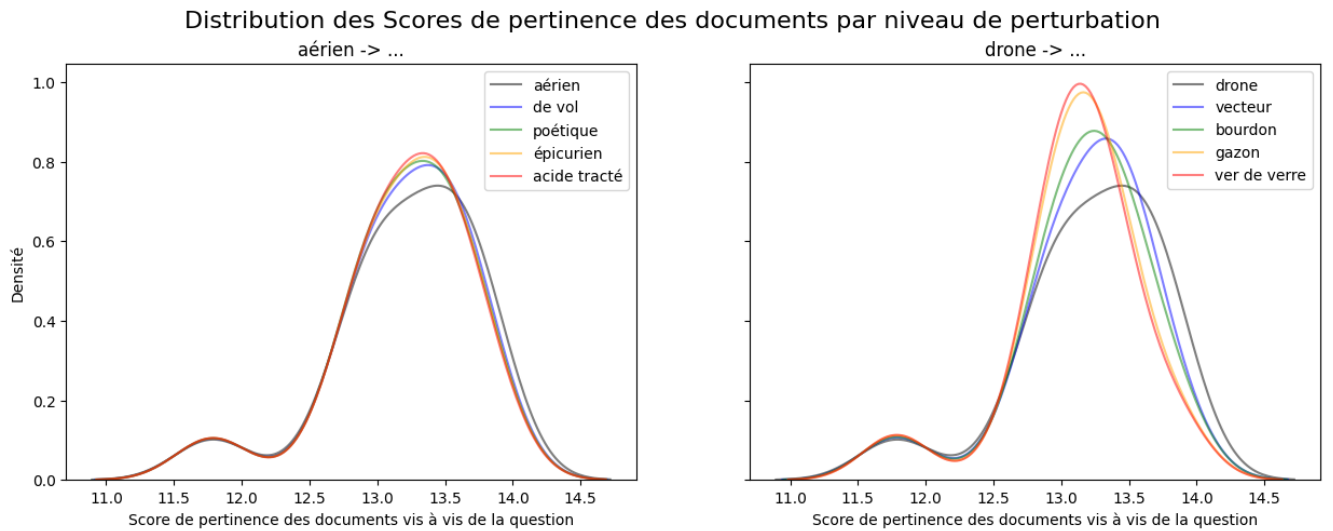


FIGURE 4.5 – Distribution des scores de pertinence des documents pour les mots remplacés *aérien* et *drone*. La distribution des scores des documents non-perturbés et pour les 4 niveaux de perturbation appliqués sont regroupés dans le même graphique pour chaque mot

des deux groupes (si le mot remplacé est plus présent dans les documents pertinents par exemple). Cette information pourra donner un éclairage différent sur la perturbation ou les nœuds identifiés comme sensibles au patching.

Par exemple, dans la figure 4.4, on s'intéresse au changement de distribution des scores pour les perturbations "remplacer *aérien* par *poétique*", "remplacer *système* par *ensemble*" et "remplacer tous les mots cible choisis par leur mot remplaçant de niveau 2" (voir la table 4.3). On constate que les documents non pertinents vis-a-vis de la question ne changent pas de score de pertinence lorsqu'on applique la perturbation *remplacer aérien par poétique* au dataset. Cela s'explique par le fait qu'aucun document non-pertinent ne contient le mot *aérien*. On note que ce phénomène est assez répandu à cause de la méthode de construction du dataset. En effet, sélectionner des extraits issus de différents PDF portant sur différents sujets en lien avec la défense force les extraits à contenir un vocabulaire spécifique au sujet du PDF qui n'est pas nécessairement le même pour tous les PDFs utilisés. Dans notre cas, le seul mot remplacé ne subissant pas ce phénomène est *système*.

La seconde visualisation complémentaire est la visualisation des distributions des scores à travers les différentes perturbations d'un même mot et à travers les différents niveaux de perturbation du mot. Cela permet de mettre en perspective les différentes perturbations et éventuellement de sélectionner la plus impactante au sein d'un même niveau.

La visualisation des distributions des scores de pertinence des documents pour chaque perturbations appliquées à un même mot comme dans la figure 4.5 permet de constater l'évolution de la distribution en fonction des niveaux

