
بسم الله الرحمن الرحيم

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ DES MATHEMATIQUES ET D'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Génie Logiciel

Par :

Amel BENAYADA
Amel MEKADDIM

Sur le thème

Exploitation de Base de Données externe pour la génération de réponses augmentées

---

Soutenu publiquement le 10 / 06 / 2025 à Tiaret devant le jury composé de :

Mr BAGHDADI Mohamed	MCA U.I.K.Tiaret	Président
Mr BOUBEKEUR Aicha	MAA U.I.K.Tiaret	Encadrant
Mr LAARADJ Zohra	MAA U.I.K.Tiaret	Examinateur

2024-2025

REMERCIMENTS

En tout premier lieu, on remercie le bon Dieu de nous avoir donné la force de mener à bien ce travail. Au nom de Dieu, le Clément et le Miséricordieux, louange à Allah.

Nous remercions spécialement notre encadrante de recherche, professeure Aicha BOUBKEUR, dont la rigueur scientifique, la disponibilité et les qualités humaines nous ont profondément touchées.

Nos remerciements s'adressent également aux membres du jury qui nous ont fait le grand honneur d'évaluer ce travail.

Nous étendons nos remerciements à tous les enseignants de notre université qui ont contribué à notre formation, chacun à sa manière, par leur dévouement, leur professionnalisme et la qualité de l'enseignement dispensé.

Finalement, nous tenons à remercier nos parents et nos surs qui nous ont soutenues et encouragée tout au long de cette recherche.

DÉDICACES

Amel B

Je dédie ce travail à ma chère mère, pour son amour inconditionnel et son soutien constant.
Je rends hommage à la mémoire de mon père, que Dieu ait son âme, dont les valeurs continuent de me guider.
Mes sincères remerciements vont également à toute ma famille mes tantes, mes oncles pour leur encouragement.
À mes amis et à tous ceux qui m'ont soutenue de près ou de loin, recevez l'expression de ma profonde gratitude.

Amel M

Je dédie ce travail à moi-même, celui sur lequel j'ai gravé mon nom, ainsi qu'à mes chers parents.
J'espère qu'ils y trouveront une source de fierté et d'honneur.
Un grand merci à mes surs pour avoir rendu ce travail si empreint d'amour.
Et merci à mes amis pour leur soutien sans faille.

RÉSUMÉ

Ce mémoire sintéresse à lutilisation des modèles de langage à grande échelle (LLM) pour améliorer le processus de recherche dinformation grâce à trois approches complémentaires : la recherche vectorielle, la génération augmentée ou guidée contextuellent et le fine-tuning. Lobjectif est de générer des réponses plus pertinentes, contextuelles et personnalisées, en intégrant des sources de connaissances externes. Chaque méthode a été évaluée séparément pour en mesurer les apports spécifiques, et leur combinaison permet denrichir les réponses générées. Le travail présente les choix techniques, les résultats expérimentaux, et propose des pistes d'amélioration pour les systèmes intelligents.

Mots clés : Modèle de langage à grande échelle, Système intelligent, Génération augmentée, Prompting, Fine-Tuning, Recherche sémantique.

ABSTRACT

This thesis explores the use of large language models (LLMs) to enhance information retrieval systems through three complementary approaches: vector search, augmented generation, and fine-tuning. The goal is to produce more relevant, contextual, and personalized responses by integrating external knowledge sources. Each method was individually evaluated to assess its specific benefits, and their combination allows for richer, more up-to-date answers. The work presents technical choices, experimental results, and improvement prospects for intelligent systems.

Keywords: Large Language Models (LLM), Prompting, Fine-Tuning, Semantic Search, Augmented Generation, Intelligent Systems

TABLE DES MATIÈRES

Contents

LISTE DES FIGURES

List of Figures

LISTE DES TABLEAUX

List of Tables

LISTE DES ABRÉVIATIONS

- IA : Intelligence Artificielle
- ML : Machine Learning
- DL : Deep Learning
- ANN : Artificial Neural Network
- DNN : Deep Neural Network
- CNN : Convolutional Neural Network
- RNN : Recurrent Neural Network
- LSTM : Long Short-Term Memory
- NLP : Natural Language Processing
- NLU : Natural Language Understanding
- NLG : Natural Language Generation
- POS Tagging : Part-Of-Speech Tagging
- NER: Named Entity Recognition
- BOW : Bag Of Words
- TF-IDF : Term Frequency - Inverse Document Frequency
- LM : Language Models
- LLM : Large Language Model
- CRFM : Center for Research on Foundation Models
- GPAI : Global Partnership on Artificial Intelligence
- SVM : Support Vector Machine
- GAN : Generative Adversarial Network
- VAN : Vision Transformer Alternative with Convolution
- GPT : Generative Pre-trained Transformer
- BERT : Bidirectional Encoder Representations from Transformers
- LLaMa : Large Language Model Meta AI
- RAG : Retrieval-Augmented Generation

INTRODUCTION GÉNÉRALE

Les larges modèles de langue (LLM), tels que GPT, BERT ou LLaMA, résolvent les tâches de traitement et de génération de réponses de manière fluide et cohérente, s'adaptant à une diversité importante de domaines et de contextes. Fondamentalement, les LLM utilisent une capacité de plusieurs milliards de paramètres et s'appuient sur des corpus d'entraînement massifs pour effectuer des tâches complexes en compréhension, traduction, résumé, dialogue et bien d'autres.

