Université de Blida 1

Faculté des Sciences

Département d'Informatique

MEMOIRE DE MASTER

En Informatique

Option : Ingénierie des Systèmes Intelligents

THÈME :

Intégration d'un Graphe de Connaissances Médicales dans un Système d'Assistance Intelligent Basé sur la Génération Augmentée par Récupération

Réalisé par :

M[lles] MEFTI *Manel et* ALLAMI *Khadidja*

Soutenu devant le jury composé de :

Dr. M. MEZZI	**Promotrice**	Université de Blida 1
Dr. M. FAREH	**Présidente**	Université de Blida 1
Dr. N. LAHIANI	**Examinatrice**	Université de Blida 1

2024 / 2025

Remerciements

Nous tenons tout d'abord à adresser notre plus profonde gratitude à **ALLAH**, Tout-Puissant, pour la force, la patience et la persévérance qu'Il nous a insufflées tout au long de ce parcours. C'est par Sa miséricorde que nous avons pu mener à bien ce travail.

Nos remerciements les plus sincères vont à **Madame Mezzi Melyara**, notre promotrice, pour sa disponibilité, ses conseils avisés et son accompagnement empreint de bienveillance. Sa rigueur scientifique et la confiance qu'elle nous a accordée ont été une source de motivation constante.

Nous exprimons également notre reconnaissance à **l'ensemble des enseignants de l'Université Saad Dahlab de Blida**, pour la richesse de leur enseignement et leur dévouement à transmettre leur savoir avec passion.

Notre gratitude s'adresse aussi à **nos familles**, dont le soutien inconditionnel, les encouragements et les prières ont constitué un appui essentiel tout au long de cette aventure.

Enfin, nous remercions chaleureusement **les membres du jury**, pour l'honneur qu'ils nous font en évaluant ce travail et pour le temps précieux qu'ils y consacrent.

M^{lles} Khadidja & Manel

Résumé

Dans le cadre d'un contexte médical dans lequel la véracité et la pertinence de l'information sont incontournables, ce travail propose le développement d'un système d'assistance médicale intelligente reposant sur une architecture de génération augmentée par récupération par un graphe de connaissances biomédicales .

Un système accessible et intuitif a été mis en place pour que l'utilisateur, qu'il soit patient ou personnel soignant, soit en mesure d'interagir facilement avec le système. La globalité de la solution a été testée à travers des scénarios expérimentaux permettant de faire preuve de l'efficacité de cette dernière.

L'objectif principal de ce mémoire est d'améliorer la précision, la contextualisation et la cohérence des réponses générées dans un cadre médical, en combinant les capacités de récupération d'information avec la génération de langage naturel adaptée au domaine biomédical.

L'approche adoptée repose sur l'utilisation de modèles avancés pour la production d'embeddings sémantiques et la génération de réponses médicales. BioBERT se distingue par sa capacité à offrir des représentations sémantiques précises des termes médicaux, tandis que BioGPT démontre une efficacité notable dans la génération de réponses pertinentes et compréhensibles, adaptées au domaine de la santé.

Mots clés :
Graphes de Connaissances, Génération Augmentée par Récupération, Appariement Sémantique, Extraction d'Information et Génération de Texte.

ملخص

في سياق طبي حيث تكون صحة المعلومات وأهميتها أمرًا ضروريًا، يقترح هذا العمل تطوير نظام مساعدة طبية ذكي يعتمد على بنية جيل معززة بالاسترداد من خلال رسم بياني للمعرفة الطبية الحيوية.

تم إنشاء نظام سهل الوصول إليه وبديهي حتى يتمكن المستخدم، سواء كان مريضًا أو مقدم رعاية، من التفاعل بسهولة مع النظام. تم اختبار الحل الشامل من خلال سيناريوهات تجريبية لإثبات فعاليته.

الهدف الرئيسي من هذه الأطروحة هو تحسين دقة وسياق واتساق الاستجابات الناتجة في البيئة الطبية من خلال الجمع بين قدرات استرجاع المعلومات وتوليد اللغة الطبيعية الملائمة للمجال الطبي الحيوي.

يعتمد النهج المتبع على استخدام نماذج متقدمة لإنتاج التضمينات الدلالية وتوليد الاستجابات الطبية. يتميز BioBERT بقدرته على تقديم تمثيلات دلالية دقيقة للمصطلحات الطبية، في حين يُظهر BioGPT كفاءة ملحوظة في توليد استجابات ذات صلة ومفهومة، تتكيف مع مجال الصحة.

الكلمات المفتاحية: الرسوم البيانية المعرفية، التوليد المعزز عن طريق الاسترجاع، المطابقة الدلالية، استخراج المعلومات وتوليد النص.

Abstract

In a medical context where the accuracy and relevance of information are essential, this work proposes the development of an intelligent medical assistance system based on an augmented generation architecture through retrieval using a biomedical knowledge graph.

An accessible and intuitive system was implemented so that users, whether patients or healthcare staff, could easily interact with the system. The entire solution was tested through experimental scenarios to demonstrate its effectiveness.

The main objective of this thesis is to improve the accuracy, contextualization, and consistency of responses generated in a medical setting by combining information retrieval capabilities with natural language generation adapted to the biomedical domain.

The adopted approach relies on advanced models for the production of semantic embeddings and the generation of medical responses. BioBERT stands out for its ability to offer precise semantic representations of medical terms, while BioGPT demonstrates a notable efficiency in generating relevant and comprehensible responses, adapted to the field of health.

Keywords :

Knowledge Graphs, Retrieval Augmented Generation, Semantic Matching, Information Extraction and Text Generation.

Table des Matières

Introduction Générale **1**
1 Contexte de Travail 1
2 Problématique 1
3 Objectif du Travail 2
4 Organisation du Mémoire 2

Chapitre I : État de l'Art **5**
1.1 Introduction 5
1.2 Génération Augmentée par Récupération 5
1.2.1 Principe 6
1.2.2 Bénéfices de l'Adoption d'un Système RAG 6
1.2.3 Processus d'Implémentation des RAG 6
1.2.4 Métriques de Similarité Vectorielle 9
1.2.5 Base de Données Vectorielle 10
1.2.6 Index Structurel 10
1.2.7 Travaux Récents des RAG dans le Domaine Medical 10
1.3 Graphe de Connaissance 12
1.3.1 Notion de Base de KG 12
1.3.2 Types de Graphes de Connaissances 13
1.3.3 Travaux Récents sur les KGs dans le Domaine Médical 15
1.4 Intégration des RAG et des Graphes de Connaissances 16
1.4.1 Travaux Récents de l'Intégration dans le Domaine Medical 16
1.5 Conclusion 18

Chapitre II : Conception et Modélisation de la Solution **19**
2.1 Introduction 19
2.2 Processus Global du Système de Génération Augmentée par Récupération . . . 19
2.3 Collection de Données 20
2.4 Prétraitement des Données 22

2.5 Génération des Embeddings Sémantiques 23
2.5.1 Modèle BioBERT 24
2.5.2 Modèle SciSpaCy 27
2.5.3 Modèle Word2vec 28
2.5.4 Comparaison des Modèles 29
2.6 Appariement entre Graphe de Connaissance et Requête 30
2.7 Intégration du Modèle de Génération dans le Système RAG 31
2.7.1 Modèle BioGPT 32
2.7.2 Modèle GPT-2 34
2.7.3 Modèle Flan-T5-Base 35
2.7.4 Comparaison des Modèles 37
2.8 **Conclusion** 38

Chapitre III : Tests et Validation de la Solution **41**
3.1 Introduction 41
3.2 Environnement de Developement 41
3.2.1 Environnement Matériel 41
3.2.2 Environnent Logiciel 41
3.3 Scénarios d'Expérimentation et d'Evaluation du Système 44
3.3.1 Evaluation du Modèles de Génération des Embeddings 44
3.3.2 Évaluation des Modèles de Génération 47
3.3.3 Interface de Requêtage 50
3.4 Conclusion 52

Conclusion Générale **52**
1 Conclusion 53
2 Perspectives 53

Bibliographie **i**

Liste des Figures

1.1 Processus de Traitement dans un Système RAG [1]. 6
1.2 Etapes de Collection de Données [2]. 7
1.3 Processus de Fragmentation de Données. 7
1.4 Vectorisation des Mots et leur Représentation Spatiale. 8
1.5 Principe de Calcul de Similarité Vectorielle [3]. 9
1.6 Métriques de Similarité Vectorielle. 9
1.7 Exemple d'un Graphe de Connaissances Simple. 13
1.8 Types de Graphes de Connaissances. 14
2.9 Processus Global du Système. 20
2.10 Visualisation Globale du Graphe de Connaissances. 21
2.11 Exemple Détaillé d'un Triplet. 22
2.12 Processus de Génération d'Embedding. 24
2.13 Étapes du Génération des Embeddings avec le Modèle BioBERT. 25
2.14 Architectures du Modèle Word2vec [4]. 28
2.15 Etapes d'Appariement entre la Requête et KG. 30
2.16 Processus du Modèle de Génération. 32
2.17 Fonctionnement du Modèle BioGPT. 33
2.18 Fonctionnement du Modèle GPT 2. 35
2.19 Fonctionnement du Modèle T5. 36
3.20 Processus de Calcul de BERTScore [5]. 48
3.21 Page d'Accueil pour la Recherche d'Information Biomédicale. 50
3.22 Page du Résultat de la RI. 51
3.23 Page de Détails des Résultats de la RI. 51
3.24 Page de la Visualisation Interactive des Resultats. 52

Liste des Tableaux

1.1 Quelques Travaux Récents des RAG dans le Domaine Médical. 11
1.2 Quelques Travaux Récents sur les KGs dans le Domaine Médical. 15
1.3 Quelques Travaux Récents sur l'Intégration des RAG et KGs dans le Domaine Medical. 17
2.4 Statistiques du Dataset. 21
2.5 Exemples de Prétraitement. 23
2.6 Étapes de Génération des Embeddings avec le Modèle SciSpaCy. 27
2.7 Comparaison des Modèles de Génération des Embeddings. 29
2.8 Comparaison des Modèles de Génération. 37
2.9 Comparaison des Réponses des Modèles. 38
3.10 Environnement Matériel. 41
3.11 Technologies Utilisés dans le Projet. 42
3.12 Outils Utilisés dans le Projet. 43
3.13 Évaluation des Modèles de Génération des Embeddings. 46
3.14 Evaluation des Modèles de Génération. 49

Liste des Abréviations

BERT Bidirectional Encoder Representations from Transformers. 24, 25

BioBERT Bidirectional Encoder Representations from Transformers for Biomedical Text Mining. 24–26

BioGPT Generative Pre-trained Transformer for Biomedical Text Generation and Mining. 32, 33, 37, 38, 44, 49, 50

GPT-2 Generative Pre-trained Transformer 2. 34, 35, 37, 38, 49, 50

IA Artificial Intelligence. 1, 5–7, 10, 18, 19, 42, 53

KG Knowledge Graphs. 1, 2, 5, 10, 12, 15–19, 21–25, 30–32, 35, 38, 44, 47, 53

LLMs Large Language Models. 1, 5, 6, 10, 16, 53

NLM National Library of Medicine. 20

NLP Natural Language Processing. 1, 9, 22, 27, 45, 50

OWL Web Ontology Language. 14

RAG Retrieval-Augmented Generation. 2, 5, 6, 10, 16–19, 24, 30–34, 36, 38, 52, 53

RDF Resource Description Framework. 14

RF2 Release Format 2. 20

SciSpaCy Scientific spaCy. 27

T5 Text-to-Text Transfer Transformer. 35–38, 49, 50

UMLS Unified Medical Language System. 18

Introduction Générale

1 Contexte de Travail

Ces dernières années, l'Artificial Intelligence (IA) s'est imposée dans de nombreux domaines, devenant un moteur essentiel, et le secteur médical n'échappe pas à cette tendance. Grâce à ses capacités remarquables, l'IA transforme profondément les pratiques et les réflexions liées aux soins en aidant les médecins à poser des diagnostics plus précis, à mieux suivre l'évolution des patients, et à personnaliser les traitements.

Aujourd'hui, les professionnels de santé sont confrontés à une masse d'informations impressionnante — dossiers patients, publications scientifiques, résultats d'analyses, nouveaux protocoles — qui constituent une richesse considérable mais aussi un défi majeur, rendant presque impossible pour un individu seul de tout suivre et exploiter efficacement.

C'est dans ce contexte que l'IA révèle toute sa valeur : non pas pour remplacer l'humain, mais pour le soutenir dans la compréhension et la prise de décisions éclairées. Cependant, malgré les avancées significatives des Large Language Models (LLMs) et des technologies de Natural Language Processing (NLP) — telles que GPT, BERT ou LLaMA — ces modèles rencontrent parfois des difficultés à fournir des réponses précises et fiables, en raison d'un manque d'ancrage solide dans des connaissances validées, structurées et actualisées.

C'est ici que les graphes de connaissances (Knowledge Graphs (KG)) médicaux jouent un rôle clé, en établissant des liens entre les concepts fondamentaux du domaine médical — maladies, symptômes, médicaments, traitements — afin d'apporter un contexte dense, stable et intégrable aux systèmes d'IA, et en particulier aux LLMs.

Dans ce projet, nous proposons « Intégration d'un Graphe de Connaissances Médicales dans un Système d'Assistance Intelligent Basé sur la Génération Augmentée par Récupération », une démarche innovante qui associe la puissance des grands modèles de langage à la richesse des graphes médicaux pour concevoir un système intelligent capable de fournir des réponses précises, pertinentes et adaptées aux besoins des professionnels de santé.

2 Problématique

Chaque jour, les données médicales se multiplient, tandis que l'information à la disposition des professionnels de santé devient de plus en plus difficile à analyser et à exploiter. La difficulté d'un accès rapide à des connaissances fiables et pertinentes peut entraver la capacité à formuler des décisions cliniques précises et personnalisées.

Il devient par conséquent impératif de concevoir des systèmes capables d'extraire et d'accéder directement à des connaissances médicales validées et structurées, afin de fournir des réponses précises et fiables dans le domaine de la santé.

Ainsi, la problématique centrale est la suivante : comment construire un assistant médical intelligent capable d'exploiter efficacement les graphes de connaissances médicales au service de la récupération et de l'accès à une information validée, précise et contextuelle ? Comment mettre en œuvre des méthodes fiables permettant d'extraire la réponse la plus appropriée à partir d'un graphe complexe et structuré, de manière à garantir une information médicale rigoureuse et adaptée aux besoins des professionnels de santé ? Enfin, comment articuler ces différents modes opératoires au sein d'un système global alliant la richesse de la connaissance structurée à la performance de l'intelligence artificielle, pour améliorer la qualité des décisions cliniques ?

Pour répondre à cette problématique, nous nous appuyons sur le système Retrieval-Augmented Generation (RAG), qui combine une recherche intelligente d'informations dans un KG médical avec la capacité de générer des réponses claires, précises et adaptées à chaque contexte.

3 Objectif du Travail

Ce projet a pour ambition de concevoir et de développer un assistant médical intelligent capable de fournir aux professionnels de santé des réponses fiables, précises et contextualisées, en s'appuyant sur l'exploitation optimale de KG médicales. L'objectif central consiste à intégrer un système de type RAG, combinant une recherche intelligente et ciblée dans un corpus structuré — tel qu'un graphe de connaissances — avec les capacités de génération en langage naturel offertes par les grands modèles de langage.

L'approche vise à maximiser la pertinence et la qualité des informations extraites, en garantissant des réponses à la fois exactes et adaptées au contexte clinique spécifique de chaque demande. Cette méthode hybride permet ainsi de surmonter certaines limites des modèles de langage classiques, en les enrichissant par des connaissances structurées et validées.

Le projet propose ainsi la mise en œuvre d'une solution opérationnelle de soutien à la décision médicale, permettant aux praticiens d'accéder facilement à un système d'information fiable, capable de faire face à la complexité croissante et à la volumétrie massive des données médicales. Par ailleurs, cette réflexion s'inscrit dans une démarche de modélisation des mécanismes d'intégration des graphes de connaissances dans un pipeline RAG, en vue d'optimiser à la fois l'extraction d'informations pertinentes et la qualité des réponses générées, pour une assistance médicale plus performante et intelligemment pilotée.

4 Organisation du Mémoire

Afin d'atteindre les objectifs cités ci-dessus, notre mémoire s'articulera autour de trois chapitres :

— **Chapitre 1 :** Etat de l'art qui contiendra deux parties :

1. Génération Augmentée par Récupération, nous définissons le concept de RAG et ses principes. Nous présenterons ensuite le fonctionnement général du pipeline, avant d'explorer ses principales applications dans le domaine médical.

2. Graphes de connaissances, nous donnerons d'abord une introduction aux graphes de connaissances biomédicaux. Nous distinguerons ensuite les différents types de graphes existants, puis nous décrirons comment ces graphes sont intégrés dans les systèmes d'intelligence artificielle médicale.

— **Chapitre 2 :** Dans ce chapitre, l'approche proposée ainsi que l'architecture globale du système développé sont présentées. La collecte et le prétraitement des données sont d'abord décrits, suivis par l'utilisation de différents modèles pour la génération des embeddings sémantiques, accompagnée d'une comparaison de leurs performances respectives. L'intégration des requêtes avec le graphe de connaissances (KG) est ensuite expliquée, ainsi que les modèles sélectionnés pour la génération des réponses médicales, dont l'efficacité est évaluée de manière comparative.

— **Chapitre 3 :** ce chapitre décrit l'environnement matériel et logiciel utilisé, les outils, les bibliothèques, ainsi que le système conçu et son application. On y trouve aussi une évaluation détaillée des performances des modèles d'embeddings et de génération, avec une analyse des résultats obtenus. Enfin, le chapitre se termine par une présentation des interfaces développées pour la plateforme .

Chapitre I : État de l'Art

1.1 Introduction

L'essor de IA dans le domaine de la santé modifie la manière dont les professionnels de santé accèdent à l'information et l'utilisent. Les systèmes avancés, tels que ceux basés sur RAG, extraient des données pertinentes de grandes bases de données pour étayer les recommandations de diagnostic et de traitement. Cependant, ces systèmes présentent des limites importantes en termes de précision et de contextualisation, ce qui conduit parfois à de graves erreurs dans des contextes cliniques sensibles.

Dans ce contexte, les KG se révèlent être des outils puissants pour construire et valider des informations complexes [6]. En établissant des relations claires entre des entités telles que les maladies, les traitements et les symptômes, ces graphes améliorent la qualité et la fiabilité des réponses générées. Par conséquent, l'intégration des KG dans les systèmes RAG constitue une approche prometteuse pour surmonter les défis de précision et de contextualisation des systèmes d'IA médicale actuels.

Dans ce chapitre, nous abordons les RAG et les KG, en mettant en évidence leur fonctionnement, leurs avantages et les défis dans le domaine médical. Nous discutons de leur intégration comme solution pour améliorer la précision et la contextualisation des recommandations en IA, tout en surmontant les limitations actuelles dans les contextes cliniques.