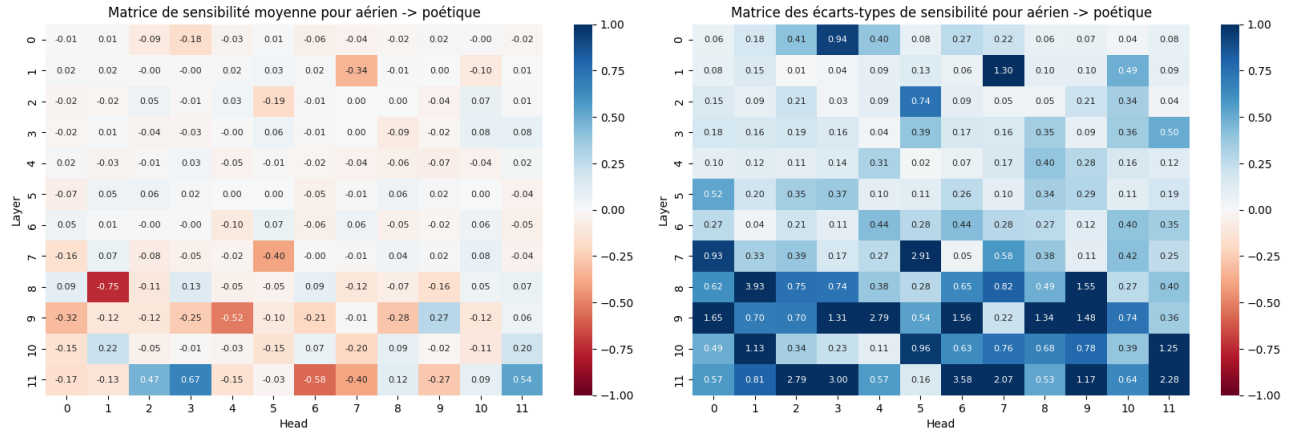


FIGURE 4.6 – Matrice de sensibilité moyenne et leurs écarts-types des nœuds pour la perturbation "remplacer *aérien* par *poétique*"

de perturbation appliqués. On constate qu'elle n'est pas toujours très marquée.

4.3 Interprétation des résultats de l'Activation Patching

Un fois la ou les perturbations choisies, on peut appliquer l'activation patching à l'aide de l'outil MechIR. On propose dans cette partie des visualisations des résultats supplémentaires pour faciliter et/ou éclairer l'analyse et la cartographie du modèle.

4.3.1 Complément à la visualisation de la sensibilité du modèle à la perturbation

Exploitation de l'écart-type Pour rappel, [Parry et al., 2025] propose de visualiser la moyenne des sensibilités du nœud à la perturbation appliquée sur chaque document. On propose de compléter la visualisation des moyennes par celle des écarts-types. Les moyennes de sensibilité peuvent être élevées (en valeur absolue) à cause de rares documents qui modifiés impactent fortement le modèle. À contrario, certaines moyennes de sensibilité pourraient être proche de zéro à cause d'un effet de compensation entre les scores pour différents documents. Les écarts-types peuvent aider à distinguer les nœuds sensibles de façon homogène (à travers les documents) à une perturbation des nœuds subissant de façon plus importante du bruit.

La figure 4.6 représente la sensibilité moyenne de chaque nœud de mE5 à la perturbation "remplacer *aérien* par *poétique*" à gauche et l'écart-type des sensibilités pour chaque documents pour les nœuds. On identifie bien les nœuds (8,1), (11,3), (9,4), (11,6) et (11,11) comme sensibles à l'aide de la matrice de sensibilités moyennes. Le nœud (11,2) est plus facilement identifiable à l'aide de la matrice des écarts-types de sensibilité. Cette matrice permet aussi de confirmer les premiers nœuds identifiés.

Adaptation de la visualisation des résultats de l'activation patching D'une perturbation à l'autre, les nœuds ont des sensibilité moyennes très variables. Dans certains cas, les sensibilités ont un très faible écart-type (à travers la matrice) et dans d'autres, plusieurs nœuds saturent l'échelle de couleur proposée par MechIR. De la même façon, certains écarts-types explosent. De plus, par définition, un écart-type est positif. La moitié de l'échelle n'est alors par utilisée.

Pour faciliter la visualisation, on propose de compléter la diversification des matrices de valeurs par une diversification des échelles de couleur. Pour maintenir un accès aux valeurs exactes, on propose une échelle adaptative allant de -50 à 50 dans le cas de matrices de sensibilité ayant des nœuds avec un score de sensibilité moyen supérieur à 20. Pour les autres cas, on maintient l'échelle -1;1. On propose de conserver la même échelle de couleur mais de l'inverser afin que les valeurs positives soient en rouge et les valeurs négatives en bleu.

En complément des matrices contenant les valeurs réelles, on propose une visualisation des valeurs ramenées dans l'intervalle -1;1 en maintenant zéro à sa valeur (division de toutes les valeurs par la sensibilité maximale en valeur absolue). On conserve la même échelle de couleur. Une telle visualisation permet de mettre en perspective les matrices d'une perturbation à l'autre : on peut comparer les nœuds ayant les sensibilités moyennes les plus élevées. D'autre part, cette approche permet d'identifier les nœuds significativement plus sensibles car ils seront plus foncés