Cependant, malgré la remarquable performance de leurs tâches, plusieurs limitations critiques ont été identifiées. L'une de leurs principales faiblesses est leur incapacité à fournir des réponses précises, actualisées et spécifiques, en particulier si leurs interrogations portent sur des connaissances non incluses ou faiblement incluses dans leurs données d'entraînement, ce qui peut conduire à des hallucinations.

Dans le domaine éducatif, par exemple, les utilisateurs peuvent chercher à obtenir des informations précises et spécifiques sur des universités, des professeurs, ou des évaluations. Or, ces données ne sont généralement pas présentes dans les corpus d'entraînement des LLM généralistes. Par conséquent, il devient nécessaire de développer des approches avancées qui leurs permettent d'augmenter leur mémoire interne par des sources de connaissances externes, structurées ou semi-structurées.

Face à ces limites, notre question primordiale peut être formulée ainsi:

" Comment développer un assistant intelligent capable de générer des réponses précises, cohérentes et adaptées à un domaine spécifique, en exploitant une ressource de connaissances externe et les capacités avancées des LLM ? "

Ce projet de fin d'études s'inscrit donc dans une perspective d'intégration de la recherche

de connaissance ciblée au service des LLM. L'objectif est de permettre à un utilisateur d'interagir en langage naturel avec le système, qui générera des réponses plus en plus précises, cohérentes et contextuellement adaptées. La solution adoptée s'appuie sur une progression méthodique, allant d'une simple recherche sémantique à une génération de réponses informatives, guidée par des techniques de Prompting en Zéro et Few shot et par un fine-tuning du modèle :

- Dans un premier temps, le LLM est couplé à une ressource de connaissances externe pour effectuer une recherche sémantique. Les réponses pertinentes sont générées en fonction de la requête de l'utilisateur.
- La deuxième méthode exploite des techniques Prompting en Zero-shot et Few-shot afin d'améliorer la qualité des réponses. Ces prompts soigneusement conçus sont utilisés pour augmenter et guider le LLM dans la génération de réponses plus pertinentes et informées.
- Enfin, une méthode de personnalisation par fine-tuning permet d'adapter le LLM au domaine spécifique des universités et des enseignants. Cette étape vise à affiner la compréhension fine des questions posées et à renforcer la cohérence ainsi que la pertinence des réponses générées.

Ce mémoire est structuré en 03 chapitres :

Le Chapitre 1 présente les bases théoriques et techniques liés aux évolutions récentes de l'IA et du traitement du langage naturel (NLP).

Le Chapitre 2 décrit les fondements théoriques essentiels à la compréhension et à la mise en uvre des systèmes d'IA générative, notamment les LLMS et leurs applications transformatrices.

Le Chapitre 3 détaille les étapes pratiques de la mise en uvre, de la construction de la base vectorielle à l'intégration du LLM, en passant par la conception des prompts et les mécanismes de génération, où les scénarios de test, les résultats obtenus, les performances du système dans différentes configurations (Zéro-shot, Few-shot, Fine-tuning), ainsi que leur analyse critique sont ainsi présentées.

Enfin, une synthèse des contributions du mémoire, ses limites, et les pistes d'amélioration futures.

CHAPITRE 1

AVANCÉES DE LINTELLIGENCE ARTIFICIELLE

1.1 Introduction

Ce chapitre pose les bases théoriques de la solution proposée, en présentant les concepts clés liés à l'intelligence artificielle(IA classique, IA générative) et au traitement du langage naturel(NLP). Il offre une vue d'ensemble des avancées technologiques à l'origine de ces disciplines et souligne leur impact transversal sur divers secteurs comme la santé, l'industrie et l'éducation.

1.2 Intelligence artificielle

Au cours de lannée 1950, le terme ń intelligence artificielle (IA) ż a été proposé par deux chercheurs pionniers de la data science, John McCarthy et Marvin Lee Minsky ; plus de soixante-dix ans plus tard, ce domaine sest très largement développé, au point dintégrer véritablement le quotidien des usagers humains et de samplifier de manière exponentielle[1].

L'intelligence artificielle (IA) est une technologie qui cherche à réproduire le fonctionnement du cerveau humain en imitant ses capacités cognitives.L'IA repose sur trois composants principaux :

- Algorithmes sophistiqués.
- Ensembles de données volumineux.
- Systèmes et matériels informatiques puissants.

Ces trois composants permettent aux systèmes dIA dapprendre et de progresser de manière itérative, en analysant et en intégrant toutes les informations à disposition[1].