1.2 Génération Augmentée par Récupération

Les LLMs génèrent du texte en se basant sur des données d'entraînement statiques, ce qui limite leur capacité à fournir des réponses précises face à des informations nouvelles ou spécifiques. Traditionnellement, l'amélioration de leurs performances nécessitait un réentraînement coûteux. Le RAG propose une alternative plus efficace en combinant la génération de texte avec une recherche d'informations externe en temps réel. Ainsi, au lieu de s'appuyer uniquement sur leur mémoire interne, les modèles peuvent récupérer des connaissances à partir de bases de données ou de documents, améliorant ainsi la pertinence et l'actualisation des réponses [7].

1.2.1 Principe

Le RAG [1] analyse l'input de l'utilisateur pour émettre une requête et scrute une base de connaissances déjà existante pour extraire les morceaux nécessaires. Les morceaux récupérés sont ajoutés au prompt initial, dotant ainsi l'IA d'un antécédent contextuel. Un prompt amplifié permet aux LLMs de proposer des réponses plus précises et nuancées. La flexibilité du RAG résout les défis statiques, le rendant dynamique pour les tâches nécessitant l'actualité des connaissances. La figure 1.1 suivante résume le processus de traitement dans un système RAG.

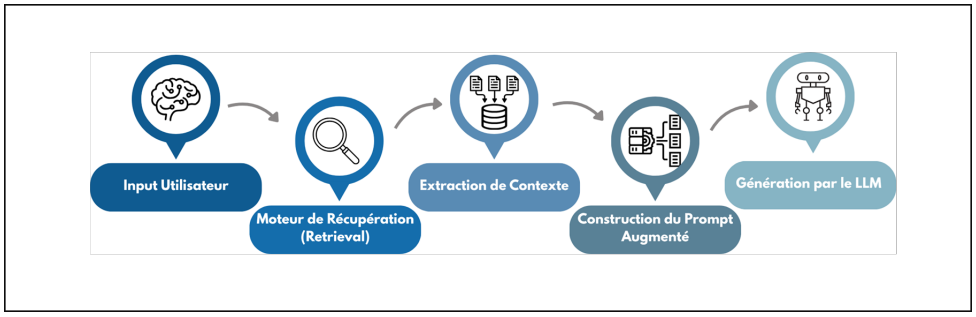


FIGURE 1.1 – Processus de Traitement dans un Système RAG [1].

1.2.2 Bénéfices de l'Adoption d'un Système RAG

La génération augmentée par récupération est une étape majeure pour LLMs, qui sont confrontés à plusieurs limites majeures [8] :

· **Des informations ciblées :** contrairement aux modèles de langage à grande échelle traditionnels, qui peuvent généralement extraire des données d'entraînement statiques, les RAG se connectent à des bases de données actuelles. Par conséquent, les réponses générées sont plus précises, plus orientées et mieux adaptées à l'exigence contextuelle.

· **Moins d'erreurs et d'hallucinations :** les modèles de langage RAG peuvent donner des réponses fausses ou fabriquées, ce qui est sévèrement limité par l'utilisation de RAG : les sources fiables et réglementées garantissent que les réponses sont toujours basées sur les faits vérifiés.

· **Qualité et adaptation améliorées :** les mécanismes de recherche et de récupération spécialement conçus d'un générateur de réponse augmentent le niveau d'acceptation de la réponse ; dans d'autres, différentes approches de RAG répondent plus précisément aux questions de l'utilisateur et donnent des réponses plus longues et plus détaillées.

1.2.3 Processus d'Implémentation des RAG

Le modèle RAG suit un processus en cinq étapes pour traiter les données [2].

· **Étape 1 : Collecte de Données**

Au premier stade du modèle RAG, une grande importance est accordée à la capacité de rassembler les bonnes informations provenant de différentes sources. Ces activités comprennent

la sélection de dépôts pertinents tels que des bases de données, des articles et d'autres sites Web qui contiennent des données pertinentes liées à la tâche ou aux sujets spécifiques. Des requêtes minutieusement élaborées sont construites afin qu'une certaine portée d'informations soit récupérée, garantissant que les données collectées sont cohérentes avec le contexte et les exigences du modèle d'IA. La figure 1.2 suivante résume les étapes de collection de données [2].

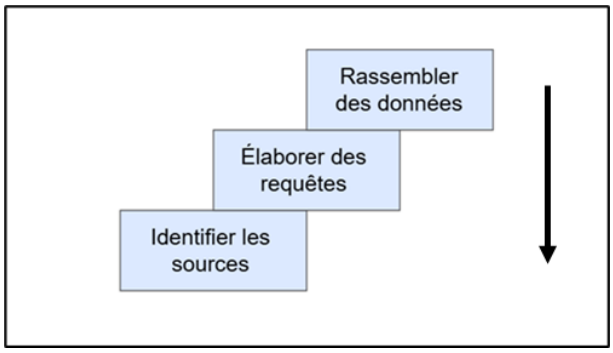


FIGURE 1.2 – Etapes de Collection de Données [2].

Étape 2 : Fragmentation des données (Chunking) Le « chunking », également connu sous le nom de division intégrative, consiste à décomposer de longs textes ou de grands documents en portions plus petites et plus gérables appelés « morceaux ». Chaque morceau est une partie du texte qui est assez cohérente en signification et peut être examinée de manière approfondie . la figure 1.3 suivante illustre ce mécanisme [2].

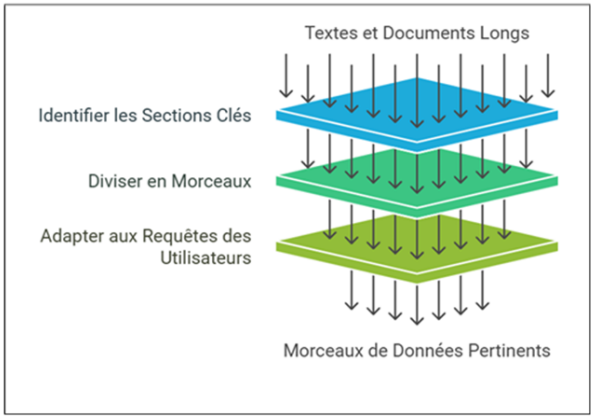


FIGURE 1.3 – Processus de Fragmentation de Données.

· **Étape 3 : Vectorisation des Documents – Embeddings**

Les embeddings représentent une méthode de vectorisation qui permet de transformer des mots ou expressions en vecteurs dans un espace à haute dimension. Ces représentations numériques capturent à la fois les relations sémantiques et syntaxiques entre les éléments linguistiques, facilitant ainsi leur traitement par des modèles d'apprentissage automatique.

· **Étape 4 : Indexation des Embeddings dans une Base de Données Vectorielle**

Une base de données vectorielle est un type spécialisé de base de données conçu pour stocker et interroger des informations représentées sous forme de vecteurs multidimensionnels. Chaque vecteur encode certaines caractéristiques ou attributs d'une donnée, qu'il s'agisse de texte, d'images, de fichiers audio ou vidéo. Le nombre de dimensions peut varier considérablement, allant de quelques dizaines à plusieurs milliers, selon la complexité et le niveau de granularité des informations traitées [9].

Ces vecteurs sont généralement générés à l'aide de techniques d'apprentissage automatique, d'embeddings lexicaux ou d'algorithmes d'extraction de caractéristiques. Contrairement aux bases de données traditionnelles, qui manipulent des données scalaires (chaînes de caractères, nombres, etc.) organisées en lignes et colonnes, les bases vectorielles reposent sur des structures de données optimisées pour la recherche de similarité dans des espaces vectoriels, telles que la recherche de plus proches voisins (k-NN). Cela implique des méthodes d'indexation et de requêtage fondamentalement différentes, adaptées à la nature continue et dense des représentations vectorielles.

La figure 1.4 suivante montre la Vectorisation des mots et leur représentation spatiale

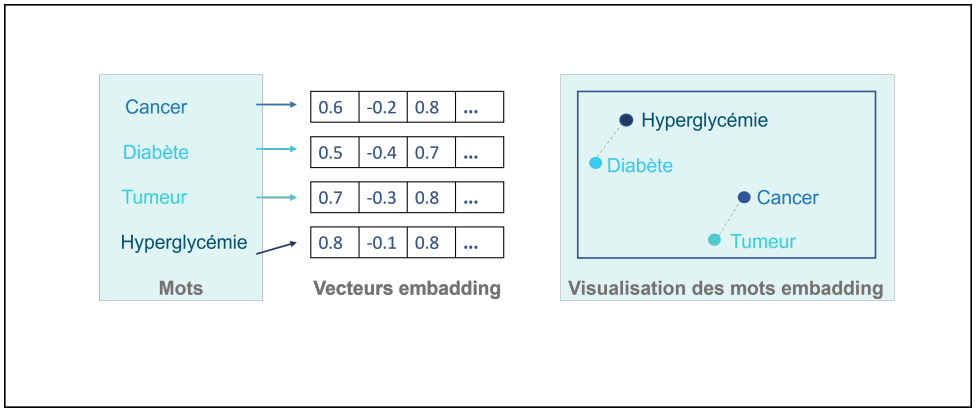


FIGURE 1.4 – Vectorisation des Mots et leur Représentation Spatiale.

· **Étape 5 : Recherche de Similarité entre la Requête Utilisateur et la Base de Données Vectorielle**

La recherche de similarité vectorielle consiste à comparer des vecteurs afin d'évaluer leur degré de proximité sémantique ou contextuelle. Ce processus repose sur l'utilisation de mesures de distance (telles que la distance cosinus, euclidienne, ou de Manhattan) permettant de quantifier la similarité entre la requête de l'utilisateur — convertie en vecteur — et les vecteurs indexés dans la base de données. Cette approche constitue le cœur des algorithmes de recherche de similarité vectorielle et permet d'identifier les éléments les plus pertinents dans un espace à haute dimension. La figure 1.5 suivante illustre ce mécanisme [3].

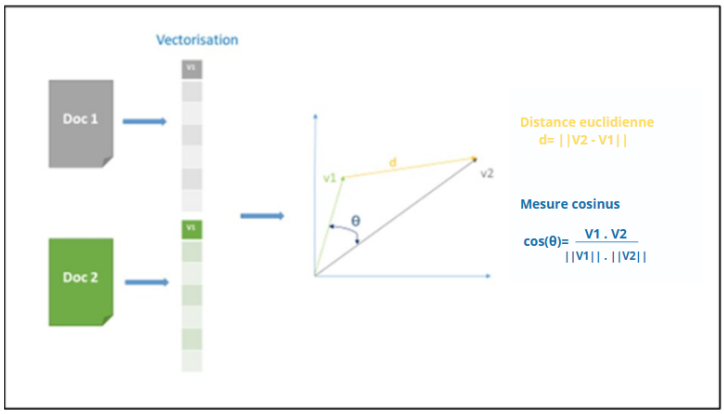


FIGURE 1.5 – Principe de Calcul de Similarité Vectorielle [3].

1.2.4 Métriques de Similarité Vectorielle

Dans le contexte du NLP, plusieurs mesures de similarité vectorielle ont été développées pour comparer des représentations numériques de mots ou de phrases. Ces représentations, appelées vecteurs, peuvent être exprimées sous forme de listes de nombres ou décrites en termes d'orientation et de magnitude. Pour mieux comprendre cela, on peut imaginer les vecteurs comme des segments de ligne pointant dans des directions spécifiques dans l'espace [10].

Parmi les principales mesures de similarité, on trouve :

· La métrique L2 ou euclidienne est la métrique "hypoténuse" de deux vecteurs. Elle mesure la magnitude de la distance entre les extrémités des lignes de vos vecteurs.

· La similarité cosinus est l'angle entre vos lignes là où elles se rencontrent.

· Le produit intérieur est la "projection" d'un vecteur sur l'autre. Intuitivement, il mesure à la fois la distance et l'angle entre les vecteurs.

La figure 1.6 suivante représente ce mécanisme [3].

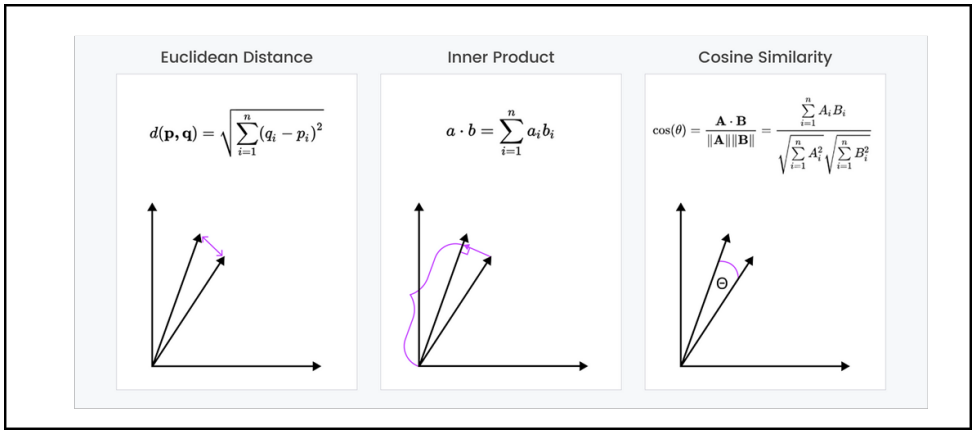


FIGURE 1.6 – Métriques de Similarité Vectorielle.

1.2.5 Base de Données Vectorielle

Une base de données vectorielle est un type spécifique de base de données qui enregistre des informations sous forme de vecteurs multidimensionnels représentant certaines caractéristiques ou qualités [11].

Les bases de données traditionnelles stockent des chaînes des données scalaires sous forme de lignes et de colonnes. En revanche, une base de données vectorielle fonctionne sur des vecteurs, donc la manière dont elle est optimisée et interrogée est assez différente [12].

Une base de données vectorielle utilise une combinaison d'algorithmes qui participent tous à la recherche du voisin le plus proche approximatif. Ces algorithmes sont assemblés dans un pipeline qui fournit une récupération rapide et précise des voisins d'un vecteur interrogé [11].

1.2.6 Index Structurel

L'organisation hiérarchique de l'information constitue une stratégie clé pour améliorer l'efficacité de la récupération d'information [13]. En structurant les données de manière ordonnée, le système RAG peut optimiser à la fois le temps de recherche et le traitement des éléments pertinents.

Index basé sur le graphe de connaissances : Le recours à un KG joue un rôle central dans la construction de cette hiérarchie informationnelle. Il permet d'établir des relations explicites entre les concepts et les entités, garantissant ainsi une cohérence structurelle et réduisant le risque d'erreurs interprétatives. De plus, cette approche permet de transformer les requêtes d'information en instructions sémantiques, facilitant une récupération plus précise. Elle offre aux LLMs un contexte enrichi pour la génération de réponses pertinentes, renforçant ainsi la performance globale du système RAG.

1.2.7 Travaux Récents des RAG dans le Domaine Medical

Le RAG a révolutionné l'application de l'IA dans le domaine médical en permettant une amélioration significative des réponses générées, de la précision des diagnostics et de la personnalisation des traitements. En combinant des modèles de langage de grande taille avec des mécanismes de récupération d'informations externes, Le RAG permet d'obtenir des résultats plus pertinents et adaptés à des situations complexes. Le tableau 1.1 suivant résume quelques exemples de travaux qui ont fait usage des RAG dans le domaine médical.

Projet	Objectif	Données	Méthode	Résultats	Contributions
MedGPT : A Generative Model for Medical Text with Retrieval-Augmented Generation [14]	Utiliser RAG pour améliorer les réponses générées à partir de textes médicaux complexes	Textes médicaux issus de bases de données spécialisées	RAG combiné avec un modèle de langage pré-entrainé sur des textes médicaux	Amélioration de 15% de la précision	Amélioration de la pertinence et de l'exactitude des réponses pour l'automatisation des conseils médicaux
Retrieval-Augmented Generation for Clinical Decision Support Systems [15]	Améliorer les systèmes de support à la décision clinique en fournissant des réponses précises aux questions médicales complexes	Dossiers médicaux électroniques et bases de données cliniques	RAG pour la récupération d'informations cliniques et la génération de recommandations personnalisées	Amélioration significative de l'exactitude de 43 à 99%	Adoption croissante dans les hôpitaux pour améliorer les diagnostics et réduire les erreurs médicales
QA-RAG : A Question-Answering System for Medical Literature with Retrieval-Augmented Generation [16]	Appliquer RAG pour générer des réponses aux questions sur la littérature médicale à partir de bases de données vastes	Articles scientifiques, études cliniques, et bases de données médicales	RAG appliqué à la récupération de documents médicaux et à la génération de réponses	Amélioration de 5 à 15% par rapport aux méthodes traditionnelles	Augmentation de l'efficacité pour traiter des questions médicales complexes
RAG in Radiology : Enhancing Image Report Generation with Retrieval-Augmented Models [17]	Utiliser RAG pour améliorer la génération de rapports radiologiques en intégrant des connaissances externes sur les images médicales	Images médicales (IRM, radiographies) et bases de données textuelles	Intégration de RAG pour combiner données d'image et textes médicaux pertinents	Augmentation de 6-10%	Réduction des erreurs de diagnostic et optimisation de l'interprétation des images médicales

TABLE 1.1 – Quelques Travaux Récents des RAG dans le Domaine Médical.

- **Discussion :** Le tableau 1.1 synthétise plusieurs travaux récents démontrant l'apport des modèles RAG dans le domaine médical. En combinant génération de texte et récupération d'informations pertinentes, ces approches permettent d'améliorer la précision des réponses, la qualité des diagnostics et la pertinence des recommandations. Des gains significatifs ont été observés, notamment une augmentation de l'exactitude jusqu'à 99 % dans les systèmes

de support à la décision clinique et une amélioration de 5 à 15 % dans le traitement de questions complexes. L'application du RAG à l'imagerie médicale montre également une réduction des erreurs d'interprétation grâce à l'intégration de données multimodales. Ces résultats confirment le rôle central du RAG dans l'évolution des outils d'intelligence médicale augmentée.

1.3 Graphe de Connaissance

Les KG jouent un rôle fondamental dans l'organisation et la représentation de données complexes, en structurant l'information selon les relations qu'entretiennent les entités entre elles[18]. Ces structures sont particulièrement pertinentes dans les domaines où les relations entre entités sont aussi importantes que les entités elles-mêmes, comme en médecine. Dans cette section, nous présenterons d'abord les principes fondamentaux des KGs, avant de décrire leurs différentes typologies, puis d'examiner des travaux récents les intégrant dans le domaine médical.