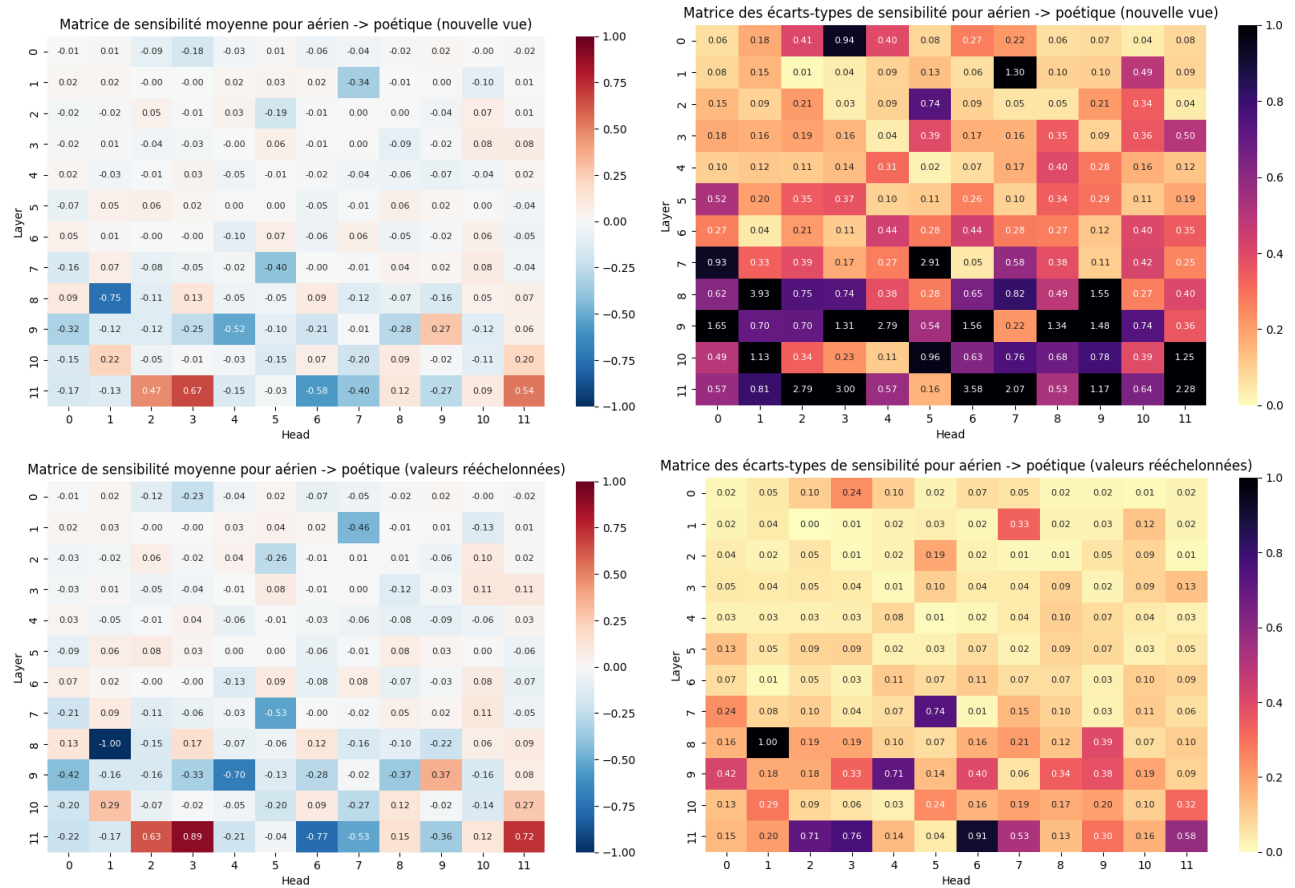


FIGURE 4.7 – Matrice de sensibilité moyenne et leurs écarts-types des nœuds pour la perturbation "remplacer *aérien* par *poétique*" avec la nouvelle visualisation (en haut) et avec le rééchelonnage (en bas)