Les modèles d'IA sont de diverses sortes, chacun dentre eux étant dédié à un type spécifique dapplications : les méthodes de *Machine Learning* permettent à une machine dacquérir des compétences dapprentissage et de prise de décision à partir de données, tandis que celles de *Deep Learning* se concentrent sur le traitement de limage et de la parole grâce à des réseaux de neurones. LIA symbolique permet de construire des systèmes intelligents capables de simuler le raisonnement humain pour résoudre des problèmes complexes ; tandis que le *NLP* et la vision par ordinateur donnent aux machines la possibilité de comprendre le langage humain et lenvironnement visuel, avec des applications dans de nombreux secteurs dactivité[1].

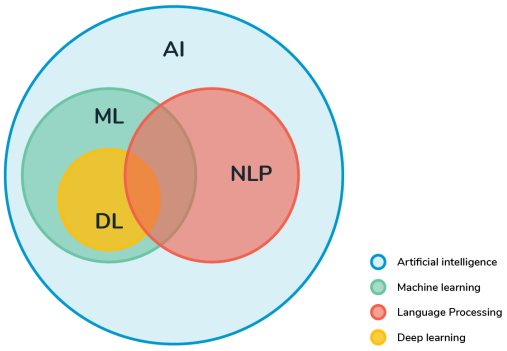

Figure 1.1: Domaines de lintelligence artificielle [13].

1.3 Machine Learning (ML)

Le machine learning, ou apprentissage automatique, est une composante clé de lIA. Il fournit aux ordinateurs un accès à de grandes bases de données, leur permettant danalyser les informations et dapprendre de façon autonome, sans quun être humain ait besoin de les reprogrammer.

De fait, les algorithmes se modifient sans intervention humaine et dépassent progressivement leur programmation initiale[2].

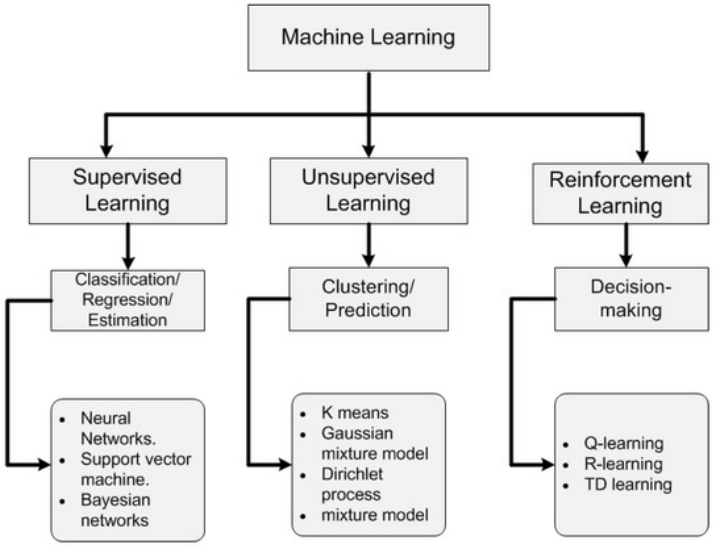

Figure 1.2: Types dapprentissage automatique[14].

Il existe plusieurs types dapprentissage automatique (figure 1.2) :

- **Apprentissage supervisé (Supervised Learning)** : basé sur des données étiquetées où chaque entrée est associée à une sortie connue. Utilisé pour la classification dimages, la détection de spam, etc.
- **Apprentissage non supervisé (Unsupervised Learning)** : seules les données dentrée sont fournies et le système découvre des structures cachées, utilisé pour le clustering, la détection d'anomalies, etc.
- **Apprentissage par renforcement (Reinforcement Learning)** : le système interagit avec son environnement pour maximiser une récompense à travers des essais et erreurs, utilisé notamment en robotique et dans les jeux vidéo.

1.4 Deep Learning (DL)

Le deep learning, ou apprentissage profond, est basé sur lutilisation de réseaux de neurones, qui reproduisent les fonctions du cerveau humain. Cette technique permet de créer une machine virtuelle composée de milliards dunités. Chaque unité est chargée deffectuer des calculs

simples avec une grande précision. Grâce à cela, il devient possible de résoudre rapidement des problèmes complexes et dexécuter des tâches dans des domaines où limprévu est fréquent[3]:

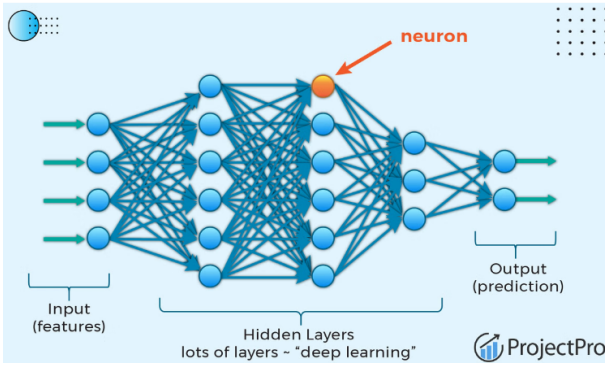

Figure 1.3: Architecture d'un réseau de neurones profond (Deep Learning)[15].

a. ANN (Artificial Neural Network)