1.3.1 Notion de Base de KG

Un KG constitue une structure de données permettant de modéliser l'information sous forme de nœuds — représentant des entités telles que des concepts, des personnes, des lieux ou des objets — reliés par des arêtes, symbolisant les relations qui les unissent (par exemple : affiliation, localisation, propriété, etc.).
Cette représentation explicite des liens entre entités facilite la visualisation des interconnexions au sein d'un ensemble de données. Les KGs sont particulièrement utiles pour organiser l'information issue de sources textuelles variées — telles que des articles de presse, des archives historiques ou encore des journaux de navigation d'un site web — en leur conférant une structuration sémantique [18].
Cela permet non seulement de mieux comprendre les contextes évoqués dans les textes, mais aussi d'analyser les relations complexes entre les entités mentionnées. Dans le domaine médical par exemple [19], ils permettent de relier des entités telles que les maladies, les symptômes, les traitements ou les gènes. Ainsi, HetioNet [1] connecte 47 000 nœuds (représentant des entités biomédicales telles que des maladies, médicaments ou gènes) via plus de 2,3 millions de relations (par exemple « traite », « exprimé dans », ou « associé à ») [20].
Enfin, cette structuration rend possible une analyse automatisée efficace, facilitant la détection de tendances et de motifs au sein de grands ensembles de données.

1. HetioNet est un graphe de connaissances intégrant des données provenant de 29 bases, représentant plus de 47 000 nœuds et 2,25 millions de relations entre entités biomédicales.

La figure 1.7 illustre un exemple de graphe de connaissances simple. Les entités (ou nœuds), représentées sous forme de cercles — comme *Person*, *Location* et *Vehicle* — sont reliées par des relations directionnelles étiquetées telles que *WORKS-IN* et *DRIVES*. Chaque relation relie une entité source (tête) à une entité cible (queue), formant ainsi un triplet (e_1, r, e_2) tel que $(\text{Resa}, \text{WORKS-IN}, \text{Malmö})$ ou $(\text{Resa}, \text{DRIVES}, \text{Honda})$, dans lequel les propriétés spécifiques aux entités sont également précisées.

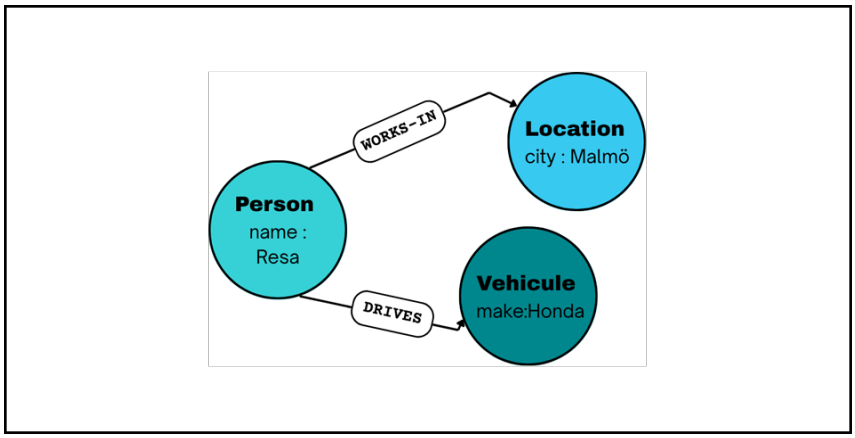


FIGURE 1.7 – Exemple d'un Graphe de Connaissances Simple.

1.3.2 Types de Graphes de Connaissances

Les graphes de connaissances se déclinent en six principaux types, chacun offrant une manière unique de structurer et relier l'information. La figure 1.8 suivante illustre ces différentes types.

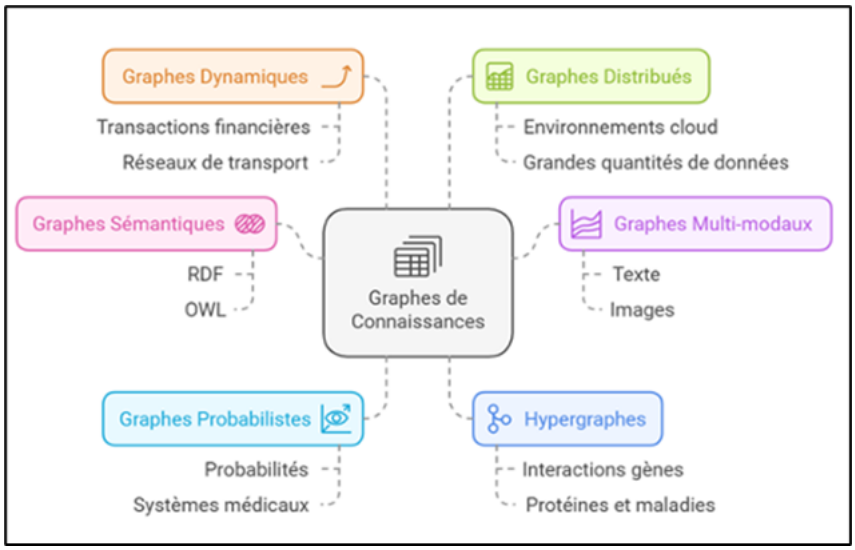


FIGURE 1.8 – Types de Graphes de Connaissances.

- **Graphes de Connaissances Sémantiques :**

  Les graphes de connaissances sémantiques utilisent des normes telles que Resource Description Framework (RDF) et Web Ontology Language (OWL) pour structurer les relations entre les concepts de manière logique, permettant aux machines de les exploiter. Par exemple [21] , les moteurs de recherche comme Google les utilisent pour relier et contextualiser des entités telles que des personnes, des lieux ou des événements afin de fournir des résultats pertinents et riches.

- **Graphes Multi-Modaux :**

  Ils intègrent différentes données (texte, images, audio, etc.) pour mieux comprendre les relations entre les entités. Ils sont par exemple utilisés pour analyser les réseaux sociaux en combinant des interactions textuelles avec des images. [22]

- **Graphes Probabilistes :** Ces graphes utilisent des probabilités pour modéliser l'incertitude ou la fiabilité des relations entre les entités. Ils sont souvent utilisés dans les systèmes médicaux ou d'aide à la décision pour gérer des données incertaines. [22]
- **Hypergraphes :** Les hypegraphes, quant à eux, peuvent représenter des relations complexes impliquant plusieurs entités simultanément. Par exemple [21], ils sont utilisés en bioinformatique pour modéliser les interactions entre gènes, protéines et maladies.
- **Graphes Dynamiques ou Temporels :** Les diagrammes temporels ajoutent une dimension temporelle pour suivre l'évolution des relations au fil du temps. Ils sont utilisés pour analyser des données telles que des transactions financières ou des réseaux de transport. [22]
- **Graphes Distribués :**

Ces structures [22] sont conçus pour être stockés sur plusieurs serveurs, permettant de gérer efficacement de grandes quantités de données. Ils sont souvent utilisés pour traiter des quantités massives de données dans des environnements cloud.

1.3.3 Travaux Récents sur les KGs dans le Domaine Médical

Les KG sont utilisés dans le domaine médical pour améliorer la précision des diagnostics, personnaliser les traitements et comprendre les mécanismes des maladies. En effet des travaux récents montrent à quel point les KG facilitent l'analyse de données biomédicales complexes, l'identification de cibles thérapeutiques, et l'optimisation des soins. Le tableau 1.2 résume quelques exemples de ces travaux dans le domaine médical :

Projet	**Objectif**	**Données**	**Méthode**	**Résultats**	**Contributions**
A Knowledge Graph Approach to Elucidate the Role of Organellar Pathways in Disease via Biomedical Reports [23]	Élucider le rôle des voies organellaires dans les maladies en analysant des rapports biomédicaux	Rapports biomédicaux	Utilisation de graphes de connaissances pour intégrer et analyser des données biomédicales complexes	5 fois supérieure à MedRGB[2]	Identification de nouvelles cibles thérapeutiques et compréhension approfondie des mécanismes pathologiques
Découverte de règles causales dans les graphes de connaissances pour la médecine personnalisée [24]	Identifier des relations causales entre traitements et résultats cliniques pour personnaliser les soins	Données cliniques sur les traitements et résultats	Application de techniques de raisonnement causal sur des KG médicaux	Amélioration de la personnalisation des traitements médicaux, réduisant les effets secondaires de 20%	Amélioration de la précision des traitements personnalisés et réduction des effets secondaires.
Graphe de connaissance et ontologie pour la représentation des critères diagnostiques de la leucémie lymphoïde chronique [25]	Représenter les critères diagnostiques de la LLC à l'aide de KG pour faciliter le diagnostic	Données diagnostiques sur la LLC	Intégration d'ontologies médicales dans un KG pour la représentation des critères diagnostiques	Accélération du diagnostic de la leucémie lymphoïde chronique, réduisant les erreurs humaines de 15%	Accélération du processus diagnostique et réduction des erreurs humaines

TABLE 1.2 – Quelques Travaux Récents sur les KGs dans le Domaine Médical.

- **Discussion :** Les KG ont émergé comme un outil puissant pour structurer et exploiter les données biomédicales complexes. Contrairement aux modèles purement basés sur l'apprentissage automatique, les KG permettent une représentation explicite des relations entre entités, facilitant ainsi l'interprétation et l'explicabilité des résultats. Les recherches récentes démontrent leur efficacité dans divers contextes médicaux, notamment l'identification de nouvelles cibles thérapeutiques, l'amélioration des traitements personnalisés et l'optimisation du diagnostic. En intégrant des ontologies médicales et des règles causales, ces systèmes renforcent la précision des analyses et réduisent les erreurs, comme observé dans le diagnostic de la leucémie lymphoïde chronique. Ces avancées montrent que les KG constituent une approche complémentaire aux modèles de RAG, en apportant une structure explicite aux connaissances utilisées. Dans la section suivante, nous examinerons l'intégration des RAG et des KG dans le domaine médical, en mettant en évidence les bénéfices combinés de ces approches pour améliorer la précision et la pertinence des analyses biomédicales.

1.4 Intégration des RAG et des Graphes de Connaissances

L'intégration des systèmes RAG avec KG vise à combiner la capacité de récupérer des informations pertinentes avec la structure explicite des données. Par exemple le framework MedGraphRAG combine des graphiques hiérarchiques avec des LLMs pour améliorer la précision du diagnostic et la contextualisation des réponses [26]. MedGraphRAG [3] organise les données en métagraphes interconnectés pour faciliter l'accès à des informations médicales complexes tout en suivant l'origine des réponses générées, augmentant ainsi leur transparence et leur fiabilité [27].

1.4.1 Travaux Récents de l'Intégration dans le Domaine Medical

L'intégration des systèmes RAG avec des KG a montré des résultats prometteurs dans le domaine médical, améliorant la précision des prédictions et la pertinence des réponses. Plusieurs études récentes ont démontré son efficacité pour des applications variées, allant de la prédiction de la mortalité hospitalière à la recherche biomédicale. Le tableau1.3 ci-dessous présente ces travaux influents.

3. MedGraphRAG :https ://github.com/SuperMedIntel/Medical-Graph-RAG (consulté en Janvier 2025)

Projet	Objectif	Données	Méthode	Résultats	Contributions
Récupérateur Basé sur Graphes pour Capturer la Longue Queue des Connaissances Biomédicales [28]	Améliorer la récupération d'informations biomédicales rares en utilisant des KG	Littérature biomédicale	Intégration de KG et récupération basée sur la similarité d'embeddings	Performance de récupération deux fois meilleure en précision et rappel	Amélioration la récupération d'informations biomédicales rares en utilisant des graphes de connaissances pour mieux capturer la longue queue des données
EMERGE : Intégration des RAG pour Améliorer la Modélisation Prédictive Multimodale [29]	Améliorer de la modélisation prédictive des dossiers de santé électroniques (EHR) multimodaux	MIMIC-III, MIMIC-IV	Extraction d'entités, alignement avec PrimeKG, fusion multimodale adaptative avec attention croisée	Amélioration de 12% par rapport aux modèles classiques	Amélioration de la précision des prédictions cliniques en utilisant des techniques avancées pour analyser les dossiers de santé électroniques
Apprentissage Graphique Multimodal sur les KG UMLS [30]	Apprendre des représentations significatives de concepts médicaux en utilisant des KG-UMLS	MIMIC-III	Réseaux de neurones graphiques sur des graphes UMLS, agrégation pour représenter des visites de patients	Améliorer la précision de la prédiction d'environ 10 à 15% par rapport aux méthodes classiques	Surpasse les méthodes existantes en incorporant des connaissances médicales préalables
Numéro 94 de la lettre d'information Digital Watch – novembre 2024 [31]	Améliorer des réponses des assistants numériques en intégrant RAG et KG	Sources d'information variées, y compris des KG	Méthodes avancées d'IA pour fournir des réponses précises et contextuelles	Augmentations de précision allant de 10 à 25%	Permettant une meilleure compréhension contextuelle et une réponse plus adaptée aux besoins des utilisateurs.

TABLE 1.3 – Quelques Travaux Récents sur l'Intégration des RAG et KGs dans le Domaine Medical.

- **Discussion :** Les travaux analysés soulignent l'impact significatif de l'intégration des modèles de RAG et des KG dans le domaine médical, notamment pour améliorer la récupération d'informations biomédicales, la modélisation prédictive et la précision des systèmes d'intelligence artificielle.

  L'utilisation des KG pour la récupération d'informations rares permet de doubler la précision et le rappel par rapport aux approches classiques, facilitant ainsi l'extraction de données dans des bases biomédicales vastes et complexes. L'intégration des RAG et

des KG dans la modélisation prédictive des dossiers de santé électroniques multimodaux améliore la précision des prédictions cliniques de 12%, favorisant une anticipation plus proactive des complications médicales. De plus, l'apprentissage graphique multimodal sur les KG-Unified Medical Language System (UMLS) améliore la précision des prédictions médicales de 10 à 15%, surpassant ainsi les méthodes traditionnelles. Enfin, l'application des RAG enrichis par des KG dans les assistants numériques augmente la précision des réponses médicales de 10 à 25%, rendant ces systèmes plus fiables et adaptés aux besoins des utilisateurs. Ces avancées témoignent des bénéfices de l'intégration des RAG et des KG en termes de précision diagnostique, fiabilité des réponses, ainsi que d'optimisation des processus de recherche et de traitement dans le domaine médical.

Cependant, cette intégration soulève plusieurs défis. L'un des principaux concerne l'alignement des données, car l'intégration de données RAG non structurées avec des données KG structurées nécessite des algorithmes complexes pour gérer les différences de terminologie et de relations. Par ailleurs, l'évolutivité et les performances représentent un autre défi majeur, notamment lorsque le KG est volumineux, entraînant une surcharge de calcul et de mémoire dans le pipeline RAG. Enfin, la maintenance des cartes constitue un enjeu clé, car le système RAG-KG exige des mises à jour fréquentes des connaissances pour suivre l'évolution rapide des progrès médicaux.

1.5 Conclusion

Les technologies de pointe démontrées mettent en évidence les progrès récents dans l'utilisation des systèmes RAG et KG pour répondre aux besoins croissants de l'IA. Chaque composant aide à résoudre un problème spécifique, mais présente également des limites qui nécessitent une approche intégrée. En résumé, l'intégration de RAG et KG représente une avancée significative dans le domaine de l'IA médicale. Ces systèmes promettent une précision accrue, une contextualisation et une fiabilité améliorées dans les environnements critiques. Le succès de cette intégration dépendra de la capacité à surmonter les défis techniques et à garantir une adoption clinique à grande échelle, ce qui en fera une priorité pour les recherches futures.

Dans le chapitre suivant nous allons présenter toutes les étapes de conception et de réalisation de notre système RAG.

Chapitre II : Conception et Modélisation de la Solution

2.1 Introduction

La génération automatique de réponses pertinentes, précises et contextualisées constitue un des éléments fondamentaux des systèmes d'assistance médicale reposant sur l'IA. Lorsqu'on interroge le système, il ne suffit pas seulement de récupérer les unités d'information il faut être capable d'accéder à des réponses exprimées dans une forme articulée, compréhensible, scientifiquement reconnue. Pour cela, nous nous appuyons sur une architecture avancée dite RAG qui combine les capacités de récupération d'informations depuis un KG et les capacités génératives des modèles de langage.

Dans ce chapitre, nous détaillons l'ensemble du processus que nous avons conçu : depuis le choix et la préparation du dataset, jusqu'à l'encodage des entités sous forme d'embeddings à l'aide de plusieurs modèles de représentation, puis l'appariement et l'intégration de ces entités dans le KG. Nous clarifions également comment sont formulées et traitées les requêtes pour interroger le KG de façon efficace avant d'être traitées par les modèles de génération, qui fournissent les réponses finales. Nous étudions enfin l'évaluation du système RAG afin d'évaluer ses performances en termes de précision, de pertinence et de qualité de réponse au service des utilisateurs.

2.2 Processus Global du Système de Génération Augmentée par Récupération

La figure 2.9 illustre les étapes que nous avons développé pour tester notre système :

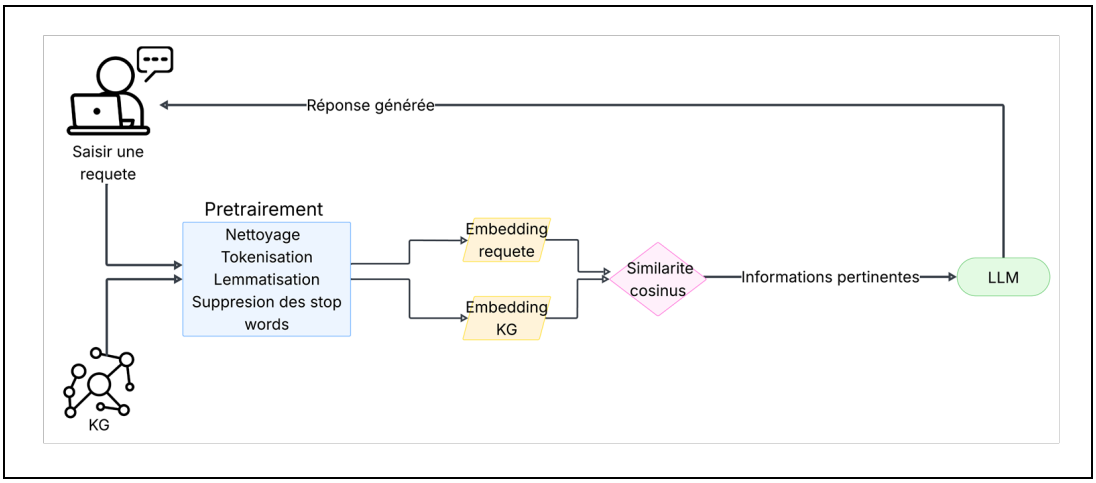
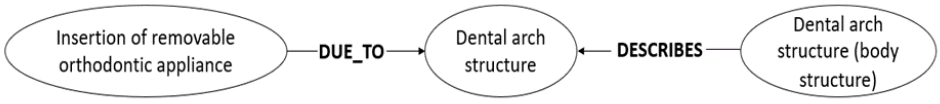


FIGURE 2.9 – Processus Global du Système.