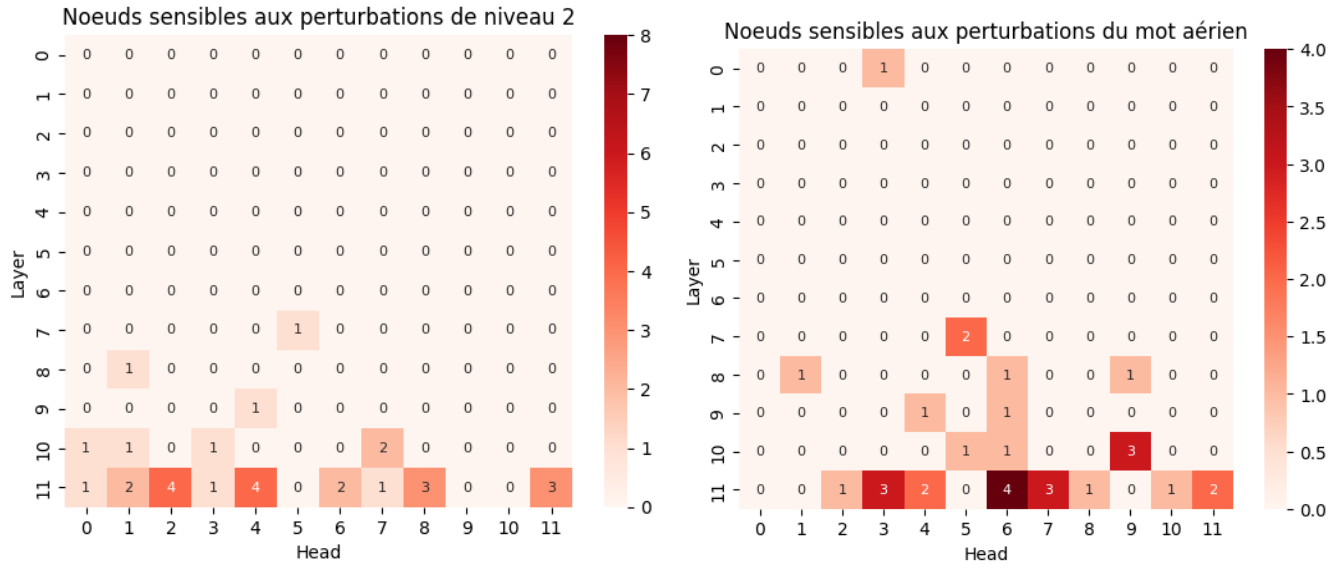
que les autres (valeurs proches de 1 en valeur absolue) et "se détachent" du reste des nœuds. Pour le cas des écarts-types, on propose les mêmes échelles mais en tronquant à 0 et avec la gamme de couleurs Noir-Violet-Orange-Jaune.

La figure 4.7 affiche les mêmes données que la figure 4.6 en appliquant la nouvelle échelle de couleurs et les nouvelles bornes et en appliquant le rééchelonnage. Le rééchelonnage permet de faciliter l'identification des nœuds en exploitant l'intégralité de l'échelle de couleur. On repère notamment le nœud (7,5) pour lequel l'identification comme nœud sensible était plus incertaine avec la visualisation précédente.

4.3.2 Critère d'identification des nœuds sensibles

L'identification des nœuds sensibles est une étape importante pour la cartographie d'un modèle. Distinguer les nœuds sensibles des nœuds qui ne le sont pas doit reposer sur un critère commun à toutes les perturbations. De plus, un critère automatisable voire quantitatif permet une détection de ces nœuds plus facile et rapide. On propose un critère de sensibilité qui prend en compte les différentes données visualisées précédemment. On considère un nœud comme sensible si le score de sensibilité moyenne est supérieur à 0.1 en valeur absolue et le score de sensibilité moyenne normalisé est supérieur à 0.5 en valeur absolue. Ainsi, seuls les nœuds impactés significativement sont considérés dans le cas où ils "se détachent" du reste des nœuds. En complément du double critère sur la sensibilité moyenne, on ajoute un second double critère sur l'écart-type de la sensibilité du nœud. Un nœud peut être sensible si son écart-type est supérieur à 0.1 et que son écart-type normalisé est supérieur à 0.5. Ainsi, les nœuds ayant une moyenne proche de zéro à cause d'un effet d'équilibrage entre les documents sont tout de même considérés comme un nœud sensible à la perturbation appliquée aux documents. Pour récapituler, un nœud est considéré comme sensible si il vérifie la condition suivante :

$$((|mean| > 0.1) \wedge (|resc_mean| > 0.5)) \vee ((std > 0.1) \wedge (resc_std > 0.5))$$



(a) Nombre de perturbations de niveau 2 impactant les nœuds du modèle (b) Nombre de perturbations appliquées à *aérien* impactant les nœuds du modèle

FIGURE 4.8 – Matrices de nœuds sensibles

Il faut tout de même noter que le choix des bornes 0.5 et 0.1 sont intéressantes pour le dataset utilisé au cours du stage mais pourraient être à ajuster lors de l'utilisation d'autres datasets.

4.3.3 Matrice des nœuds sensibles

Pour considérer qu'un nœud traite une notion de langage ou de raisonnement (identifie le vocabulaire de spécialité par exemple), il doit être sensible à plusieurs des perturbations visant cette compétence. On peut aussi s'intéresser à la persistance de cette sensibilité à travers différents datasets similaires pour une même perturbation ou à travers différentes perturbations similaires pour un même dataset. Pour faciliter cette analyse de la persistance de la sensibilité d'un nœud, on propose de visualiser la matrice des nœuds sensibles. Pour chaque nœud, repéré par ses coordonnées, elle indique le nombre de perturbations pour lesquelles le nœud est considéré comme sensible au regard du critère présenté dans la sous-section précédente.