Les réseaux neuronaux artificiels (ANN) modélisent le fonctionnement des neurones biologiques en organisant des unités interconnectées en couches hiérarchiques. Comme dans le cerveau, chaque couche traite des informations et les transmet à la suivante. Un ANN basique se compose de trois couches : lentrée (réception des données), la couche cachée (traitement des informations) et la sortie (décision finale basée sur les données analysées)[3].

b. DNN (Deep Neural Network)

Les réseaux neuronaux profonds (DNN) étendent les réseaux neuronaux artificiels (ANN) en ajoutant des couches intermédiaires entre lentrée et la sortie. Ces couches exécutent des opérations mathématiques (linéaires ou non linéaires) pour convertir les données en résultats probables, avec des seuils ajustables par lutilisateur. La décomposition des calculs en couches distinctes et leur grande quantité expliquent leur nom de ń réseaux profonds ż[3].

c. CNN (Convolutional Neural Network)

Les réseaux neuronaux convolutifs (CNN) sont des variantes de réseaux neuronaux spécialisés en vision par ordinateur, basés sur trois types principaux de couches : les couches convolutionnelles, les couches de pooling (ou sous-échantillonnage) et les couches entièrement

connectées. Certaines implémentations incluent aussi des couches de normalisation (comme Batch Normalization). La convolution applique un noyau (filtre) à une image dentrée pour générer une carte de caractéristiques, tandis que le pooling réduit la dimensionnalité en extrayant des informations clés depuis des sous-régions de limage. Ces mécanismes permettent une analyse efficace des données visuelles grâce à une hiérarchie de couches[3].

d. RNN (Recurrent Neural Network)

Les réseaux neuronaux récurrents (RNN) sont une classe de modèles dapprentissage profond spécialisés dans le traitement des données séquentielles , comme le langage, la parole ou les séries temporelles. Contrairement aux réseaux neuronaux classiques qui supposent lindépendance des points de données, les RNN possèdent une boucle temporelle qui leur permet de maintenir une forme de mémoire de linformation précédente. Cela leur donne la capacité de prendre en compte le contexte passé lors du traitement dun nouvel élément dans la séquence[3].

e. LSTM (Long Short-Term Memory)

LSTM est une architecture de réseau neuronal récurrent (RNN) conçue pour mémoriser des informations sur de longs intervalles. Elle résout ainsi le problème de dépendance à long terme rencontré par les RNN classiques, grâce à ses portes spécifiques (d'entrée, de sortie et de forget) qui contrôlent le flux dinformation. Spécialisée dans la prédiction à partir de séries temporelles et la classification de données séquentielles, elle est utilisée dans des applications comme Google Translate, Apple Siri et Amazon Alexa[3].

f. Transformers

Larchitecture Transformer, introduite en 2017 dans larticle *"Attention is All You Need"*, est une innovation majeure qui est devenue la base essentielle des grands modèles de langage (LLMs), révolutionnant le traitement du langage naturel[9].

La version simplifiée de larchitecture Transformer se présente de la manière suivante :

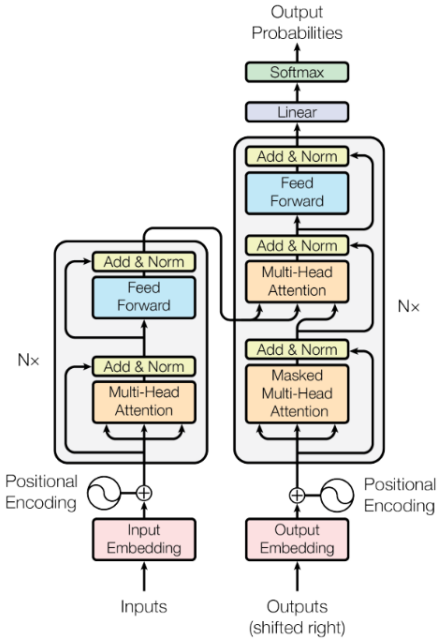

Figure 1.4: Transformer - Architecture du modèle[9].

Un Transformer se compose essentiellement des éléments suivants :

1. Embedding Layer (Couche d'embedding): cette couche convertit les tokens dentrée (mots ou parties de mots) en vecteurs, qui sont des représentations numériques intégrant à la fois le sens et le contexte du texte. Ces vecteurs permettent au modèle de traiter efficacement les données textuelles en leur donnant une structure mathématique exploitable.

2. Positional Encoding (Encodage positionnel) : cette étape ajoute des informations sur la position de chaque token dans la séquence à son embedding. Cela compense labsence de traitement séquentiel inhérent au Transformer, permettant au modèle de comprendre lordre des mots. Sans cette étape, le modèle ne pourrait pas différencier deux phrases ayant les mêmes mots mais dans un ordre différent.

3. Encoder and Decoder (Encodeur et Décodeur) : l'**encodeur** traite le texte dentrée pour en extraire le contexte, en utilisant des mécanismes tels que l'auto-attention et des réseaux de neurones feedforward. Le **décodeur**, quant à lui, génère des réponses cohérentes en prédisant les mots suivants dans une séquence, en tenant compte du contexte extrait par l'encodeur.