Dans les sections qui suivent nous allons décrire en détails la logique de chacune des étapes de cette architecture

2.3 Collection de Données

Pour nos expérimentations, nous avons utilisé un sous-ensemble du dataset SNOMED CT [4] International Edition, disponible sur le site officiel de la National Library of Medicine (NLM). Ce jeu de données est constitué de concepts médicaux, de descriptions associées et de relations sémantiques entre ces concepts. Il est mis à jour régulièrement pour refléter les dernières avancées en matière de terminologie clinique. Le format utilisé pour les données est le Release Format 2 (RF2), qui organise les informations sous forme de fichiers tabulaires. Ces fichiers contiennent des informations détaillées sur les concepts, leurs descriptions, ainsi que les relations entre les différents concepts. Voici un exemple illustrant un triplet enrichi d'une description liée au concept cible :



Afin de garantir la qualité et la pertinence des données utilisées, une phase de filtrage a été réalisée. Le processus consiste à sélectionner uniquement les concepts actifs dans le dataset et à extraire les descriptions et relations associées à ces concepts.

Cette étape de filtrage permet de conserver uniquement les concepts actifs, ainsi que leurs descriptions en anglais et de type pertinent, tels que les synonymes ou les termes préférés. De plus, seules les relations entre concepts actifs ont été conservées. Les données filtrées ont ensuite

4. SNOMED CT https://www.nlm.nih.gov/healthit/snomedct/international.html (Consulté le :03/2025)

été intégrées dans Neo4j est un outil qui permet de visualiser et d'exploiter facilement un KG . Le tableau 2.4 montre le nombre de données du dataset :

Nombre total de concepts	**Nombre total de descriptions**	**Nombre total de relations**
10 000	35 992	39 137

TABLE 2.4 – Statistiques du Dataset.

Pour illustrer la structure du KG obtenu à partir des données filtrées, la figure 2.10 présente une vue d'ensemble du graphe, où chaque nœud représente un concept médical et chaque flèche correspond à une relation sémantique entre deux concepts. Cette représentation permet de visualiser la richesse et la complexité des interconnexions dans le domaine médical.

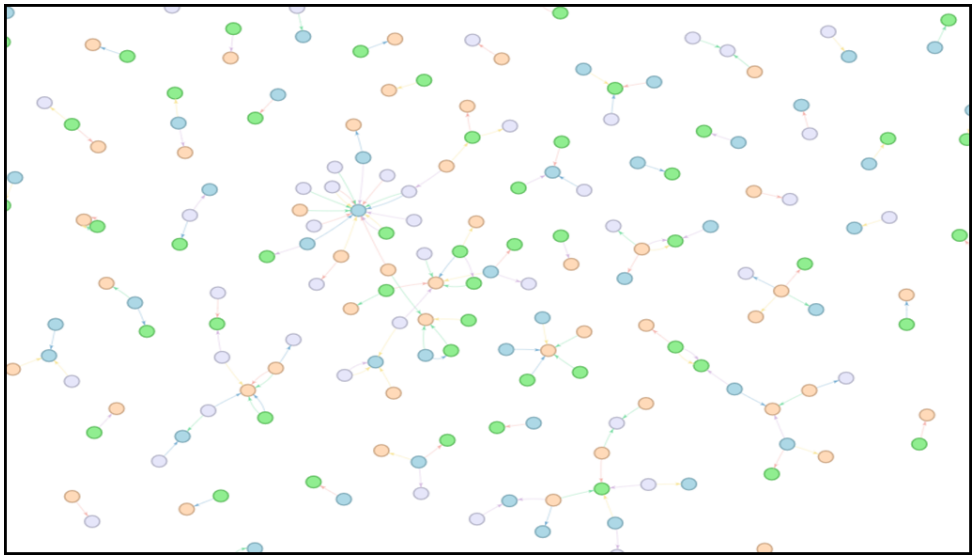


FIGURE 2.10 – Visualisation Globale du Graphe de Connaissances.

Par ailleurs, afin de mieux comprendre la structure d'un triplet, la figure 2.11 propose un zoom sur un exemple spécifique de relation entre deux concepts, mettant en évidence le lien sémantique qui les unit. Ces visualisations facilitent la compréhension de la structuration des données dans le graphe.

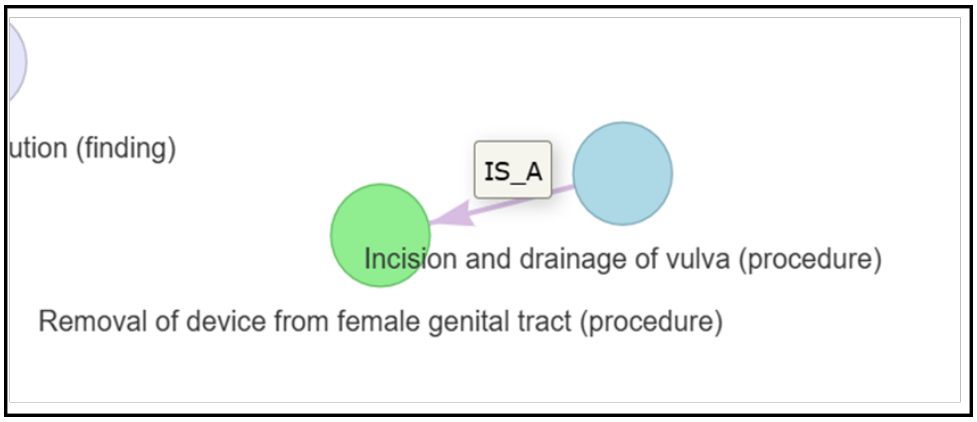


FIGURE 2.11 – Exemple Détaillé d'un Triplet.

2.4 Prétraitement des Données

Le prétraitement des données constitue une phase critique dans l'ensemble d' un processus d' analyse textuelle , en particulier celui du NLP discipline qui contribue à la transformation des données brutes, hétérogènes et souvent désordonnées en données exploitables par les algorithmes d' analyse ou d'apprentissage automatique .

D' après [32]. le prétraitement constitue une étape incontournable du NLP, il vise à transformer un texte brut en version simplifiée, cohérente et standardisée qui en conséquence plus aisément interprétable par un système informatique. Cela facilite la compréhension automatique du message émis au travers du texte et l' extraction d' informations pertinentes. En effet, un texte brut peut contenir des éléments perturbateurs tels que des chiffres isolés, des signes, des symboles, des erreurs de typographie, etc, qui doivent être éliminés ou neutralisés pour interpréter correctement et se focaliser sur l'analyse des données linguistiques vraiment pertinentes. Dans le cadre de notre projet, nous avons appréhendé la phase de prétraitement sur l'intégralité des nœuds et des relations du KG médical référentiel stockées en Neo4j. L' application de ce traitement nous a permis d'instaurer une cohérence textuelle au sein de la base de données tout en améliorant les performances des opérations de recherche et de raisonnement les étapes suivantes ont été réalisées :

· Mise en Minuscules : Conversion de tout le texte en minuscules pour éviter les différences de casse qui pourraient affecter la cohérence de l'analyse. Par exemple, "Cancer", "cancer" et "CANCER" sont normalisés en une seule forme, ce qui évite les doublons lors de l'analyse.

· Suppression des Caractères Spéciaux : Élimination des caractères non alphanumériques, tels que les symboles de ponctuation et les emojis, qui n'apportent généralement pas d'informations utiles à l'analyse.

· Tokenisation : Découpe des phrases en unités linguistiques de base tokens (souvent mots) pour analyser les éléments déterminants par la fréquence d ' occurrences ou encore indexations.

· Suppression des Stop Words : Élimination des mots courants (par exemple, "le", "de", "et") ces mots n'apportent généralement pas de valeur ajoutée à l'analyse sémantique.

· Lemmatisation : L ' opération de lemmatisation a permis de réduire la forme de chaque mot à sa forme canonique (lemme) .Par exemple : « traités » , « traiter » , « traitement » peuvent être réduites es à leur lemme « traiter » . Cela permet d' unifier les occurrences morphologiques d' un même concept lexical pour faciliter leur modélisation sémantique.

Le tableau 2.5 présente des exemples issus de textes biomédicaux, en indiquant le contenu **avant** et **après** chaque traitement :

Étape de prétraitement	**Avant prétraitement**	**Après prétraitement**
Mise en minuscules	The PATIENT Was Diagnosed With Diabetes.	the patient was diagnosed with diabetes.
Suppression des caractères spéciaux	blood-pressure ? check-up !	blood pressure check up
Tokenisation	blood pressure check up	['blood', 'pressure', 'check', 'up']
Suppression des stop words	['the', 'patient', 'was', 'diagnosed']	['patient', 'diagnosed']
Lemmatisation	diagnosed, diagnosing, diagnosis	Diagnose

TABLE 2.5 – Exemples de Prétraitement.

Pour chaque nœud et relation une propriété supplémentaire a été ajoutée, contenant la version nettoyée du nom d'origine.

2.5 Génération des Embeddings Sémantiques

Les embeddings sémantiques sont absolument centraux pour notre projet dans lequel nous voulons représenter et exploiter les connaissances biomédicales. L'objectif principal est de faciliter la transformation des entités (les nœuds) et des relations du KG en un vecteur numérique continu dans un espace continu de dimension fixée, tout en gardant leur sens sémantique, la figure 2.12 suivante représente le processus de génération d'embedding.

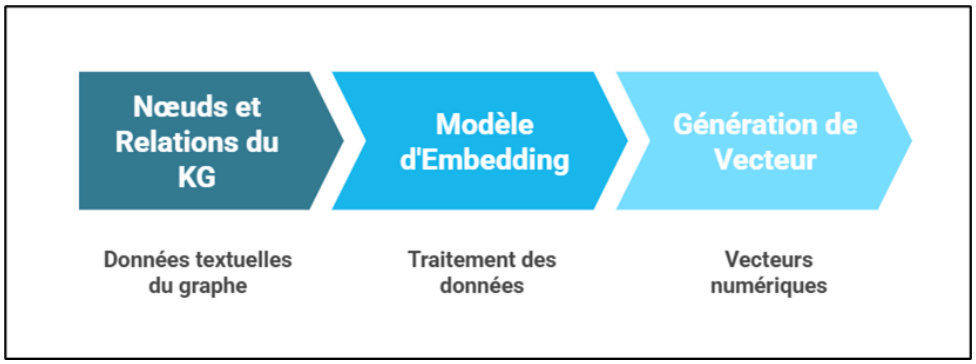


FIGURE 2.12 – Processus de Génération d'Embedding.

L'objectif que cette vectorisation doit permettre de réaliser est de récupérer des relations sémantiques implicites du KG, de telle sorte que les entités qui sont proches sémantiquement soient aussi proches géométriquement dans l'espace vectoriel.

À titre d'exemple des entités biomédicales étroitement connues comme Diabètes et Insulin Resistance sont dotées de vecteurs assez proches, représentant ainsi leur proximité sémantique : Embedding(Diabetes) $\approx Embedding(InsulinResistance)$

Par contraste des relations de nature différente souvent présentes dans les mêmes contextes, comme treats et causes, continuent à être dotées de représentations clairement différenciées, traduisant le sens respectif des relations : Embedding(treats) $\neq Embedding(causes)$

Dans notre travail, nous avons adopté des modèles d'embedding différents afin de comparer leurs performances dans l'intégration des informations textuelles du KG et notre système RAG. Chacun de ces modèles sera présenté et expliqué dans les sections qui suivent.

2.5.1 Modèle BioBERT

D'après [33]. La Bidirectional Encoder Representations from Transformers for Biomedical Text Mining (BioBERT) [5] est une version spécialisée de Bidirectional Encoder Representations from Transformers (BERT) conçue pour capturer les subtilités du langage biomédical.

Selon [34], le modèle BioBERT est ainsi initialisé à partir des poids du modèle BERT, qui a été préalablement entraîné sur des corpus généralistes issus de Wikipedia [6] et de BooksCorpus [7], puis réentraîné spécifiquement sur des données du domaine biomédical, résumés de PubMed [8] et articles en texte intégral PubMed Central [9].

L'efficacité de de ce modèle est évaluée sur trois tâches clés de fouille de texte biomédical, la reconnaissance d'entités nommées, l'extraction de relations et la réponse à des questions en utilisant diverses stratégies de préentraînement, faisant appel à différents corpus du domaine

5. BioBERT https://huggingface.co/dmis-lab/biobert-base-cased-v1.1 (Consulté le : 04/2025)
6. Wikipedia (https ://www.wikipedia.org/)
7. BooksCorpus https ://huggingface.co/datasets/SamuelYang/bookcorpus
8. PubMed https ://pubmed.ncbi.nlm.nih.gov/
9. PubMed Central https ://www.ncbi.nlm.nih.gov/pmc/

général et biomédical, ainsi que leur combinaison permettant d'analyser les apports possibles de chaque type de données sur la performance du modèle. Dans un second temps, une analyse plus poussée des résultats entachant BERT et BioBERT permet de bien rendre compte de l'intérêt et de la pertinence des corpus spécialisés pour contribuer à renforcer les performances sur les tâches biomédicales. Ce processus permet à BioBERT de comprendre les termes médicaux et scientifiques qui manquent fréquemment dans les corpus standards [34].

Dans le cadre de notre travail BioBERT est appliqué pour générer des représentations vectorielles (embeddings) pour des textes biomédicaux. Les ensemble des **nœuds** et des **relations** sont mappés à des vecteurs dense représentant son sens dans un contexte de santé. De tels intégrations sont utilisées pour améliorer notre KG ainsi que pour augmenter la précision des systèmes de recherche et de recommandation dans le domaine de la santé.

Le schéma 2.13 ci-dessous présente les principales étapes de la génération des embeddings avec le modèle BioBERT.

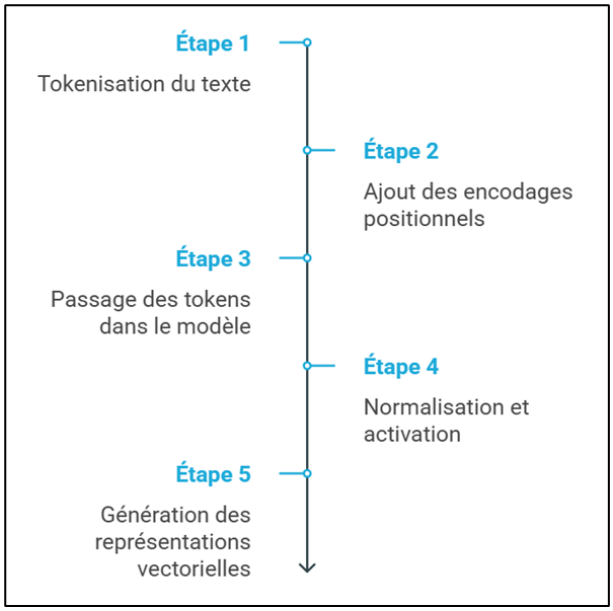


FIGURE 2.13 – Étapes du Génération des Embeddings avec le Modèle BioBERT.

Nous alllons ci-dessous, expoliquer chaque étape et clarifier son rôle et son importance dans le cadre de notre système.

1. Tokenisation du texte : La première étape pour chaque noeud et relation du KG elle consiste à décomposer les unités en tokens unités élémentaires du texte qui pourront être prises en compte par le modèle BioBERT. Cette opération est réalisée par le tokenizer qui opère sur le texte brut pour le convertir en séquence de tokens numériques.

2. **Ajout des encodages positionnels :** Avant d'entrer dans le modèle, chaque token est doté d'un encodage positionnel. L'importance de cette étape vient du fait qu'elle permet au modèle de conserver la position relative des mots dans la phrase, ce qui est d'importance pour

le modèle car l'architecture Transformer ne fait pas le travail d'organisation linéaire des mots traditionnellement requis.

3. Passage des tokens dans le modèle BioBERT : Une fois positionnés les tokens sont introduits dans le modèle BioBERT qui se compose d'un grand nombre d'unités identiques organisées en couches qui elles s'appellent des Transformers. Chaque couche dispose d'un mécanisme permettant au modèle de comparer chaque mot aux autres mots de la phrase dans le cadre d'un processus d'évaluation de quel mot va être considéré dans sa contribution relative et donc d'affiner la représentation de chaque token en tenant compte du contexte.

4. Normalisation et activation : Entre chaque couche, des opérations de normalisation et des fonctions d'activation non linéaires sont appliquées. Elles assurent la stabilité de l'apprentissage et permettent au modèle de mieux capter les relations complexes présentes dans le texte.

5. Génération des représentations vectorielles : Après le passage complet dans toutes les couches le modèle produit un tenseur. Ce tenseur contient un vecteur dense pour chaque token enrichi par le contexte global du texte.

L'algorithme 1 décrit le processus utilisé pour générer les représentations sémantiques à l'aide du modèle BioBERT pour les entités (nœuds) et les relations du graphe de connaissances médicales.

Algorithme 1 : Génération des embeddings sémantiques avec BioBERT pour nœuds et relations

Input : Ensemble de nœuds N et de relations R extraits du graphe de connaissances médicales

Output : Embeddings vectoriels E_N pour les nœuds et E_R pour les relations

Initialisation : Charger le modèle BioBERT et son tokenizer;

Pour chaque nœud n dans N : Extraire le nom nettoyé du nœud t_n;

    Tokeniser t_n avec le tokenizer BioBERT;

    Passer les tokens dans BioBERT sans mise à jour des poids (inférence);

    Extraire le tenseur de sortie (représentation contextuelle);

    Moyenniser les vecteurs de chaque token pour obtenir l'embedding e_n;

    Stocker e_n dans Neo4j comme attribut du nœud n;

Pour chaque relation r dans R : Nettoyer et normaliser le nom de la relation t_r (lemmatisation, suppression des stop words, remplacements);

    Tokeniser t_r avec BioBERT;

    Passer les tokens dans BioBERT en mode inférence;

    Extraire et moyenniser les vecteurs pour obtenir l'embedding e_r;

    Stocker e_r dans Neo4j comme propriété de la relation r;

2.5.2 Modèle SciSpaCy

Scientific spaCy (SciSpaCy) [10] est une extension de la bibliothèque spaCy, optimisée pour le NLP dans le domaine biomédical et scientifique. Elle propose des modèles linguistiques entraînés sur des corpus spécialisés comme PubMed, afin d'assurer une meilleure couverture des entités, termes et relations spécifiques à ce domaine. Dans notre étude, nous avons utilisé le modèle `en_core_sci_lg`, une version large intégrant des vecteurs de mots pré-entraînés de 200 dimensions, permettant une représentation dense et sémantiquement riche des textes scientifiques. Il repose sur un pipeline NLP composée de plusieurs modules successifs :

• **Tokenisation :** qui segmente le texte en unités lexicales (tokens) ;

• **POS-tagging :** qui assigne à chaque mot sa catégorie grammaticale (nom, verbe, adjectif, etc.) ;

• **Parser :** qui identifie les relations de dépendance syntaxique entre les tokens ;

• **Reconnaissance d'entités nommées :** (NER) qui permet de détecter des entités biomédicales telles que des maladies, des substances ou des organes.