La figure 4.8 illustre deux usages des matrices de nœuds sensibles. La première compte le nombre de fois qu'un nœud est sensible aux perturbations décrites par la troisième colonne de la table 4.3. On constate qu'aucune tendance ne se dégage : les nœuds les plus sensibles ne le sont que pour la moitié des perturbations. La seconde matrice s'intéresse à la sensibilité des nœuds pour une ensemble de perturbations portant sur le même mot. On s'intéresse aux quatre perturbations de quatre niveau différents appliquées au mot *aérien*. Le nœud (11,6) est sensible aux quatre perturbations et les nœuds (11,3), (11,7) et (10,9) sont sensibles à trois des quatre perturbations étudiées.

Chapitre 5

Résultats expérimentaux

5.1 Résultats et Analyse

5.1.1 Application à la cartographie du vocabulaire de la défense français dans mE5

Les résultats obtenus à l'aide des outils et éléments méthodologiques présenté dans le chapitre précédent ne sont pour l'instant pas très concluants.

La majorité des perturbations permettent de mettre en évidence quelques nœuds sensibles. La majorité d'entre eux se situe dans les dernières couches du modèle, et une grande partie de ces nœuds appartient à la dernière couche du modèle. Il n'est pas exclu que cette sensibilité soit le reflet de l'étalement des documents dans l'espace de représentation appris par le modèle mE5. Le cas échéant, la qualité de l'amélioration qui pourra être déduite de ces informations sera limitée. Il en va de même pour les explications qui pourraient en être déduites.

Au delà de la localisation des nœuds identifiés par les perturbations, il ne semble pas se dégager de tendance permettant d'aboutir à une cartographie de la gestion du vocabulaire de spécialité de la défense. En effet, certains nœuds sont sensibles à plusieurs perturbations portant sur un même mot comme dans la figure 4.8 pour les perturbations portant sur le mot ciblé *aérien*. Cependant, il ne se dégage pas de tendance plus globale regroupant tout ou partie du vocabulaire de spécialité de la défense pour l'instant.

Toutefois, ces faiblesses sont à modérer. L'échelle à laquelle les expérimentations ont été effectuées étant très faible (peu de mots, petit dataset unique, un seul domaine,...) il n'est pas exclu que des résultats supplémentaire apparaissent en passant à plus grande échelle.

5.1.2 Préconisations d'utilisation

Afin de profiter pleinement des outils et de la méthode proposée, il est important de rester vigilant sur certains points.

Pour que les perturbations permettent de localiser le vocabulaire spécifique visé ou plus généralement d'identifier une compétence linguistique ou cognitive, les perturbations utilisées doivent couvrir le plus possible l'ensemble de documents ou extraits utilisés. Pour cela, il peut être pertinent d'utiliser plusieurs petits datasets dont le contenu est orienté afin que certains mots de vocabulaire apparaissent plus facilement.

Pour le cas de la localisation du vocabulaire de spécialité, il faut rester vigilant sur le choix des mots à remplacer. Une partie des mots porteurs de sens ne font pas partie du vocabulaire spécifique. Appliquer une perturbation sur ces mots aura un impact sur la distribution des scores de pertinence des documents perturbés cependant elle ne permettra pas de localiser le vocabulaire de spécialité en l'appliquant à l'activation patching.

Une vigilance particulière doit aussi être portée sur le bruit et les biais. Dans le cas des perturbations ayant un impact modéré sur la représentation des documents au sein du modèle, les nœuds peuvent ne montrer qu'une faible sensibilité. Le bruit induit par l'activation patching peut alors prendre le dessus et créer des faux positifs. De la même façon, le choix d'utiliser plusieurs petits datasets augmente le risque d'introduire un biais lors de leur construction. Une analyse systématique et rigoureuse des données fournies par l'activation patching et les étapes préalables permet la plupart du temps de les identifier. Il en va de même pour la construction des perturbations. Elles reposent sur la diversité du vocabulaire et la maîtrise de la langue de spécialité du concepteur.

5.2 Perspectives d'évolution de la méthode proposée

Au cours de la conception de la méthode proposée, des faiblesses ont été identifiées et des pistes d'amélioration ont été évoquées. Lors de la formulation des niveaux de perturbation, la notion d'"absence de lien" entre le mot cible et le mot remplaçant était fondée sur une intuition. [Greimas and Courtès, 1979] propose une définition de l'"incompatibilité" entre deux mots qui pourrai être exploitée afin de garantir le niveau de la perturbation à sa construction. Un outil permettant de faciliter le choix des mots remplaçants reste à construire.

Aucune méthode n'est proposée pour l'instant pour de mettre en perspective les matrices de sensibilité moyenne d'une même perturbation appliquée sur deux datasets différents. Cela reste une piste pour étoffer la méthodologie proposée.

Au delà de ces faiblesses, [Parry et al., 2025] propose deux autres types de perturbations que nous n'avons pas exploité dans la méthodologie. Une diversification des types perturbations et l'adaptation des outils d'interprétation des résultats est une piste d'exploration d'autant plus que les perturbations de type **prepend** et **append** donnent des résultats intéressants dans ses travaux.