4. Self-Attention Mechanism (Mécanisme d'auto-attention) : le mécanisme d'auto-attention permet au modèle dévaluer limportance relative des différents mots dans la séquence dentrée. Cela lui donne la capacité de comprendre le contexte et les relations entre les mots sur lensemble du texte, ce qui est crucial en langage naturel, où le sens dun mot peut varier selon son contexte dans une phrase. Par exemple, le mot "banque" peut désigner une institution financière ou le bord d'une rivière, selon le contexte.

5. Feedforward Neural Networks (Réseaux de neurones feedforward) : à la fois dans lencodeur et le décodeur, ces réseaux appliquent des transformations supplémentaires aux données traitées par le mécanisme dauto-attention. Ils captent ainsi des caractéristiques plus fines et un contexte plus nuancé du langage, permettant au modèle de mieux généraliser et de produire des résultats plus précis.

6. Layer Normalization & Residual Connections (Normalisation des couches et connexions résiduelles) : ces techniques, intégrées dans les blocs du modèle, augmentent la stabilité de lentraînement, évitent le problème de disparition du gradient et facilitent lapprentissage de réseaux neuronaux plus profonds. La normalisation des couches permet de maintenir les activations dans une plage stable, tandis que les connexions résiduelles aident à propager efficacement les gradients pendant l'entraînement.

1.5 Traitement du langage naturel

Le NLP est une discipline centrée sur la compréhension, la manipulation et la génération de langage humain par les machines. Il se situe à la croisée de linformatique et de la linguistique, et vise à permettre aux machines dinteragir directement avec les êtres humains en utilisant le langage naturel[5]. Le NLP peut être divisé en deux sous langages :

- NLU (Natural Language Understanding) ou compréhension du langage naturel, a pour objectif dinterpréter le sens dun texte. Pour cela, elle analyse la nature, la structure et les relations entre les mots. Cette tâche repose sur la résolution des ambiguïtés suivantes:

  - Ambiguïté lexicale : un mot peut avoir plusieurs significations ;
  - Ambiguïté syntaxique : une phrase admet plusieurs structures grammaticales ;
  - Ambiguïté sémantique : un énoncé porte plusieurs interprétations ;
  - Ambiguïté anaphorique : un terme renvoie à une référence précédente, mais sons sen est modifié ou incertain.

  Lidentification du sens des mots sappuie ensuite sur des lexiques (dictionnaires spécialisés) et des grammaires formelles. Toutefois, cette tâche est complexifiée par des phénomènes comme la synonymie (sens identiques) ou la polysémie (multiples significations)[5].

- NLG (Natural Language Generation) ou génération du langage naturel, génère automatiquement du texte à partir de données structurées, en produisant des phrases grammaticalement correctes et sémantiquement cohérentes. Cette tâche, complexe et centralisée en NLP, suit trois étapes clés :

  - Structuration du contenu : sélection et organisation des informations principales issues des données ;
  - Planification des phrases : définition de la syntaxe et de la logique denchaînement des idées ;
  - Production linguistique : transformation des éléments structurés en phrases lisibles et adaptées au contexte.

  Les défis résident dans la précision contextuelle, la gestion des ambiguïtés et la fluidité des expressions, tout en maintenant une grammaire rigoureuse[5].

Le NLP repose sur des techniques de normalisation textuelle permettant de structurer et d'exploiter les données linguistiques. Ces étapes incluent la segmentation des phrases, la tokenisation, le stemming, la lemmatisation, le balisage grammatical (POS tagging), la reconnaissance d'entités nommées (REN) et la suppression des mots vides (stop words) [5]. Une fois le texte normalisé, il est converti en représentations numériques exploitables par des modèles [5]: sac de mots (BoW), TF-IDF, N-grammes et word embeddings (Word2Vec, GloVe, FastText), facilitant ainsi l'analyse sémantique et contextuelle (Table 1.1).

Table 1.1: Techniques NLP de Base

Technique	**Entrée**	**Sortie**
Sentence Segmentation	"Bonjour ! Comment allez-vous ?"	["Bonjour !", "Comment allez-vous ?"]
Tokenization	"I love NLP!"	["I", "love", "NLP", "!"]
Stemming/Lemmatization	"running", "better"	Stemming:["run", "better"]; Lemmatization:["run", "good"]
Part-of-Speech tagging	"The cat sleeps"	[("The", "DT"), ("cat", "NN"), ("sleeps", "VBZ")]
Named Entity Recognition	"Apple is hiring."	["Apple" (ORG)]
Stop Words	"The cat is cute"	["cat", "cute"]
Bag-of-words	Documents : ["cat dog", "dog mouse"]	"cat":1, "dog":2, "mouse":1
TF-IDF	"le chat mange une souris"	["le": 0.0, "chat": 0.0, "mange": 0.378, "une": 0.511, "souris": 0.772]
N-grammes	Documents : ["cat dog", "dog mouse"]	Bigrammes:[("cat", "dog"): 1, ("dog", "mouse"): 1]
Word Embedding	"king", "queen", "man", "woman"	king $\rightarrow$ [0.89, -0.42], queen $\approx$ king - man + woman

Aujourd'hui, le NLP est l'un des principaux moteurs de l'IA et couvre un très large champ d'application[5]:

- Analyse des sentiments : repérer les éléments subjectifs dans un texte pour en extraire lopinion de lauteur. Utilisée pour évaluer le niveau de satisfaction des clients ou des utilisateurs.
- Traduction automatique : repose sur une analyse approfondie et une modélisation du texte, notamment grâce à des techniques comme la Traduction automatique statistique.
- Chatbots : systèmes qui sappuient sur le NLP pour gérer efficacement des tâches courantes, comme informer les clients sur des produits ou répondre à leurs questions.
- Classification des textes : permet de trier, structurer et catégoriser des ensembles de textes. Utilisée dans la modération de contenu, par exemple pour détecter des fake news.