Chaque mot est ensuite représenté par un vecteur dense issu de l'espace sémantique du modèle, et un embedding global du texte est obtenu en calculant la moyenne des vecteurs de tous les tokens du texte.

Le tableau 3.11 ci-dessous illustre ces différentes étapes appliquées à l'expression "acute myocardial infarction" :

Étapes	**Résultat**
Texte d'entrée	"acute myocardial infarction"
Tokenisation	['acute', 'myocardial', 'infarction']
POS-tagging	['ADJ', 'ADJ', 'NOUN']
Analyse syntaxique (parser)	acute $\rightarrow$ modifies $\rightarrow$ infarction
Reconnaissance d'entités nommées (NER)	['acute myocardial infarction']
Embedding par token	[0.12, 0.08, ..., 0.45] [0.12, 0.08, ..., 0.45] [0.12, 0.08, ..., 0.45] (3 vecteurs de 200D)
Vecteur global du texte	0.15,0.07,...,0.39 (vecteur final 200D)

TABLE 2.6 – Étapes de Génération des Embeddings avec le Modèle SciSpaCy.

10. SciSpaCy en_core_sci_lg https://allenai.github.io/scispacy/ (Consulté le : 05/2025)

2.5.3 Modèle Word2vec

Word2Vec [11] est un modèle d'apprentissage distribué des représentations lexicales visant à projeter les mots d'un vocabulaire dans un espace vectoriel continu de dimension fixe, de manière à capturer leurs propriétés sémantiques et syntaxiques. Dans cet espace, les mots ayant des contextes similaires dans les corpus d'entraînement sont représentés par des vecteurs proches, facilitant ainsi l'inférence de relations sémantiques à travers des opérations algébriques simples sur les vecteurs.

Comme illustré dans la figure 2.14 , le modèle propose deux architectures distinctes [4] : Skip-gram et CBOW

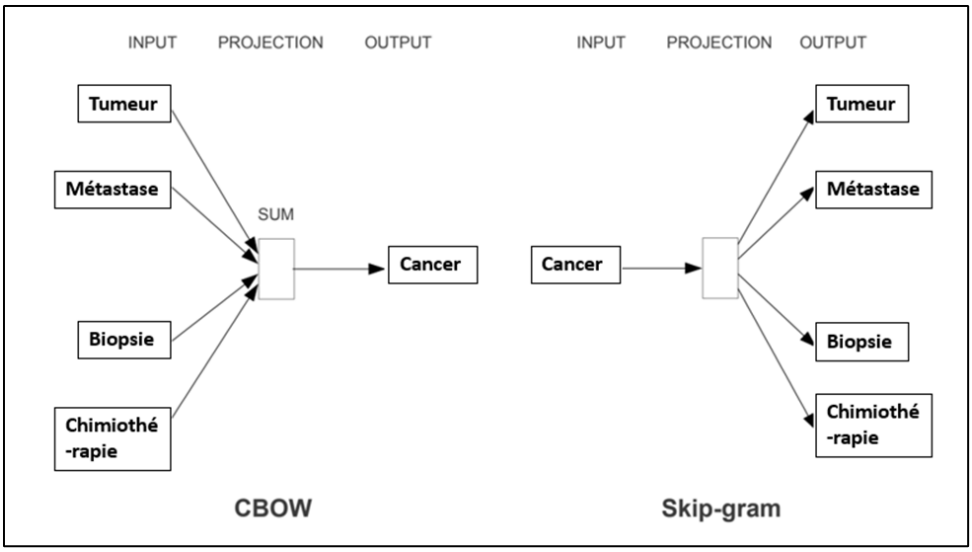


FIGURE 2.14 – Architectures du Modèle Word2vec [4].

Continuous Bag of Words (CBOW) : cette architecture prédit un mot cible à partir de son contexte, c'est-à-dire les mots qui l'entourent dans une fenêtre glissante.

Skip-Gram : à l'inverse, cette architecture apprend à prédire le contexte d'un mot donné, ce qui la rend particulièrement efficace pour représenter des mots rares.

Dans notre projet, nous avons utilisé un modèle Word2Vec pré-entraîné sur des corpus biomédicaux tels que PubMed et PMC, ce qui permet de bénéficier de représentations vectorielles adaptées au vocabulaire médical (telles que les vecteurs ont une dimension de 200). Ce modèle a été entraîné à l'aide de l'architecture Skip-Gram, qui s'est révélée particulièrement efficace pour apprendre des représentations précises, même pour les mots rares – ce qui est crucial dans le domaine médical.

11. Word2Vec préentraîné sur les corpus biomédicaux PubMed et PMC, fourni par le projet BioASQ. https://bioasq.org/ (Consulté le : 05/2025)

2.5.4 Comparaison des Modèles

Le tableau 2.7 compare les trois modèles : BioBERT, SciSpaCy et Word2Vec. Il met en évidence leurs architectures, méthodes d'apprentissage et caractéristiques clés.

Caractéristiques	**BioBERT**	**SciSpaCy**	**Word2Vec**
Architecture	Transformer basé sur BERT (12 couches, 768 dimensions, attention multi-tête)	Pipeline spaCy avec embeddings statiques et composants NLP (CNN pour embeddings)	Réseau shallow (CBOW ou Skip-gram)
Type d'embeddings	Contextualisés, basés sur token (WordPiece tokenizer)	Statique, vecteurs GloVe-like appris sur corpus biomédical	Statique, vecteurs de mots appris sur corpus biomédical
Corpus d'entraînement	PubMed abstracts + PMC full texts	Corpus scientifique biomédical large	PubMed abstracts + PMC full texts
Méthode d'apprentissage	Pré-entraînement masked language model + fine-tuning sur tâches biomédicales	Pré-entraînement statique basé sur co-occurrence et NLP	Prédictif (CBOW ou Skip-gram)
Dimension des embeddings	768 dimensions	300 dimensions	200 dimensions
Contextualisation	Oui, embeddings varient selon le contexte de la phrase	Non, embeddings fixes pour chaque mot	Non, embeddings fixes pour chaque mot
Limitations	Coût computationnel élevé, nécessite GPU pour inférence rapide	Embeddings statiques limités pour contexte complexe	Ne gère pas l'ambiguïté ou le contexte des mots

TABLE 2.7 – Comparaison des Modèles de Génération des Embeddings.

Comme montré dans ce tableau, chaque modèle présente des forces et des limites spécifiques. BioBERT se distingue par sa capacité à produire des embeddings contextuels riches, adaptés aux tâches complexes en biomédecine, bien que son coût computationnel soit élevé. SciSpaCy offre une solution rapide et pratique avec des embeddings statiques optimisés pour le domaine scientifique, tandis que Word2Vec, plus léger, reste efficace pour des tâches nécessitant des représentations simples et rapides. Après cette analyse, nous allons effectuer une évaluation pour déterminer quel modèle serait le plus adapté à notre projet.

2.6 Appariement entre Graphe de Connaissance et Requête

L' appariement entre la requête de l' utilisateur exprimée en langage naturel et les entités du KG constituent une étape centrale des systèmes RAG . Pour produire une réponse pertinente à une question en langage naturel, notre système met en œuvre une stratégie d'appariement sémantique entre la requête de l' utilisateur et les entités (nœuds) et les relations du KG médicales. Ce processus permet de trouver dans le graphe, les éléments les plus proches du sens exprimés par l'utilisateur, dans un contexte sémantique biomédical.

La figure 2.15 représente les étapes d'appariement entre requête et KG :

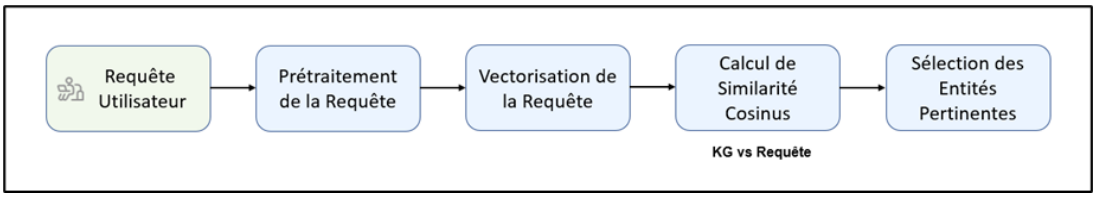


FIGURE 2.15 – Etapes d'Appariement entre la Requête et KG.

- **Prétraitement de la requête :** Pour garantir une cohérence linguistique entre la requête et les éléments du KG la requête de l'utilisateur est tout d 'abord soumise aux mêmes traitements que ceux appliqués aux entités du graphe : mise en minuscules, suppression des caractères spéciaux, tokenisation, suppression des mots vides, lemmatisation.
- **Géneration des Embadding :** Une fois nettoyée la requête est vectorisée sous la forme d'un vecteur dense à l' aide du modèles des embadding prend en compte les particularités sémantiques du domaine médical en accordant le contexte dans lequel apparaissent les termes.
- **Calcul de la similarité :** Pour réaliser l'appariement nous avons fait appel à une mesure de similarité cosinus entre le vecteur de la requête et l'ensemble des vecteurs d'entités et de relations des KG. Comme méthode heuristique de mesure de similarité entre deux vecteurs la similarité cosinus est définie à partir du cosinus de l'angle qui sépare les deux vecteurs. Cette mesure est fréquemment utilisée dans les modèles en recherche d'information et en fouille de texte pour évaluer le rapprochement sémantique entre deux vecteurs dans un espace multidimensionnel.

  La similarité cosinus de deux vecteurs A et B est calculée comme suit :

$$\text{Sim}_{\cos}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \times \|\vec{B}\|}$$

  Plus le cosinus est proche de 1, plus les vecteurs sont similaires, tandis qu'un cosinus proche de 0 indique une similarité faible.

Le système sélectionne alors les entités ayant la plus grande similarité à la requête, et considérées comme correspondantes au plus aux attentes dans le KG à partir desquelles seront extraites ou générées les réponses

Nous présentons ici un exemple d'appariement entre une requête médicale utilisateur et les entités du KG :

Input : What is the associated morphology of androgen dependent hirsutism ?

Output :

· Nœud similaire : androgen dependent hirsutism

· Relation extraite : associated morphology

· **Cible** : *female structure body*

· **Similarité** : 0.9497

2.7 Intégration du Modèle de Génération dans le Système RAG

Après avoir réalisé la première étape de notre système RAG qui consiste en l'extraction d'informations pertinentes à partir du KG suite à l'appariement entre la requête de l'utilisateur et les entités du graphe, il devient essentiel de générer une réponse textuelle claire et contextualisée à partir des éléments extraits.

Le modèle constitue le cœur de la génération finale de réponse. Il joue le rôle de **composants générateurs** permettant de réaliser du texte médical cohérent pertinent et informatif à partir d'un prompt enrichi par des informations extraites du KG . Voici les étapes du fonctionnement du modèle de génération :

- **Source contextuelle** : À partir de l'appariement entre la requête et les entités du KG, nous extrayons les informations médicales correspondant le mieux sémantiquement (entités et relations). Ces éléments sont pour former une source enrichie au modèle .
- **Prompt textuel** : Une fois la source contextuelle, elle est mise en relation avec un prompt textuel standardisé, ici sous la forme : « Based on this knowledge, we can say that : ». Ce prompt est la commande d'exécution pour le modèle qui débute la réponse.
- **Étapes de la génération :** Une fois la source et le prompt fusionnées, le modèle passent dans la phase de génération autoregressive qui se déploie token par token, soit mot par mot, soit morceau par morceau. C'est la condition de la logique de la réponse et de la cohérence avec les informations médicales disponibles

La figure 2.16 résume les étapes d'intégration du modèle de génération dans notre système RAG :

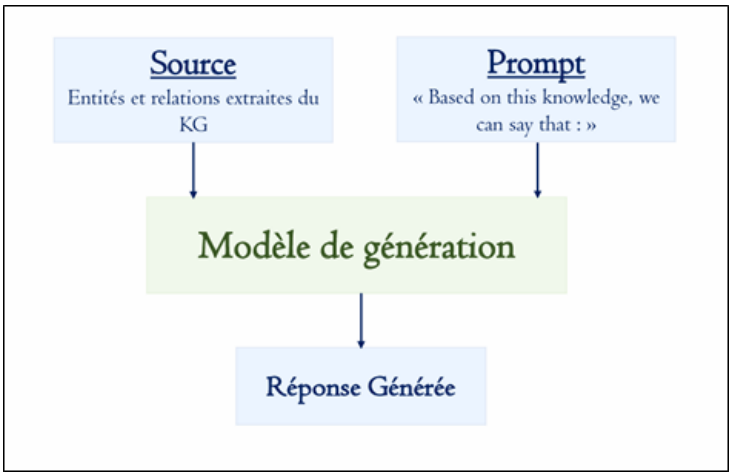


FIGURE 2.16 – Processus du Modèle de Génération.

Dans le cadre de notre système RAG nous avons utilisé trois modèles de génération en mode d'inférence : **BioGPT** [12] ,**GPT-2** [13] et **Flan-T5-Base** [14] ou nous allons définir chaque modèle utilisé et détailler son fonctionnement en mettant en évidence les différentes étapes de génération de texte .

2.7.1 Modèle BioGPT

Generative Pre-trained Transformer for Biomedical Text Generation and Mining (BioGPT) est un modèle de langage pré-entraîné spécialisé en biomédical, qui repose sur l'architecture Transformer. Pré-entraîné sur des résumés d'articles scientifiques PubMed il génère des sentences apparemment cohérentes et pertinentes au sein de la biomédecine. Il reçoit en entrée d'une part une source contextuelle et d'autre part un prompt textuel et génère autoregressivement des séquences biomédicales. Ce modèle semble en faveur pour une génération textuelle médicale rencontrée en RAG en connectant des informations extraites de KG [35].

- **Phase d'entraînement (training)**

Au moment de l'entraînement le modèle était alimenté avec la séquence qui contient :

— · **Source :** une entité biomédicale ou un concept qui peut être une protéine ou un médicament (ex : l'insuline)

— **prompt :** une phrase ou fragment de texte qui va commencer à amorcer une inférence logique (ex : we can conclude that)

12. BioGPT https://github.com/microsoft/BioGPT (Consulté le : 04/2025)
13. GPT2 https ://huggingface.co/openai-community/gpt2 (Consulté le : 04/2025)
14. Flan-T5 https ://huggingface.co/google/flan-t5-base (Consulté le : 04/2025)

— **target :** la séquence cible que le modèle va devoir apprendre à prédire (ex : the interaction between A and B is R).

Donc BioGPT était entraîné à générer des phrases biomédicales qui soient à la fois cohérentes avec le contexte donné et scientifiquement pertinentes en le rendant capable de compléter des affirmations biomédicales sous un contexte donné.

- **Phase d'inférence (inference)**

Une fois que l'entraînement a été effectué, le modèle est utilisé en phase d'inférence pour générer automatiquement des textes biomédicaux à partir d'un contexte partiellement connu. Dans ce cas : Le source et prompt sont fournis à BioGPT.

C'est le modèle qui va produire automatiquement le target et qui va compléter la phrase en produisant une formulation qui soit cohérente et plausible par rapport à ses connaissances.

La figure 2.17 représente le fonctionnement du modèle BioGPT dans notre système RAG .

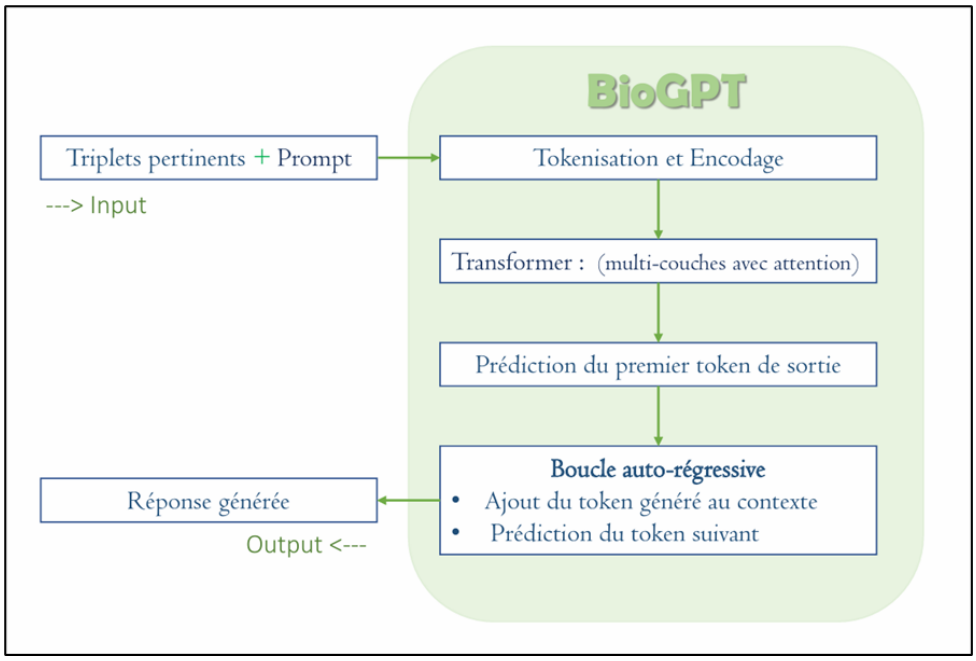


FIGURE 2.17 – Fonctionnement du Modèle BioGPT.

BioGPT reçoit en entrée un ou plusieurs triplets extraits du RAG constitué d'entités et de leur relations pertinentes ainsi qu'un prompt qui guide la génération. Le modèle commence par tokeniser et encoder cette entrée, puis traite les tokens dans un réseau Transformer profond avec mécanisme d'attention spécialement adapté au domaine biomédical. Ensuite le premier token de sortie est prédit. S'installe alors un fonctionnement auto-régressif car à chaque étape, le token généré vient s'ajouter au contexte pour prédire le token suivant, et ce jusqu'à atteindre un token de fin ou dépasser la limite de longueur. C'est ainsi que l'on peut produire des réponses textuelles sémantiquement précises et lexicalement adaptées au langage biologique des données biomédicales

2.7.2 Modèle GPT-2

Generative Pre-trained Transformer 2 (GPT-2) est un modèle de langage généré par OpenAI basé sur l'architecture. Faisant partie des modèles autoregressifs pré-entraînés sur de grands corpus de textes récupérés sur Internet (environ 40 Go de textes filtrés collectés sous la dénomination WebText), le modèle est entraîné sur un objectif de modélisation du langage classique : prédiction du mot suivant (en étant conditionnée par le contexte donné par les mots précédents) [36].