De façon plus générale, il serai pertinent d'étendre la méthode actuelle à l'ensemble de la langue de spécialité. Les travaux actuels ne portent que sur le vocabulaire. La conception de perturbations permettant d'identifier les tournures spécifiques à un domaine de spécialité est une perspective intéressante.

Chapitre 6

Conclusions et Perspectives

6.1 Conclusion

Au cours de ce stage, j’ai exploré la littérature scientifique portant sur le RAG et son explicabilité. J’ai découvert les différentes perspectives d’évolution de ces systèmes et les perspectives d’explicabilité pour les tâches l’extraction d’information à l’aide des méthodes d’interprétabilité mécaniste. J’ai repris le système RAG proposé par [Tran et al., 2024] et conçu un dataset type à partir des données fournies par le challenge RAG@EvalLLM. Enfin, j’ai proposé une méthodologie d’usage de l’activation patching avec l’outil MechIR [Parry et al., 2025]. Cette dernière inclut la conception des perturbations ainsi que la visualisation et l’interprétation des résultats fournis par MechIR.

6.2 Perspectives générales

Au cours du stage, j’ai finalement choisi de poursuivre un sujet de thèse différent de celui visé par ce stage. Les travaux réalisés constituent néanmoins une première exploration méthodologique autour de l’interprétabilité mécaniste appliquée aux modèles d’extraction d’information pour le RAG.

Dans l’éventualité d’une reprise de ces travaux, voici quelques perspectives pour tendre vers le résultat initialement prévu. Dans un premier temps, L’identification des nœuds proposée ne constitue pas encore une cartographie d’un modèle d’extraction. C’est la première étape. Une fois la cartographie obtenue, l’extraction d’explication du fonctionnement du modèle lors de l’usage doit être définie. Il est nécessaire de garder en tête que l’utilisateur visé n’est pas informaticien. Par conséquent, un travail doit être effectué pour rendre les explications accessible au public visé.

Par ailleurs, d’autres perspectives d’usage d’une telle cartographie existent. Cette dernière peut servir d’orientation pour améliorer ou spécialiser un modèle. De premiers résultats existent déjà sur l’utilisation de l’activation patching appliqué au pruning.

Bibliographie

- [Artifex Software, 2026] Artifex Software (2026). PyMuPDF4LLM : PDF Markdown Extraction for LLM workflows.
- [Bell et al., 2022] Bell, A., Solano-Kamaiko, I., Nov, O., and Stoyanovich, J. (2022). It’s Just Not That Simple : An Empirical Study of the Accuracy-Explainability Trade-off in Machine Learning for Public Policy. In *2022 ACM Conference on Fairness Accountability and Transparency*, pages 248–266, Seoul Republic of Korea. ACM.
- [Cabr , 1998] Cabr , M. T. (1998). *La terminologie : th orie, m thode et application / Maria Teresa Cabr *. U. Linguistique. Armand Colin. Paris.
- [Chen et al., 2025] Chen, B., Khare, A., Kumar, G., Akula, A., and Narayana, P. (2025). Seeing Beyond : Enhancing Visual Question Answering with Multi-Modal Retrieval. In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., Schockaert, S., Darwish, K., and Agarwal, A., editors, *Proceedings of the 31st International Conference on Computational Linguistics : Industry Track*, pages 410–421, Abu Dhabi, UAE. Association for Computational Linguistics.
- [Chen et al., 2024] Chen, C., Merullo, J., and Eickhoff, C. (2024). Axiomatic Causal Interventions for Reverse Engineering Relevance Computation in Neural Retrieval Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1401–1410, Washington DC USA. ACM.
- [Cohere, 2024] Cohere (2024). Rerank Multilingual v3.0.
- [Fan et al., 2024] Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., and Li, Q. (2024). A Survey on RAG Meeting LLMs : Towards Retrieval-Augmented Large Language Models. arXiv :2405.06211.
- [Garouani et al., 2024] Garouani, M., Mothe, J., Barhrhouj, A., and Aligon, J. (2024). Investigating the Duality of Interpretability and Explainability in Machine Learning. In *2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 861–867. arXiv :2503.21356 [cs].
- [Geiger et al., 2021] Geiger, A., Lu, H., Icard, T., and Potts, C. (2021). Causal Abstractions of Neural Networks. arXiv :2106.02997 [cs.AI].
- [Greimas and Court s, 1979] Greimas, A. J. and Court s, J. (1979). *S miotique. Dictionnaire raisonn  de la th orie du langage*. Hachette, Paris.
- [Hofst tter et al., 2021] Hofst tter, S., Lin, S.-C., Yang, J.-H., Lin, J., and Hanbury, A. (2021). Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. arXiv :2104.06967 [cs.IR].
- [Hou et al., 2025] Hou, J., Cosma, G., and Finke, A. (2025). Advancing continual lifelong learning in neural information retrieval : Definition, dataset, framework, and empirical evaluation. *Information Sciences*, 687 :121368.
- [Huang et al., 2025] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and Liu, T. (2025). A Survey on Hallucination in Large Language Models : Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, 43(2) :1–55.
- [Lewis et al., 2020] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., K ttler, H., Lewis, M., Yih, W.-t., Rockt schel, T., Riedel, S., and Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv : Computation and Language*.
- [Liu et al., 2025] Liu, P., Liu, X., Yao, R., Liu, J., Meng, S., Wang, D., and Ma, J. (2025). HM-RAG : Hierarchical Multi-Agent Multimodal Retrieval Augmented Generation. *arXiv.org*.
- [L , 2024] L , X. H. (2024). BM25S : Orders of magnitude faster lexical search via eager sparse scoring. arXiv :2407.03618.
- [Ma et al., 2023] Ma, X., Gong, Y., He, P., Zhao, H., and Duan, N. (2023). Query Rewriting for Retrieval-Augmented Large Language Models. arXiv :2305.14283 [cs.CL].