Autres applications : reconnaissance des caractères, génération de texte, synthèse automatique, marketing, détection de fraude, correction automatique, etc.

1.6 Modèle de langage (Language Models)

Un modèle de langage (LM) est un modèle d'apprentissage automatique qui prédit les mots suivants dans une phrase ou attribue une probabilité à chaque mot possible. Il peut également évaluer la probabilité d'une phrase entière. Ces modèles sont utiles pour générer des phrases grammaticalement correctes, corriger des erreurs (orthographe, grammaire), améliorer la reconnaissance vocale et aider dans les systèmes de communication alternative en suggérant des mots probables[7].

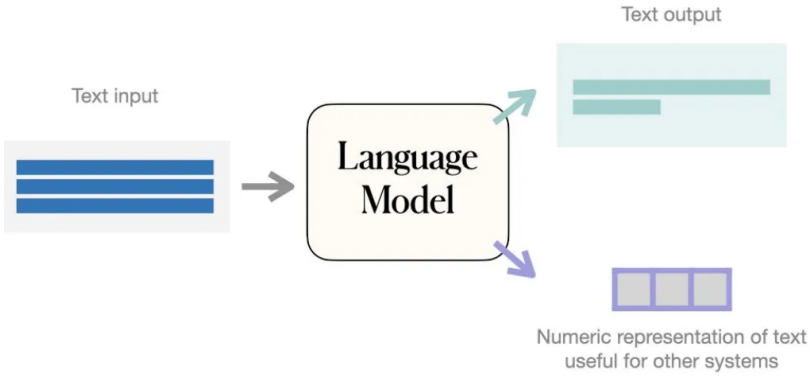

Figure 1.5: Modèle de langage[16].

Il existe deux catégories principales des modèles de langage[7]:

a. Modèles basés sur les statistiques (n-grammes)

Les modèles probabilistes, comme les n-grammes, prédisent la probabilité dun mot en fonction des n mots précédents dans une séquence. Ces probabilités sont calculées à partir de grandes bases de données textuelles en analysant la fréquence des mots et des séquences. Bien quils soient simples et efficaces pour capturer des relations locales, ils ne prennent pas en compte les dépendances à long terme entre les mots[8].

Plusieurs types de modèles de n-grammes sont à distinguer, notamment :

- Unigrammes, qui traitent chaque mot de façon isolée.
- Bigrammes, qui examinent la probabilité dun mot en fonction du mot précédent.

- Trigrammes, qui prennent en compte les deux mots précédents pour évaluer sa probabilité.

Ce principe peut se prolonger à des séquences plus longues, comme des 4-grammes, 5-grammes, etc.

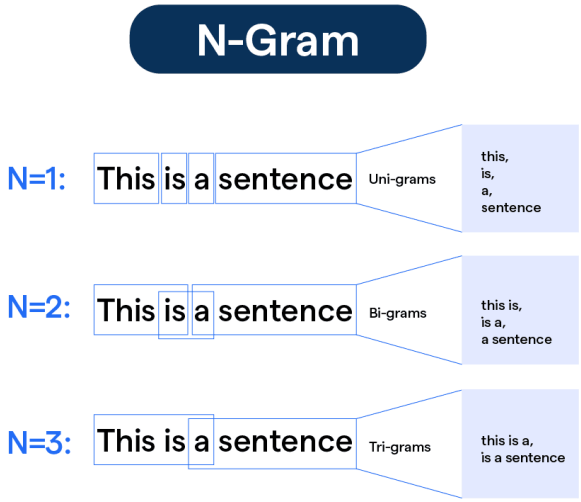

Figure 1.6: Types de modèles de n-grammes[17].

b. Modèles de langues modernes basés sur les réseaux de neurones

Les modèles modernes utilisent des réseaux de neurones, tels que les RNN, LSTM ou Transformers, pour prédire la probabilité des mots dans un contexte plus large. Contrairement aux n-grammes, ils intègrent une mémoire qui leur permet de capturer des relations complexes entre les mots, même éloignés dans une phrase. Les mots sont représentés sous forme de vecteurs denses (*word embeddings*), ce qui permet de mesurer leur similarité sémantique et d'améliorer la généralisation[8].