Contrairement aux approches classiques spécifiquement supervisées conçues pour des tâches précises, GPT-2 est conçu comme un modèle multitâche non supervisé, capable de réaliser des tâches variées du traitement du langage naturel sans ajustement spécifique à chaque tâche. Grâce à sa taille importante (1.5 milliard de paramètres pour la plus grande version) et à son pré-entraînement sur des données variées, GPT-2 peut générer des textes longs, cohérents, et contextuellement pertinents, en répondant à des prompts variés [36].

- **Pré-entraînement :**

  GPT-2 est pré-entraîné sur un corpus de textes massifs pris dans une base de données massive qui sont extraits d'Internet, le corpus WebText, qui comprend environ 8 millions de documents [36]. L'objectif de cette phase de pré-entraînement est de minimiser la perte de la probabilité croisée (cross-entropy loss) de la prédiction du modèle et du mot réel suivant dans chaque séquence de texte, ce qui amène donc le modèle à apprendre la structure grammaticale, les relations sémantiques, le style et les faits du monde réel de manière non supervisée [37].

- **Inférence :**

  Le modèle générateur réclame un prompt, qui est l'amorce du texte à générer, puis il produit un mot à la fois, à chaque itération, à l'aide du contexte établi à partir de tous les prédits générés jusqu'ici, selon un processus de génération auto-régressive. L'architecture de réseau de neurones de type transformateur dans laquelle GPT-2 est formé permet d'obtenir aussi de longues séquences de texte qui sont alors cohérentes et contextuellement adaptées [36, 37].

La figure 2.18 représente le fonctionnement de GPT-2 dans notre système RAG

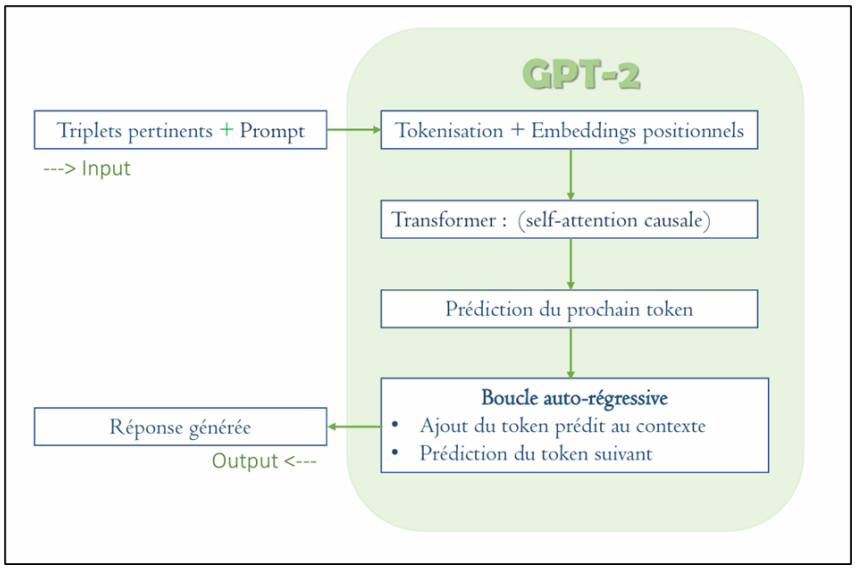


FIGURE 2.18 – Fonctionnement du Modèle GPT 2.

GPT-2 prend en entrée des triplets issu du KG(entités et relation) transformé en prompt textuel. Le prompt est tokenisé avec ajout d'embeddings positionnels pour préserver la structure de la phrase. Ces tokens passent ensuite dans un Transformer à attention causale, qui préserve l'ordre temporel des mots lors de la génération. Le modèle prédit ce qui pourrait être le prochain token en se fondant uniquement sur les tokens précédemment générés, et il recommence en suivant un processus itératif auto-régressif, en intégrant chaque nouveau token dans le contexte à travers une passerelle énonciative efficace et informative, afin de le prédire immédiatement et successivement, jusqu'à générer un token de fin.

2.7.3 Modèle Flan-T5-Base

Text-to-Text Transfer Transformer (T5) correspond à une version fine-tunée et optimisée de l'architecture T5 développée par Google Research. Présenté par Raffel et al. (2020), le T5 se distingue par sa volonté d'unifier toutes les tâches de traitement du langage naturel au sein d'un texte unifié d'entrée et de sortie sous forme de texte : "text-to-text" [38]. En permettant de réunir de façon systématique une grande quantité de tâches (classification, traduction, résumé, question-réponse, etc.), la représentation des tâches sous forme textuelle permet de tirer parti d'apprentissages réalisés sur une tâche pour les transférer vers une autre.

Le modèle T5 représente une avancée et un perfectionnement du modèle T5 d'origine. Au cœur de T5 l'idée consiste à permettre une meilleure prise en charge du suivi d'instructions explicites. Le T5 est fine-tuné (affiné) sur une base massive contenant des milliers de tâches et instructions naturelles. Cet entraînement supplémentaire a pour but d'améliorer la capacité à généraliser en cas de tâches inédites (i.e. non vues pendant l'entraînement), notamment en situation de zero-shot learning (c'est à dire résoudre une tâche sans exemple spécifique) [39].

- **Pré-entraînement :**

Dans un premier temps, le modèle T5 a été pré-entrainé sur un corpus colossal via une tâche de masking nommée ”Span corruption”, où plusieurs morceaux (spans) du texte sont masqués et le modèle doit les reconstruire afin d'apprendre une représentation riche du langage [39].

- **Phase de Fine-Tuning Instruction (Flan) :**

  Dans un second temps, T5 est fine-tuné sur un grand nombre de tâches différentes (traduction, classification, génération de texte, réponse à des questions, etc.) avec des instructions données au modèle comme prompts natural, pour chaque tâche. Cela consiste à proposer un texte qui indique au modèle ce qu'il doit faire, par exemple « traduire cette phrase en français » ou « résumer ce paragraphe » [39].

- **Inférence (génération) :**

  Enfin, lors de l'inférence, T5 reçoit une instruction sous forme de texte et génère une réponse elle aussi sous forme de texte. Cette approche text-to-text permet de simplifier de façon extrêmement importante l'usage du modèle, qui peut être utilisé pour une large gamme de tâches simplement en adaptant le prompt d'entrée [39].

La figure 2.19 représente le fonctionnement global du modèleT5 dans notre système RAG

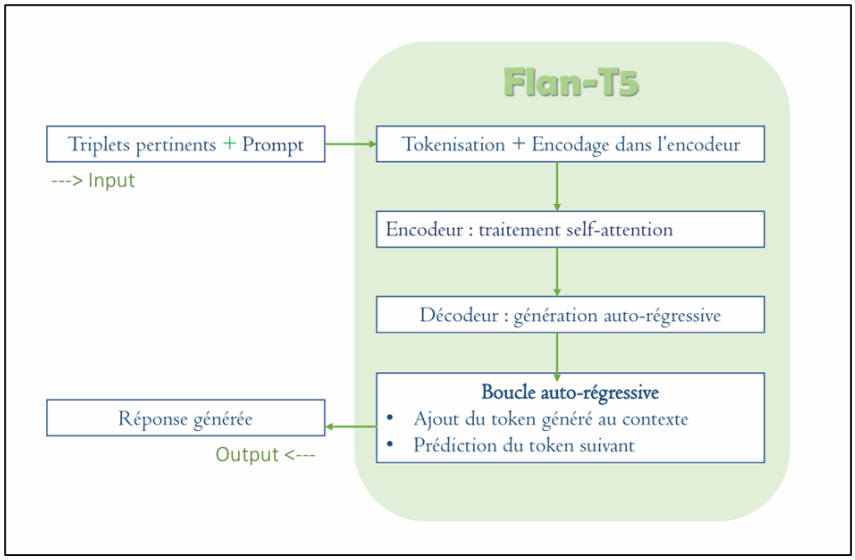


FIGURE 2.19 – Fonctionnement du Modèle T5.

T5 reçoit en entrée un triplet transformé en prompt textuel servant d'instruction. Ce prompt est tokenisé et encodé par l'encodeur du Transformer qui analyse l'ensemble du texte d'entrée pour en extraire une représentation contextuelle riche. Le décodeur lui génère la réponse de manière auto-régressive en produisant un token à la fois tout en s'appuyant sur la représentation encodée via un mécanisme d'attention croisée. À chaque étape le token généré est ajouté au contexte pour guider la prédiction du token suivant.

2.7.4 Comparaison des Modèles

Pour éclairer les particularités et les rôles respectifs des divers modèles de génération utilisés dans notre cadre RAG, le tableau 2.8 propose une comparaison globale entre BioGPT, GPT-2 et T5 sur la base de différents critères clés.

Caractéristiques	**BioGPT**	**GPT-2**	**Flan-T5-Base**
Domaine de spécialisation	Biomédical	Généraliste (tous domaines)	Généraliste (multitâche avec suivi d'instructions)
Architecture	Transformer	Transformer	Transformer (Text-to-Text)
Corpus d'entraînement	Résumés PubMed biomédicaux	Corpus WebText ($\tilde{4}$0 Go de données Internet)	C4 Corpus (Colossal Clean Crawled Corpus)
Type de tâche	Génération biomédicale cohérente à partir de prompts	Génération générale conditionnée par contexte	Génération conditionnée par instructions (text-to-text)
Phase de pré-entraînement	Modélisation de texte biomédical	Prédiction du mot suivant	Reconstruction de spans masqués
Phase de fine-tuning	Pas de fine-tuning dans notre cas (utilisation en inférence directe)	Non applicable (modèle général)	Fine-tuning sur tâches d'instructions multiples (Flan)
Mode d'inférence	Génération à partir d'une source contextuelle biomédicale + prompt	Génération auto-régressive à partir d'un prompt	Génération basée sur une instruction textuelle

TABLE 2.8 – Comparaison des Modèles de Génération.

Nous présentons dans le tableau2.9 un exemple de requête ainsi que les réponses générées respectivement par les trois modèles utilisés : **BioGPT, GPT-2** et **T5**.

Requête	**What pathological process is investigated in the radioiron oral absorption study ?**
Modèle	**Réponse générée**
BioGPT	Radioiron oral absorption study procedure pathological process ferric ion.Based on this knowledge : The use of radioiron oral absorption studies method can give us accurate information concerning absorption of iron in different pathological processes, with or without gastrointestinal lesions.
GPT-2	Radioiron oral absorption study procedure pathological process ferric ion.Based on this knowledge : The use of magnetic resonance imaging (MRI) in the treatment of oral cancer is safe and effective... (hors sujet).
FLAN-T5	Radioiron oral absorption study procedure pathological process ferric ion.

TABLE 2.9 – Comparaison des Réponses des Modèles.

En analysons les réponses il est évident que **BioGPT** se distingue par la pertinence et la qualité de sa génération. Il saisit parfaitement le contexte médical et fournit une réponse logique, expliquant que l'étude de l'absorption du fer par voie orale permet de mieux comprendre certains processus pathologiques. Sa réponse répond donc directement à la question. En revanche GPT-2 s'écarte totalement du sujet en évoquant l'IRM et le cancer buccal ce qui n'a rien à voir avec la question, rendant ainsi sa réponse totalement hors de propos. T5 il se contente de répéter les termes du triplet sans fournir d'explication ou de développement

Après cette analyse, nous allons presenter une evaluation quantitative et qualitative de notre système pour effectuer une évaluation pour déterminer quel modèle serait le plus adapté.

2.8 Conclusion

Ce chapitre nous a permis d'étudier en profondeur le fonctionnement de notre système RAG et donc plus spécifiquement des modèles génératifs intégrés. Tout d'abord, nous avons soigneusement sélectionné un dataset pertinent, puis nous avons expérimenté plusieurs méthodes pour calculer les embeddings, apparié les entités du KG et enfin ossé ces étapes aux modèles de génération pour produire des réponses médicales de qualité.

Grâce à l'intégration de modèles comme BioGPT, GPT-2 et Flan-T5-Base, notre système réagit aux informations non seulement en les restituant mais les convertit en réponses claires, adéquates et intelligibles pour les utilisateurs avec des spécificités qu'implique le domaine médical. Chaque modèle apporte ici ses atouts, qu'il s'agisse de précision, de fluidité ou de spécialisation. Enfin, l'évaluation que nous avons réalisée montre que cette approche est prometteuse et ouvre de belles perspectives pour l'avenir. Ce travail pose déjà des bases solides pour construire des

systèmes d'assistance médicale plus intelligents, plus fiables et plus utiles pour les professionnels comme pour les patients.

Dans le chapitre suivant, nous allons conclure le mémoire en justifiant nos choix de langage et outils d'implémentation, les résultats détailles de nos exiperimentations ainsi que leur analyse.

Chapitre III : Tests et Validation de la Solution

3.1 Introduction

Dans ce chapitre nous présentons l'ensemble des outils, technologies et bibliothèques nécessaires à la mise en œuvre de notre système.Nous expliquons les choix techniques en termes de solidité et d' efficacité et nous présentons les résultats détaillés, pour finir par une présentation générale de la plateforme ainsi que de ses principales fonctionnalités.

3.2 Environnement de Developement

Nous allons maintenant décrire successivement l'environnement matériel, puis l'environnement logiciel utilisés dans le cadre de ce projet

3.2.1 Environnement Matériel

Le tableau 3.10 ci-dessous présente les caractéristiques des deux ordinateurs portables que nous avons utilisés :

Composant	**ASUS**	**FUJITSU**
RAM	8 Go DDR4	8 Go DDR4
SSD	512 Go	256 Go
CPU	Intel Core i7-1065G7	Intel Core i5-8350U
Système d'exploitation	Windows 11	Windows 10

TABLE 3.10 – Environnement Matériel.

3.2.2 Environnent Logiciel

Dans cette partie, nous allons présenter l'environnement technique utilisé pour réaliser ce projet, en précisant le langage de programmation, les outils et les bibliothèques que nous avons

employés.

Langage de l'application

Pour réaliser ce projet, nous avons utilisé Python , Flask et Neo4j [15] . Le tableau ci-dessous 3.11 en donne une description pour chacun.

Langage	Définition
Python 	Python est un langage de programmation de haut niveau interprété et orienté objet. Il se définit par sa simplicité syntaxique et la richesse de son écosystème . Il est largement utilisé pour les applications scientifiques, industrielles et web en raison de sa rapidité de développement et de sa lisibilité. Au sein de ce projet, le choix s'est tourné vers Python car c'est un langage généraliste reconnu pour ses aptitudes dans des domaines de pointe tels que l'IA, l'analyse de données et le web. Pour assurer la communication harmonieuse entre les modules du projet, il s'est révélé être un choix pertinent.
Flask 	Flask est un framework open-source de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de la programmation Python, associé à un système de templates. Flask a eu un rôle central dans le relais entre la computation Python et un utilisateur distant. Voilà pourquoi nous avons utilisé cet outil de création d'interface web qui permet de consulter les résultats, d'émettre des requêtes et d'interagir avec le système.
Neo4j 	Le Système de Gestion de Base de Données Neo4j est un type de base de données orientée graphe, c'est-à-dire qu'il permet de stocker et de gérer des données très connectées sous forme de nœuds (entités) et de relations (liens). Neo4j permet de représenter des relations complexes entre différentes entités et de répondre aisément à des requêtes exploratoires dans l'analyse de réseaux de données. Il est utilisé dans des secteurs comme la santé, la finance, les réseaux sociaux et la recommandation. Dans notre cas, Neo4j a permis de modéliser la base de connaissances à visée médicale, permettant ainsi l'exploration de liens entre maladies, symptômes et traitements.

TABLE 3.11 – Technologies Utilisés dans le Projet.

Outils

En plus d'autres outils ont été essentiels pour le développement et le bon déroulement du projet.ils sont résumés dans le tableau 3.12 ci-dessous :

15. Neo4j https://neo4j.com/ (Consulté : 03/2025)

Outil	Description
Visual Studio Code 	Visual Studio Code [16] est un éditeur de code source léger et flexible proposé par Microsoft. Il permet de développer dans de nombreux langages, et est apprécié pour sa rapidité, sa lisibilité et ses nombreuses extensions telles que la coloration syntaxique, l'autocomplétion ou encore le débogage.
Jupyter Notebook 	Jupyter Notebook [17] est un environnement interactif permettant d'écrire et d'exécuter du code par blocs, appelés cellules. Très utile pour le prototypage, l'analyse de données et la visualisation, il permet d'intégrer dans un même document du code, des graphiques et du texte explicatif pour mieux documenter et présenter les résultats.
Google Collab 	Google Colaboratory [18] est un outil gratuit permettant de prototyper des modèles de Machine Learning avec des GPU et TPU, directement dans un navigateur web. Il s'agit d'un service de notebooks Jupyter hébergé, facile à utiliser sans configuration requise. Les machines virtuelles de Colab offrent des ressources informatiques puissantes mais sont limitées par des quotas et des interruptions en cas d'inactivité. Pour plus de ressources et une utilisation prolongée, des versions payantes comme Colab Pro et Colab Pro+ sont disponibles.
Overleaf 	Overleaf [19] est une plateforme en ligne collaborative spécialisée dans la création, l'édition et la publication de documents LaTeX. Principalement dédiée aux travaux académiques et scientifiques, elle permet une collaboration en temps réel et offre un aperçu instantané du document final.

TABLE 3.12 – Outils Utilisés dans le Projet.

Les bibliothèques utilisées

re : re [20] a été utilisé pour faire de la recherche de motifs dans du texte, semblable au travail du détective à la recherche de mots ou de schémas précis. nous l'avons utilisé pour nettoyer et transformer les textes en repérant certaines structures spécifiques.

Json : ou JavaScript Object Notation [21] est un système de notation lié au langage javascript (il fait partie de la librairie standard de python). Le module json de python permet d'utiliser ce système de notation avec python, pour notamment "sérialiser" des objets python de type dict (dictionnaire) ou list

FAISS : FAISS [22] développé par Meta, est vraiment un outil efficace pour faire rapidement de la recherche sur les ressemblances entre différents éléments d'un grand ensemble de données. Nous avons utilisé pour retrouver les vecteurs de textes susceptibles de correspondre aux réponses les plus proches d'une question posée.

Ast : Le module ast [23] nous a permis de transformer en objets Python réels les chaînes de caractères ressemblant à du code Python, ce qui était utile lors de l'analyse de certaines structures

20. re (https ://docs.python.org/3/library/re.html Consulté : 04/2025)
21. JSON https ://docs.python.org/3/library/json.html (Consulté : 04/2025)
22. FAISS https ://faiss.ai/ (Consulté : 04/2025)
23. Ast https ://docs.python.org/3/library/ast.html (Consulté : 04/2025)

de données dynamiques.

Torch : PyTorch [24] est un framework d'une puissance incroyable, dédié à l'intelligence artificielle. Cette application nous a permis d'accéder à des modèles de modèles de deep learning comme BioGPT, outil de traitement intelligent du langage biomédical.