- [Nanda and Bloom, 2022] Nanda, N. and Bloom, J. (2022). TransformerLens.
- [Ni et al., 2025] Ni, B., Liu, Z., Wang, L., Lei, Y., Zhao, Y., Cheng, X., Zeng, Q., Dong, L., Xia, Y., Kenthapadi, K., Rossi, R., Dernoncourt, F., Tanjim, M. M., Ahmed, N., Liu, X., Fan, W., Blasch, E., Wang, Y., Jiang, M., and Derr, T. (2025). Towards Trustworthy Retrieval Augmented Generation for Large Language Models : A Survey. arXiv :2502.06872.
- [Parry et al., 2025] Parry, A., Chen, C., Eickhoff, C., and MacAvaney, S. (2025). MechIR : A Mechanistic Interpretability Framework for Information Retrieval. In *Advances in Information Retrieval - 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6-10, 2025, Proceedings, Part V*, volume 15576 of *Lecture Notes in Computer Science*, pages 89–95. Springer.
- [Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch : an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA.
- [Somvanshi et al., 2026] Somvanshi, S., Islam, M. M., Rafe, A., Tusti, A. G., Chakraborty, A., Baitullah, A., Chowdhury, T. I., Alnawmasi, N., Dutta, A., and Das, S. (2026). Bridging the Black Box : A Survey on Mechanistic Interpretability in AI. *ACM Computing Surveys*, 58(8) :1–35.
- [Sparck Jones, 1972] Sparck Jones, K. (1972). A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. *Journal of Documentation*, 28(1) :11–21.
- [Touvron et al., 2023] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). LLaMA : Open and Efficient Foundation Language Models. arXiv :2302.13971.
- [Tran et al., 2024] Tran, T. T., González-Gallardo, C.-E., and Doucet, A. (2024). Retrieval Augmented Generation for Historical Newspapers. In *Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries*, pages 1–5, Hong Kong China. ACM.
- [Wang et al., 2022] Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., and Wei, F. (2022). Text Embeddings by Weakly-Supervised Contrastive Pre-training.
- [Wang et al., 2024] Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. (2024). Multilingual E5 Text Embeddings : A Technical Report.
- [Wang et al., 2023] Wang, L., Yang, N., and Wei, F. (2023). Query2doc : Query Expansion with Large Language Models. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.
- [Wolf et al., 2020] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P. v., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). HuggingFace’s Transformers : State-of-the-art Natural Language Processing. arXiv :1910.03771 [cs.CL].
- [Xu et al., 2024] Xu, S., Pang, L., Shen, H., Cheng, X., and Chua, T.-S. (2024). Search-in-the-Chain : Interactively Enhancing Large Language Models with Search for Knowledge-intensive Tasks. arXiv :2304.14732.
- [Zhu, 2025] Zhu, Y. (2025). From Static to Dynamic : A Streaming RAG Approach to Real-time Knowledge Base. arXiv :2508.05662.

Annexe A

Lieu du stage, outils et ressources utilisés

A.1 Laboratoire d’Informatique Fondamentale d’Orléans

Le stage présenté dans ce rapport s’est déroulé au sein du Laboratoire d’Informatique Fondamentale d’Orléans (LIFO), unité de recherche en informatique à l’Université d’Orléans et à l’INSA Centre Val de Loire. Le laboratoire rassemble des enseignants-chercheurs, chercheurs, doctorants et personnels d’appui autour de travaux couvrant un large spectre de l’informatique fondamentale.