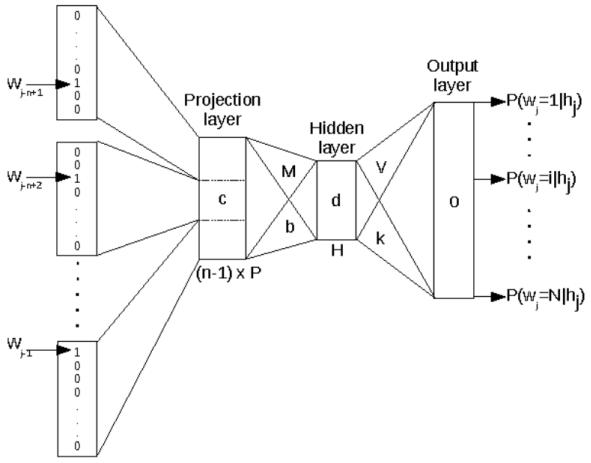

Figure 1.7: Modèles de langage modernes basés sur les réseaux de neurones[18].

1.7 Intelligence Artificielle Générative

L'IA générative est une branche de l'IA spécialisée dans la création autonome de contenus, qu'il s'agisse de textes, d'images, de vidéos, de musique ou d'autres formes de données. Contrairement à l'IA classique, qui excelle dans des tâches spécifiques comme la classification ou la prédiction, l'IA générative vise à produire de nouveaux contenus originaux qui imitent ceux créés par des humains. Son objectif est de générer des données réalistes et innovantes dans divers formats. Ces progrès ont été rendus possibles grâce à l'émergence des modèles de fondation, des architectures puissantes capables de traiter et de généraliser des connaissances à partir de vastes ensembles de données[4].

Les modèles de fondation, introduits en 2021 par le Centre de recherche sur les modèles de fondation (CRFM) de Stanford, désignent des systèmes dIA entraînés sur des données massives, généralement via une auto-supervision à grande échelle, et adaptables à un large éventail de tâches. Souvent qualifiés de ń systèmes dIA à usage général ż (GPAI), ces modèles sont conçus pour accomplir des tâches variées, comme la génération de texte, la manipulation dimages ou la création audio, tout en servant de base polyvalente pour des applications spécialisées. Des modèles comme GPT-3 et GPT-4 dOpenAI, utilisés dans ChatGPT ou Bing, montrent concrètement ce concept en action. Certains modèles sont monomodaux (traitant une seule modalité, comme le texte), tandis que dautres, multimodaux, combinent plusieurs types dentrées (texte, images, vidéos) pour produire des sorties diversifiées[4].

Leur développement repose sur des volumes considérables de données issues dInternet et nécessite des ressources informatiques importantes. Ces modèles exploitent le principe de lapprentissage par transfert, permettant de réutiliser les connaissances acquises dans un domaine pour résoudre des problèmes dans dautres contextes. Cette capacité leur confère une grande adaptabilité et une diversité dapplications, justifiant leur rôle central dans lévolution de lintelligence artificielle moderne[4].

L'IA générative se distingue de l'IA traditionnelle par ses capacités étendues et ses applications innovantes. Si lIA traditionnelle se concentre sur lanalyse des données existantes, la reconnaissance de motifs et la réalisation de prédictions, lIA générative va au-delà en créant de nouveaux contenus à partir de ce quelle a appris lors de son entraînement. Elle excelle dans la génération de données originales, permettant non seulement dinterpréter les informations, mais aussi de produire du texte, des images ou dautres formes de contenu basé sur des modèles identifiés. En résumé, lIA traditionnelle sert à comprendre et classer les données, tandis que lIA générative utilise ces données pour créer quelque chose de nouveau.

Afin de clarifier ces différences, une analyse détaillée sous forme de tableau (Table 1.2)est proposée pour mettre en lumière les spécificités de chaque approche.

Table 1.2: Tableau comparatif entre lIA Classique et lIA Générative

Critère	IA Traditionnelle	IA Générative
Type de Contenu	Structuré (tableaux, lignes, colonnes)	Non structuré (texte, images, audio, vidéo, etc.)
Modèles Utilisés	Modèles spécifiques : Régressions linéaires/logistiques, arbres de décision, SVM, réseaux neuronaux simples.	Modèles de fondation : (transformers, GANs, VAEs)
Apprentissage	Apprentissage supervisé ; le modèle est formé sur des données étiquetées où des entrées et des sorties sont fournies.	Apprentissage supervisé ; le modèle est formé sur des données étiquetées où des entrées et des sorties sont fournies.
Capacités	Analyse de patterns, prédiction, classification, optimisation.	Création de contenu original, génération de variations, transfert de style, synthèse de données.
Polyvalence	Spécialisée dans des tâches spécifiques avec des performances élevées.	Plus polyvalente, capable de générer divers types de contenu à partir dune même architecture.
Limites	Peu flexible pour des tâches créatives, nécessite des données bien étiquetées.	Risque de "hallucinations", manque dexplicabilité.
Exemples dApplications	Systèmes de recommandation, reconnaissance dimages, analyse de sentiments, trading algorithmique.	Chatbots avancés (GPT), génération dimages (DALL-E), création musicale (Jukebox), design assisté.