Transformers : La fameuse bibliothèque transformers [25] développée par Hugging Face met à disposition de la communauté des modèles pré-entraînés capables de comprendre et de générer du texte. Nous a permis de tirer parti de modèle d'origine spécialisée (par example BioGPT dans notre système afin d'obtenir des réponses de qualité.

scikit-learn : Avec les fonctions de Scikit-learn [26] , nous avons pu mesurer la similitude entre deux textes ou deux vecteurs. La fonction (cosine similarity) nous'a permis ainsi de comparer des phrases et la comparaison des phrases les plus pertinentes à une question.La fonction normalize du module Scikit-learn nous a permis d'homogénéiser nos données afin de conduire des comparaisons sur un même pied d'égalité. Sans elle, certaines valeurs auraient eu trop de poids.

NumPy : NumPy [27] ressemble à une super calculatrice pour Python. Grâce à elle, les calculs mathématiques sont rapides et efficaces, en particulier lorsqu'il s'agit de manipuler des tableaux ou des matrices, ce qui était utile pour traiter les données numériques au cœur du système.

3.3 Scénarios d'Expérimentation et d'Evaluation du Système

Dans cette partie nous allons évaluer le système RAG à travers les modèles d'embedding, les modèles de générations et l'interface de requêtage de la plateforme.

3.3.1 Evaluation du Modèles de Génération des Embeddings

Dans le but de mesurer les performances de modèles de génération des embeddings , un ensemble de paires question-réponse a été généré automatiquement à partir du KG .

Pour chaque type de relation contenue dans le KG un modèle de question en a été défini. Par exemple :

- Pour la relation : "CAUSATIVE_AGENT" , il s'agit de la question "What causes [concept] ?".
- Pour la relation "HAS_MANIFESTATION", cela donne "What manifests from [concept] ?".

Les questions ont été construites à partir des triplets extraits (concept source, relation, concept cible), en y insérant le concept source dans le modèle, et considérant le concept cible comme réponse attendue.

24. PyTorch https ://pytorch.org/ (Consulté : 04/2025)
25. Transformers https ://huggingface.co/docs/transformers/ (Consulté : 04/2025)
26. Scikit-learn https ://scikit-learn.org (Consulté : 04/2025)
27. NumPy https ://numpy.org/ (Consulté : 04/2025)

Un échantillon représentatif de ces paires a ensuite été retenu pour constituer un jeu de données d'évaluation qui a été utilisé pour évaluer la capacité des modèles de génération des embeddings à répondre correctement à des questions générées .

Nous avons utilisé plusieurs métriques standards dans le domaine du NLP : le F1-score, la métrique ROUGE-L pour la qualité des résumés générés, BERTScore pour l'évaluation du contenu sémantique

F1-score

Le F1-score est une mesure synthétique qui combine la précision (proportion de prédictions correctes parmi les prédictions positives) et le rappel (proportion de cas positifs correctement identifiés). Il s'agit de leur moyenne harmonique, ce qui permet de tenir compte des déséquilibres entre ces deux mesures. Cette métrique est particulièrement adaptée aux tâches où il est essentiel de minimiser à la fois les faux positifs et les faux négatifs. Elle se calcule comme suit :

$$F1 = 2 \times \frac{P \times R}{P + R}$$

Où :

— P : Précision = $\frac{VP}{VP+FP}$

— R : Rappel = $\frac{VP}{VP+FN}$

— VP : Vrais positifs

— FP : Faux positifs

— FN : Faux négatifs

Exact Match

La métrique Exact Match mesure le pourcentage de prédictions qui correspondent exactement à la séquence de référence. Elle est particulièrement exigeante car elle ne tolère aucune erreur de mot, de ponctuation ou d'ordre. C'est une métrique stricte mais efficace dans les tâches telles que la réponse à une question ou la traduction automatique, où la fidélité textuelle est primordiale.

$$EM = \frac{\text{Nombre de prédictions exactes}}{\text{Nombre total d'exemples}}$$

Où :

— Nombre de prédictions exactes : nombre de sorties identiques à la référence

— Nombre total d'exemples : taille de l'ensemble de test

ROUGE-L

La métrique ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation) s'appuie sur la plus long sous-séquence commune (LCS) entre la séquence générée et la référence. ROUGE-L

accorde une importance particulière à la structure de la phrase et à la conservation de l'ordre des mots.

$$ROUGE\text{-}L = \frac{LCS}{\text{Longueur de la séquence de référence}}$$

Où :

— LCS : Longest Common Subsequence (plus longue sous-séquence commune)

— Longueur de la séquence de référence : nombre de mots dans la phrase de référence

BERTScore

Contrairement aux métriques traditionnelles basées sur des correspondances exactes de mots, le BERTScore utilise les représentations contextuelles issues du modèle BERT pour comparer la proximité sémantique entre la sortie générée et la référence. Il repose sur la similarité cosinus entre les vecteurs d'embedding des mots.

$$\text{BERTScore} = \frac{1}{n}\sum_{i=1}^{n} \max_j \cos\left(e_i^{\text{pred}}, e_j^{\text{ref}}\right)$$

Où :

— e_i^{pred} : embedding du i$^{\text{ème}}$ mot de la prédiction

— e_j^{ref} : embedding du j$^{\text{ème}}$ mot de la référence

— n : nombre total de mots dans la phrase générée

Cosine Similarity

La similarité cosinus mesure l'angle entre deux vecteurs dans un espace vectoriel. Elle permet d'évaluer la proximité sémantique entre deux phrases ou documents.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \times \|B\|}$$

Où :

— A, B : vecteurs d'embedding à comparer

— $A \cdot B$: produit scalaire entre A et B

— $\|A\|$, $\|B\|$: norme euclidienne des vecteurs

Le tableau suivant 3.13 indique les résultats obtenus :

Modèle	F1-score	Exact Match	ROUGE-L	BERTScore	Cosine Similarity
BioBERT	**0.96**	**0.95**	**0.96**	**0.98**	**0.97**
SciSpaCy	0.91	0.89	0.91	0.95	0.92
Word2vec	0.82	0.80	0.82	0.89	0.84

TABLE 3.13 – Évaluation des Modèles de Génération des Embeddings.

Analyse des Résultats Obtenus

L'analyse comparative des performances des modèles d'embedding BioBERT, SciSpaCy et Word2vec met en évidence des écarts significatifs selon les différentes métriques d'évaluation utilisées. Le modèle BioBERT se démarque par des résultats nettement supérieurs, avec un F1-score de 0.96, une mesure d'Exact Match de 0.95, un score ROUGE-L de 0.96, un BERTScore de 0.98 et une similarité cosinus de 0.97. Ces performances traduisent la capacité de BioBERT à générer des représentations contextuelles riches et précises, en particulier dans le domaine biomédical pour lequel il a été pré-entraîné. SciSpaCy présente des résultats intermédiaires (F1-score de 0.91, BERTScore de 0.95), indiquant une bonne efficacité tout en demeurant moins performant que BioBERT sur l'ensemble des indicateurs. En revanche, Word2vec, qui repose sur des représentations statiques des mots, obtient des scores sensiblement plus faibles (F1-score de 0.82, Exact Match de 0.80), illustrant ses limites dans la capture des relations sémantiques complexes.

En définitive, cette évaluation met en évidence l'avantage significatif des modèles contextuels de type Transformer, tels que BioBERT, pour les tâches de traitement du langage naturel en contexte médical. Le choix du modèle doit ainsi être effectué en fonction des exigences de la tâche cible, du domaine d'application et des ressources computationnelles disponibles.

3.3.2 Évaluation des Modèles de Génération

Pour évaluer les différents modèles de génération, nous avons calculé les mesures de performances, notamment le rappel, la précision, le F1-Score et la Perplexité.

Nous avons travaillé sur un ensemble de références extraites à partir de 20 % du KG contenant environ 8 000 triplets, que nous avons ensuite utilisées comme données d'entrée pour le modèle de génération en vue de l'évaluation

Métrique BERTScore

Le fonctionnement de BERTScore repose sur plusieurs étapes que nous détaillons ci-dessous :

1. **Représentation sémantique des mots** BERTScore mesure la qualité d'un texte généré dans la mesure où il évalue la similarité sémantique entre chaque mot d'un texte de prédiction et d'un texte de référence. Pour cela, il recourt à des représentations vectorielles denses conçues par des modèles de langage préentraînés, comme BERT en vertu desquels chaque mot ou sous-mot est amené à correspondre à un vecteur contextualisé, le sens du mot dépendant à la fois de sa position syntaxique et de son voisinage au sein d'une phrase. Cette approche permet de contourner certaines des limites que présentent les métriques traditionnelles reposant sur une correspondance stricte des mots [5].

2. **Encodage et calcul de similarité** Pour comparer deux textes, BERTScore encode en premier lieu séparément le texte généré (candidat) et le texte de référence, à l'aide d'un modèle tel que BERT ensuite il calcule la similarité cosinus entre les vecteurs d'un mot du candidat et les vecteurs de chacun des mots de référence. Ce mécanisme de couplage permet de tenter de mieux établir la correspondance sémantique entre le mot généré et ceux de la référence [5].
3. **Calcul des métriques** Dans l'étape d'après, après avoir mesuré les similarités pour chaque mot BERTScore extrait le mot le plus similaire dans la référence. A partir de ces appariements on en déduit :
    - **La précision :** pour savoir dans quelle mesure chaque mot généré est bien un mot de la référence
    - **Le rappel :** pour vérifier si tous les éléments attendus de la référence sont présents dans la prédiction
    - **Le score F1** : comme la moyenne harmonique de la précision et du rappel utile pour avoir une appréciation globale de la qualité du texte généré [5].

La figure 3.20 illustre le processus de calcul de BERTScore :

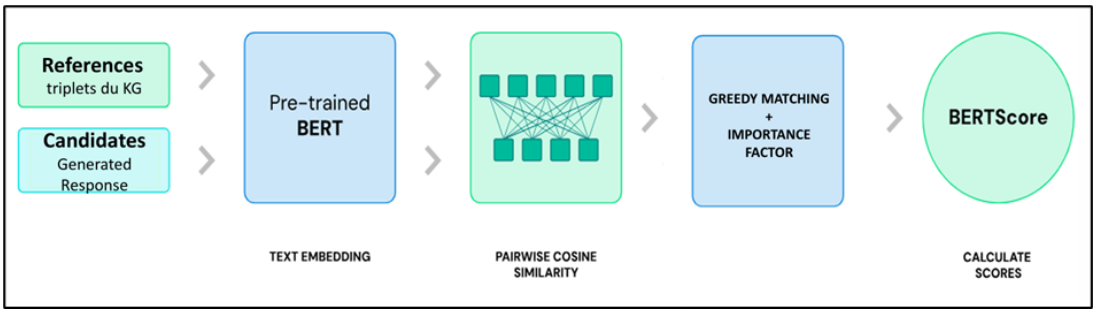


FIGURE 3.20 – Processus de Calcul de BERTScore [5].

Cette procédure d'alignement sémantique fait que BERTScore constitue un bon score pour évaluer un texte paraphrasé reformulé ou encore exprimé avec des synonymes car il valorise davantage la similarité de sens des mots au lieu de leur stricte égalité de surface. [5]

Perplexité

La perplexité est une métrique utilisée pour évaluer la capacité prédictive d'un modèle de langage. Elle mesure à quel point le modèle est incertain face à un texte donné : une faible perplexité indique que le modèle est confiant dans ses prédictions, tandis qu'une perplexité élevée signale plus d'incertitude. En termes simples la perplexité représente la moyenne du degré de "surprise" du modèle face aux mots générés. C'est une métrique particulièrement utilisée pour juger de la qualité des modèles génératifs modernes en traitement du langage naturel . [40]

Le calcul de la perplexité s'effectue en quelques étapes. Tout d'abord, nous effectuons une tokenisation des phrases de générées et référence c'est-à-dire que nous procédons à une division en mots ou sous-mots des phrases considérées. Ensuite, à partir de la phrase de référence nous effectuons une distribution de probabilité des mots en comptant le nombre d'apparitions de chaque mot ; la probabilité de ce mot est ensuite normalisée par le nombre total de mots de la phrase. Ensuite pour évaluer la perplexité, chaque mot génère une probabilité correspondante d'être présent dans la phrase de référence ainsi la formule de perplexité est basée sur la somme des logarithmes des inverses des probabilités observées au sein d'un même corpus et nous élevons cette dernière à la valeur exponentielle. Pour obtenir la perplexité. Nous pouvons donc conclure qu'il s'agit d'un modèle de performances limitées car plus elle est petite, plus la phrase générée est en accord avec la phrase de référence.

$$P = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(W_i \mid w_1, w_2, \ldots, W_{i-1})\right)$$

Où :

- **N :** est le nombre total de mots dans le texte.
- **Wi :** représente le i ème mot du texte.
- **P(Wi) :** est la probabilité prédite par le modèle pour le i-ème mot donné les mots précédents

Le tableau 3.14 suivant indique les résultats obtenus :

Modèle	Précision	Rappel	F1-Score	Perplexité
BioGPT	**0.8688**	**0.9549**	**0.9097**	**8.02**
GPT-2	0.8366	0.9448	0.8873	8.45
Flan-T5	0.7548	0.7905	0.7716	16.22

TABLE 3.14 – Evaluation des Modèles de Génération.

Analyse des Résultats Obtenus

D'après les résultats obtenus BERTScore, BioGPT obtient les meilleurs résultats avec une précision de 0.868 un rappel de 0.9549 et un F1-score de 0.9097. Ces résultats indiquent que BioGPT génère des textes qui sont plus proches des références, avec une couverture et un alignement sémantique bien plus élevés. On retrouve en deuxième position GPT-2 bien que légèrement moins performant (précision : 0.8366, rappel : 0.9448, F1-score : 0.8873). T5 affiche en revanche des résultats beaucoup plus faibles (précision : 0.7548, rappel : 0.7905, F1-score : 0.7716) qui sont quatre en termes de précision et de cohérence

Concernant la perplexité BioGPT affiche les meilleures valeurs (perplexité : 8.02) des textes générés plus naturels et mieux adaptés aux modèles de langage. En deuxième position bien que performant GPT-2 présente une perplexité de 8.45 légèrement moins fluide. T5 , qui présente la perplexité la plus élevée (16.22) fournit des textes plus imprévisibles et moins cohérents que ceux

des deux autres modèles. Toutefois, les trois perplexites reste plus au moins correcte sachant que dan le domaine NLP, une bonne perplexité doit être inférieure ou égale à 40.

Pour conclure le modèle dont les résultats sont les plus satisfaisants est BioGPT dont on peut retenir les meilleurs scores en précision, rappel, F1-score et perplexité. En deuxième position se situe le modèle GPT-2, alors que T5 se cantonne finalement aux performances les plus mauvaises, notamment en ce qui concerne la fluidité et la cohérence des textes produits

3.3.3 Interface de Requêtage

La plateforme web que nous avons développée dispose de deux pages principales ayant pour rôle d'interagir avec l ' utilisateur et d 'afficher les résultats : la page index et la page détails. **Index de la page :** il s'agit de la page d' accueil dans laquelle l' utilisateur pose sa question à travers un formulaire . La figure 3.21 qui suit représente la page d'accueil pour la recherche d'information biomédicale :

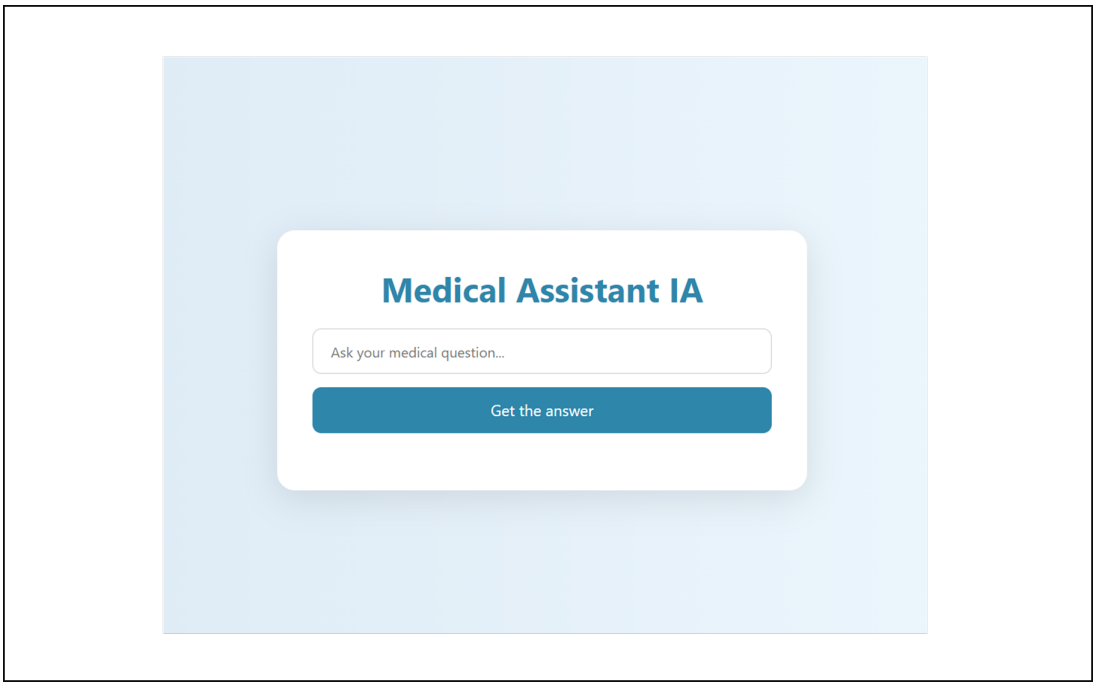


FIGURE 3.21 – Page d'Accueil pour la Recherche d'Information Biomédicale.

Une fois saisie, la plateforme :

1. Recherche le concept le plus pertinent en interrogeant les embeddings stockés dans Neo4j.
2. Identifie la relation la plus pertinente liée à ce concept .
3. Génère avec le modèle BioGPT une réponse enrichie.