Les activités de recherche du LIFO sont organisées en cinq équipes, chacune développant une expertise dans un domaine particulier de l’informatique fondamentale ou appliquée. Cette structuration permet de couvrir un large éventail de thématiques de recherche, allant des fondements théoriques de l’informatique jusqu’au développement de méthodes et d’outils répondant à des problématiques concrètes. Les équipes du laboratoire sont les suivantes :

- L’équipe **GAMoC** (Graphes, Algorithmes et Modèles de Calcul) s’intéresse à l’algorithmique dans toutes sortes de structures discrètes (graphes, automates, pavages,...). Ces travaux se divisent en deux thèmes : l’étude des problèmes NP-difficiles et l’étude des nouveaux modèles de calcul.
- L’équipe **LMV** (Langages, Modélisation et Vérification) porte sur la fiabilité et la sécurité des logiciels. Ses travaux portent notamment sur la conception d’outils garantissant la présence de propriété critiques pour les logiciels.
- L’équipe **PaMDA** s’inscrit dans le domaine des Sciences des Données. Ses travaux se divisent en deux axes. L’axe parallélisme explore les modèles et patterns de programmation spécifiques à certaines architectures. L’axe Base de Données porte sur la structuration et l’optimisation de l’interrogation des données.
- L’équipe **SDS** (Sécurité des Données et des Systèmes) s’inscrit dans le domaine de la sécurité informatique au sens large. Son axe sécurité des données porte sur la confidentialité et l’anonymat et l’axe sécurité des systèmes porte sur la conception et la preuve de protocoles de sécurisation.
- L’équipe **CA** (Contraintes et Apprentissage) est structurée autour de trois axes. La modélisation et la programmation par contraintes, l’apprentissage automatique sous toutes ses formes (apprentissage supervisé, profond, hybride, par renforcement,...) et le traitement automatique des langues. Ces trois axes intègrent un thème transverse : l’intégration de connaissances et l’interprétabilité des modèles.

Dans le cadre de ce stage, j’ai intégré l’équipe Contraintes et Apprentissage sous la supervision de Mme Halftermeyer. Les travaux réalisés s’inscrivent dans l’axe de recherche de l’équipe consacré au traitement automatique des langues naturelles (TALN) et intègrent l’axe transversal à travers mes propositions méthodologiques d’explicabilité.

A.2 Outils utilisés durant le stage

Les travaux réalisés au cours de ce stage ont mobilisé différents outils logiciels et ressources de calcul dédiés au développement, à l’expérimentation et à l’évaluation de méthodes d’explicabilité pour les modèles de traitement automatique des langues.

Les développements réalisés durant ce stage ont été principalement effectués en **Python**, sous trois formats complémentaires : des **notebooks Jupyter** pour le prototypage et l’exploration des données, des **scripts** pour l’exécution des expérimentations, et une **interface en ligne de commande (CLI)** afin de structurer les outils développés et de faciliter leur réutilisation.

Les principales bibliothèques utilisées sont **PyTorch** [Paszke et al., 2019] pour le développement et l’exécution des modèles, **Hugging Face** [Wolf et al., 2020] pour l’accès aux modèles de traitement automatique des langues,

TransformerLens [Nanda and Bloom, 2022] pour l’analyse interne des architectures Transformer, ainsi que des bibliothèques plus spécialisées telles que **PyMuPDF4LLM** [Artifex Software, 2026] pour l’extraction de contenu à partir de documents PDF.

Le suivi du développement a été assuré à l’aide de **Git** pour faciliter la gestion des versions et le suivi du bon déroulé du stage.

Les expérimentations les plus coûteuses en calcul, telles que la génération des représentations des données ou l’évaluation des différentes architectures étudiées, ont été exécutées sur les grappes de calcul Centre de Calcul Scientifique en région Centre-Val de Loire (**CaSciModOT**) et **Mirev**, mises à disposition des chercheurs du LIFO.

Enfin, plusieurs outils d’intelligence artificielle générative ont été utilisés de manière ponctuelle pour des tâches d’assistance : **ChatGPT** pour la reformulation et le développement de fonctions Python simples, **DeepL** pour la traduction d’extraits de documents scientifiques, et **Litmaps** ainsi que **Perplexity** pour la veille bibliographique et l’exploration de la littérature. Ces outils n’ont pas été utilisés pour la conception des contributions scientifiques présentées dans ce mémoire.

A.3 Financement

Ce travail a bénéficié d’une aide de l’État gérée par l’Agence Nationale de la Recherche au titre de France 2030 portant la référence "Minerve" ANR-22-EXES-0010.

Annexe B

Dépôt Git des outils développés durant le stage

Les outils et scripts développés dans le cadre de ce stage sont disponibles sur le dépôt GitLab suivant :

GitLab : <https://gogit.univ-orleans.fr/lifo/ah/ragrights>

Ce dépôt contient notamment :

- des traces d’exploration bibliographique ;
- les notebooks d’exploration ;
- les scripts d’expérimentation ;
- l’interface en ligne de commande développée au cours du stage ;
- la documentation d’utilisation ;
- les différents supports de communication et rapports produits dans ce cadre.