La figure 3.22 représente la page du résultat de la recherche d'information :

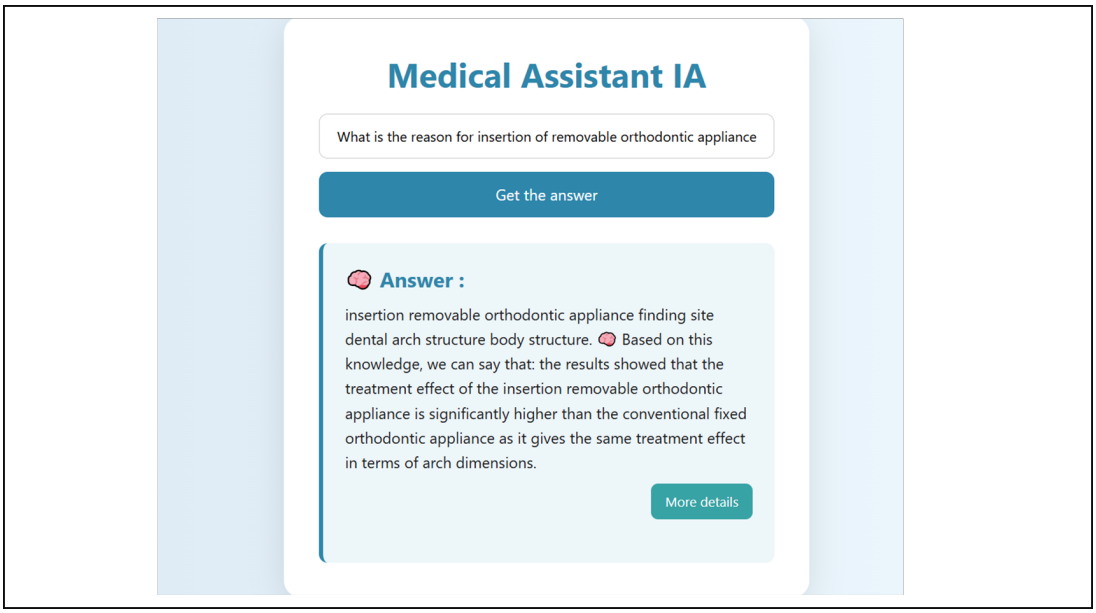


FIGURE 3.22 – Page du Résultat de la RI.

La réponse ainsi que les informations structurantes (concept, relation, cibles identifiées) sont affichées directement sur cette page. L' utilisateur a ensuite la possibilité de cliquer pour voir le résultat de l'appariement sémantique entre la requête de l'utilisateur .

Page détails : Cette page permet d'explorer de manière approfondie les résultats générés par la plateforme en réponse à une question posée. Elle affiche la question initiale, le concept extrait, la relation sémantique identifiée, les cibles associées, ainsi que le score de similarité indiquant la pertinence du résultat. Un encadré présente également la réponse générée par le modèle BioGPT. . La figure 3.23 ci-dessous illustre l'interface de cette page

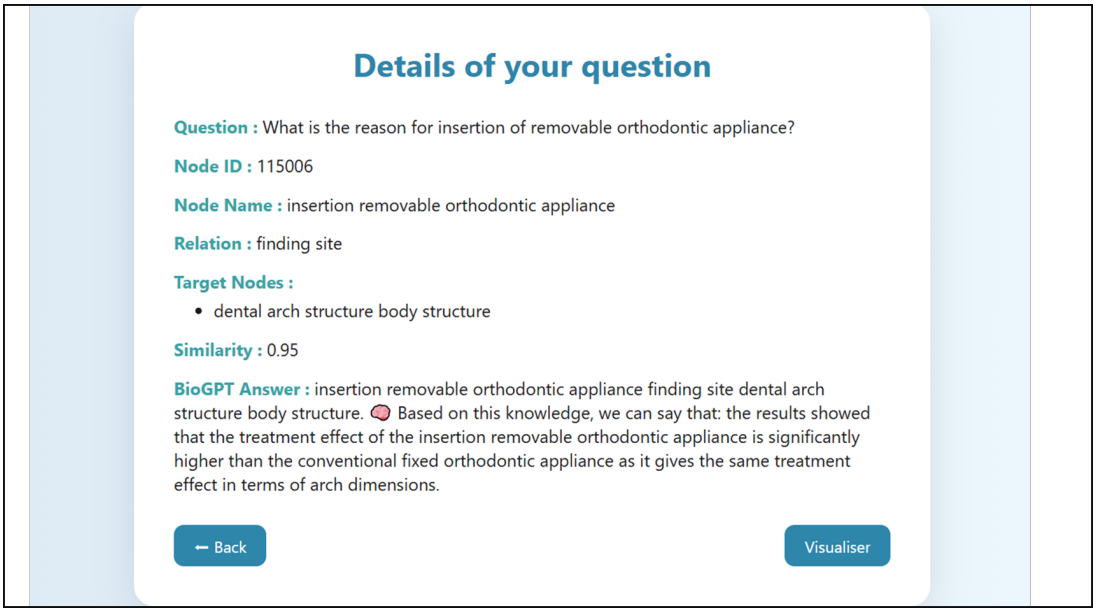


FIGURE 3.23 – Page de Détails des Résultats de la RI.

De plus, un bouton permet d'accéder à une visualisation interactive sous forme de graphe, illustrant de façon intuitive les liens entre les concepts, ce qui facilite l'interprétation des relations sémantiques dans le graphe de connaissances. La figure ci-dessous 3.24 représente la visualisation interactive associée.

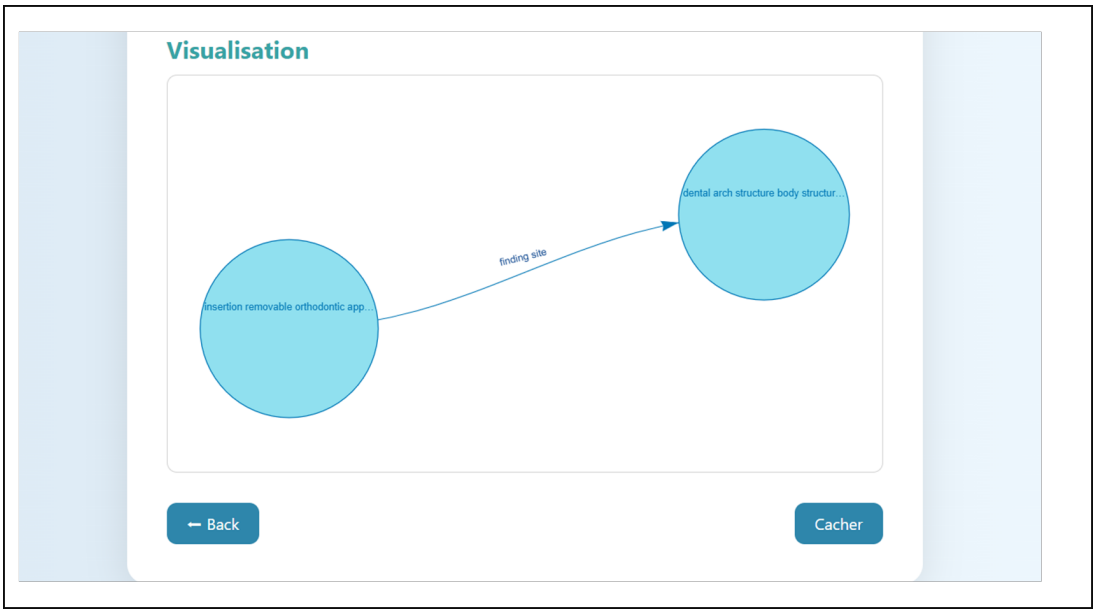


FIGURE 3.24 – Page de la Visualisation Interactive des Resultats.

3.4 Conclusion

Dans ce chapitre, nous avons testé et validé notre solution dans un environnement bien défini. Nous avons commencé par évaluer les différents modèles d'embeddings sémantiques, puis les modèles de génération de texte, pour enfin tester l'interface de requêtage du système.

Les résultats montrent clairement que l'intégration d'un graphe de connaissances dans le pipeline RAG améliore la qualité et la pertinence des réponses générées. Les modèles testés ont été comparés de manière équitable, et ceux qui exploitent les connaissances structurées du graphe ont montré des performances supérieures.

De plus, l'interface développée permet une interaction simple et efficace, ce qui rend le système plus accessible pour une utilisation réelle dans un contexte médical. En résumé, ces expérimentations confirment la construction un assistant médical plus précis, cohérent et intelligent

Conclusion Générale

1 Conclusion

Face à une véritable explosion des données médicales et à leur complexité croissante, ce travail s'inscrit dans le projet de concevoir un système intelligent apte à faciliter l'accès à une information médicale pertinente, contextualisée et rigoureuse. À cet égard, nous avons proposé un assistant médical fondé sur l'intégration d'un graphe de connaissances dans un cadre de RAG.

Ce dernier a permis de bien exploiter les relations existant entre entités médicales grâce d'une part à l'extraction des informations pertinentes et les modèles de génération de texte en mesure de restituer des réponses naturelles, correctes et et adaptées au contexte des requêtes médicales. La mise en œuvre du pipeline RAG permet ainsi de bien articuler l'un avec l'autre deux aspects cruciaux : d'une part, la richesse sémantique de KG valide, d'autre part la flexibilité permise par l'adaptation de LLMs.

Dans les travaux citer dans le premier chapitre, beaucoup de recherches se sont consntrées sur l'utilisation des kg pour améliorer l'accès à l'information médicale. Certains projets se sont intéressés à la génération de réponses plus précises .Dans notre travail, nous avons combiner ces avancées dans un seul système.

De ce fait, le système conçu répond promptement et avec pertinence aux problèmes rencontrés : il permet une meilleure récupération d'informations médicales fiables, il facilite l'extraction de réponses pertinentes à partir d'un graphe structuré et il améliore l'aide à la décision des cliniciens par l'automatisation de l'assistance. Cet projet constitue une première étape vers des systèmes d'aide à la décision de plus haut niveau utilisant et les caractéristiques des graphes sémantiques et les avancées de l'IA générative en faveur des professionnels de santé et des patients.

2 Perspectives Ce projet représente une première étape dans une recherche de mise au point d'un assistant médical intelligent pertinent. Pour aller plus loin, plusieurs voies peuvent être explorées pour rendre le système plus complet, plus personnel, et plus adapté à un usage concret sur le terrain

— Enrichir le graphe avec des sources plus variées et multilingues :Une bonne possibilité consisterait à intégrer d'autres bases de connaissances médicales comme PubMed ou

DrugBank, mais aussi d'inclure des ressources multilingues. Cela permettrait d'avoir plus de cas médicaux en vue et donc de toucher un public plus large.

— Adapter les réponses au profil de l'utilisateur : Tout utilisateur n'a pas les mêmes besoins : un patient, un généraliste ou un spécialiste n'ont pas besoin du même niveau d'information. En y ajoutant une personnalisation tenant compte de son profil (précédentes consultations, récurrence des réponses, etc.) permettrait d'apporter des réponses encore plus rigoureusement pertinentes.

— Mettre à jour le graphe en temps réel : La médecine évolue très vite. Pour garantir des réponses toujours à jour, il serait d'un grand intérêt de rendre ce graphe évolutif, en l'alimentant automatiquement à partir de nouvelles publications scientifiques ou de données récentes, afin que tout soit au plus proche de la réalité médicale actuelle.

— Tester le système dans le milieu professionnel de la santé : constitue réellement un élément clé. Tester l'utilité de l'assistant en contexte clinique avec des professionnels de santé permet de comprendre au mieux les besoins concrets, mais aussi les limites du système tout en permettant de consolider sa pertinence sur la base des réactions de terrain dans l'optique d'en améliorer le fonctionnement.

Bibliographie

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2021.

[2] Sofiane Kartobi and Mohamed Riad Ould Abdallah. Développement d'un assistant intelligent basé sur l'ia générative : application au sein de kpmg deal advisory, 2024.

[3] Yuhua Li, Zuhair A. Bandar, and David McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(4) :871–882, 2003.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

[5] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore : Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*, 2020.

[6] Junde Wu, Jiayuan Zhu, and Yunli Qi. Medical graph rag : Towards safe medical large language model via graph retrieval-augmented generation. *arXiv preprint arXiv :2408.04187*, 2024. Disponible en ligne : https ://arxiv.org/abs/2408.04187.

[7] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models : Techniques and applications. 2024.

[8] Sihem Boutebal and Samah Belbaki. Développement d'un chatbot bancaire intelligent : Comparaison de l'application de la dpo et du rag pour l'amélioration de l'interaction client chez kpmg, 2024.

[9] Sebastian Bruch. *Foundations of Vector Retrieval*. Monograph, 2024. Covers algorithms and data structures for large-scale vector search.

[10] Frank Zheng, Xin Li, and Angela Liu. *Vector Database : Unlocking the Power of High-Dimensional Search*. Zeta Alpha Publishing, 2024. Disponible en ligne : https ://www.amazon.com/Vector-Database-Unlocking-High-Dimensional-Applications/dp/B0DGR9KX4C.

[11] Eric Thompson and Mei Zhang. *Designing Intelligent Search Systems with Vector Databases*. TechNova Press, New York, NY, USA, 2023. Disponible en ligne :https ://www.vespa.ai/resources.

[12] Alexey Borisov. *Vector Databases Explained : Foundations and Applications*. AI Research Publishing, 2023. Disponible en ligne : https ://vectordb.ai/book.

[13] Mehdi Amor Ouahmed. Développement d'un assistant juridique artificiel. Mémoire de fin d'études, génie industriel – data science / intelligence artificielle, École Nationale Polytechnique d'Alger, Alger, Algérie, 2024. Consulté en juin 2025.

[14] Zeljko Kraljević, Anthony Shek, Daniel Bean, Rebecca Bendayan, James Teo, and Richard Dobson. Medgpt : Medical concept prediction from clinical narratives. *arXiv preprint arXiv :2107.03134*, 2021. https ://arxiv.org/abs/2107.03134.

[15] Yuxing Lu, Xukai Zhao, and Jinzhuo Wang. Clinicalrag : Enhancing clinical decision support through heterogeneous knowledge retrieval. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 64–68, Bangkok, Thailand, 2024.

[16] Yucheng Shi, Shaochen Xu, Tianze Yang, Zhengliang Liu, Tianming Liu, Quanzheng Li, Xiang Li, and Ninghao Liu. Mkrag : Medical knowledge retrieval augmented generation for medical question answering. *arXiv preprint arXiv :2309.16035*, 2023. Revised August 2024.

[17] Steven Song, Anirudh Subramanyam, Irene Madejski, and RobertL. Grossman. Lab-rag : Label boosted retrieval-augmented generation for radiology report generation. *arXiv preprint arXiv :2411.16523*, 2024. Published Nov 25, 2024.

[18] Christophe Du Mouza. Fusion d'entités dans des graphes de connaissances, 2024. Sujet de thèse proposé, consulté en juin 2025.

[19] S. Budhdeo, J. Zhang, Y. Abdulle, P. M. Agapow, D. G. J. McKechnie, M. Archer, V. Shah, E. Forte, A. Noori, M. Zitnik, H. Ashrafian, and N. Sharma. Scoping review of knowledge graph applications in biomedical and healthcare sciences. *medRxiv*, 2023.

[20] Hejie Cui, Jiaying Lu, Shiyu Wang, Ran Xu, Wenjing Ma, Shaojun Yu, Yue Yu, Xuan Kan, Chen Ling, Liang Zhao, Joyce C. Ho, Fei Wang, and Carl Yang. A survey on knowledge

graphs for healthcare : Resources, applications, and promises. *Journal of Biomedical Informatics*, 148 :104123, 2025. Version antérieure disponible sur arXiv :2306.04802.

[21] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *International Conference on Learning Representations (ICLR)*, 2024. Proceedings ID : arXiv :2310.04562.

[22] Shuai Zhang, Hongzhi Wang, Xin Wu, et al. Knowledge graphs meet multi-modal learning : A comprehensive survey. *arXiv*, 2024.

[23] Alexander Pelletier, Dylan Steinecke, Dibakar Sigdel, Irsyad Adam, John Harry Caufield, Vladimir Guevara-Gonzalez, Joseph Ramirez, Aarushi Verma, Kaitlyn Bali, Katherine Downs, Wei Wang, Alex Bui, and Peipei Ping. A knowledge graph approach to elucidate the role of organellar pathways in disease via biomedical reports. *Journal of Visualized Experiments*, 200 :e65084, 2023. Published Oct 13, 2023.

[24] Mark Hewitt, Srayanta Mukherjee, Anton Van Pamel, Leonid Zhukov, and Julien Delile. A framework to discover predictive (causal) rules from knowledge graph paths. *BMC Bioinformatics*, 24 :512, 2023.

[25] Cédric Piriou, Sylvie Desprès, Julien Nobécourt, Claire Le Roy, and Céline Irles. Graphe de connaissance et ontologie pour la représentation des données de la llc. In *Actes des Journées d'Informatique Médicale (INFOMED), PFIA 2023*, Strasbourg, France, 2023. Consulté en juin 2025.

[26] Abhinav Kimothi. *A Simple Guide to Retrieval Augmented Generation*. Manning Publications, 2025. 256 pages, eBook et imprimé ; couverture complète de RAG, y compris code et exemples pratiques à l'aide de LangChain.

[27] Gradient Flow. Graphrag : Advances in ai for medical question answering, n.d. Consulté le 02 décembre 2024, sur https ://gradientflow.com/graphrag-medgraphrag/.

[28] Julien Delile, Srayanta Mukherjee, Anton Van Pamel, and Leonid Zhukov. Graph-based retriever captures the long tail of biomedical knowledge. *arXiv preprint*, arXiv :2402.12352, 2024. Accepted at 1st Machine Learning for Life and Material Sciences Workshop, ICML 2024.

[29] Yinghao Zhu, Changyu Ren, Zixiang Wang, Xiaochen Zheng, Shiyun Xie, Junlan Feng, Xi Zhu, Zhoujun Li, Liantao Ma, and Chengwei Pan. Emerge : Integrating rag for improved multimodal ehr predictive modeling. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, page 3549–3559, October 2024.

[30] Gunnar Burger, Manuel and Rita Kuznetsova. Multi-modal graph learning over UMLS knowledge graphs. In *Proceedings of the 3rd Machine Learning for Health Symposium*, volume 225 of *Proceedings of Machine Learning Research*, pages 52–81. PMLR, December 2023.

[31] DiploFoundation. Digital watch newsletter – issue 94, november 2024, 2024. Consulté en janvier 2025.

[32] Hugo Ayats. Construction de graphes de connaissance à partir de textes avec une i.a. centrée-utilisateur. In *TALN 2022 – 29e Conférence sur le Traitement Automatique des Langues Naturelles*, pages 33–46, Avignon, France, 2022.

[33] Bochra Hadj Kilani and Najem Dhaher. Méthodologie d'approche du text mining et du nlp pour la recherche urbaine : revue de la littérature. Preprint sur HAL, November 2023. Version du 15 novembre 2023, DOI :10.31219/osf.io/bf8xs.

[34] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4) :1234–1240, 2020.

[35] Microsoft Research. Biogpt : Generative pre-trained transformer for biomedical text generation and mining. *arXiv preprint arXiv :2210.10341*, 2022.

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *arXiv preprint arXiv :1901.00596*, 2019.

[37] Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. *arXiv preprint arXiv :2005.14165*, 2020.

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2020.

[39] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc V. Le, and Denny Zhou. Finetuned language models are zero-shot learners (flan). *arXiv preprint arXiv :2210.11416*, 2022.

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, and Quoc Le. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv :2201.11903*, 2022.