1.8 Domaines d'application de l'IA Avancée

L'IA est en constante évolution et trouve de nouvelles applications dans divers domaines[6]:

1. Santé et Médecine

- Diagnostic Médical : les algorithmes d'IA analysent des données complexes (imagerie, dossiers médicaux) pour identifier des motifs invisibles à l'il humain, soutenant ainsi les professionnels de santé dans des décisions critiques.
- Recherche Pharmaceutique : en modélisant des interactions moléculaires, l'IA accélère la découverte de médicaments, réduisant le temps et les coûts associés aux essais cliniques.
- Médecine de Précision : les systèmes d'IA intègrent des données génomiques et historiques des patients pour concevoir des traitements personnalisés, améliorant l'efficacité thérapeutique[6].

2. Transport et Véhicules Autonomes

Les véhicules autonomes exploitent des technologies d'IA (reconnaissance d'images, apprentissage profond) pour détecter les obstacles, planifier des trajets et prendre des décisions en temps réel, visant à réduire les accidents et optimiser la mobilité urbaine[6].

3. Assistance Virtuelle et Service Client

- Chatbots Intelligents : déployés dans les centres d'appel et les plateformes en ligne, ces outils répondent aux requêtes clients 24/7, intégrant le traitement du langage naturel (NLP) pour des interactions fluides.
- Assistants Vocaux : Siri, Alexa ou Google Assistant utilisent l'IA pour interpréter les commandes orales, gérer des tâches quotidiennes et s'adapter aux préférences des utilisateurs[6].

4. Finance et Technologie Financière (Fintech)

- Analyse Prédictive : l'IA détecte les fraudes en identifiant des comportements anormaux, prédit les tendances du marché et évalue les risques de crédit via des données hétérogènes.
- Gestion Financière : des applications utilisent l'IA pour conseiller des budgets personnalisés, automatiser les épargnes ou optimiser les investissements[6].

5. Commerce Électronique et Marketing

Les moteurs de recommandation d'IA croisent les données d'achat, les préférences et les comportements de navigation pour proposer des produits ou contenus adaptés, boostant l'engagement client[6].

6. Éducation

Les plateformes éducatives intelligentes adaptent le contenu pédagogique au rythme et aux lacunes des apprenants, facilitant un apprentissage personnalisé et interactif[6].

1.9 Conclusion

Le long de ce chapitre, nous avons présenté les fondements de lIA classique, de lIA générative et du NLP. Ces concepts constituent les bases théoriques et techniques nécessaires pour comprendre les évolutions récentes de lIA, notamment lémergence des grands modèles linguistiques (LLMs) et leurs applications transformatrices.

Le prochain chapitre sera dédié à une exploration approfondie des LLMs, en mettant particulièrement en avant leur capacité à intégrer des données externes via des méthodes comme le RAG (Retrieval-Augmented Generation) . Nous explorerons également des méthodes avancées telles que le fine-tuning , le prompt engineering et d'autres stratégies innovantes.

CHAPITRE 2

VERS LA GÉNÉRATION AUGMENTÉE PAR RECHERCHE

2.1 Introduction

Ce chapitre illustre la dynamique actuelle du domaine de l'IA avancée, avec des efforts concentrés sur l'amélioration de la pertinence contextuelle, la réduction des hallucinations et l'adaptation à des applications spécifiques.

La génération augmentée est une approche innovante du NLP.Elle améliore considérablement les capacités des LLM traditionnels en intégrant un accès en temps réel à des sources de données externes.Cette méthodologie permet de générer du contenu non seulement contextuellement pertinent, mais également à jour, remédiant ainsi à une limitation critique des LLM conventionnels qui reposent uniquement sur des données statiques de pré-apprentissage, ce qui peut conduire à des réponses absolues ou inexactes.

2.2 Large Language Models

Le langage est essentiel pour les interactions humaines et la communication avec les machines, permettant en même temps l'expression personnelle et lexécution de tâches complexes.

Pour répondre aux besoins croissants en traduction, résumé automatique ou échanges conversationnels, des modèles généralistes ont été conçus par les humains. Ces dernières années, grâce à des architectures innovantes comme les Transformers, une puissance de calcul accrue et lutilisation de vastes bases de données, les chercheurs ont développé des LLMs. Ces systèmes, créés par lingéniosité humaine, produisent des résultats proches des performances humaines et génèrent du texte cohérent, tout en sadaptant à une multitude de tâches complexes[10].

Lhistoire des LLMs débute avec les premières expérimentations en traduction automatique pendant la seconde guerre mondiale. Au fil du temps, les chercheurs sont passés de systèmes basés sur des règles à des approches statistiques, avant de concevoir l'architecture Transformer, qui a révilutionné le domaine. Des étapes clés comme BERT et Seq2Seq ont permis lémergence de modèles avancés tels que GPT-4 et LLaMA, soulignant lingéniosité humaine et la collaboration scientifique derrière ces innovations[10].

La figure 1.8 montre une chronologie du développement des LLMs comptant plus de 10 milliards de paramètres, en mettant en lumière des avancées marquantes et des évolutions récentes. Conçue par des chercheurs humains, cette visualisation montre la progression rapide de ces modèles, en soulignant des étapes clés et des innovations qui ont élargi leurs capacités et repoussé leurs limites.

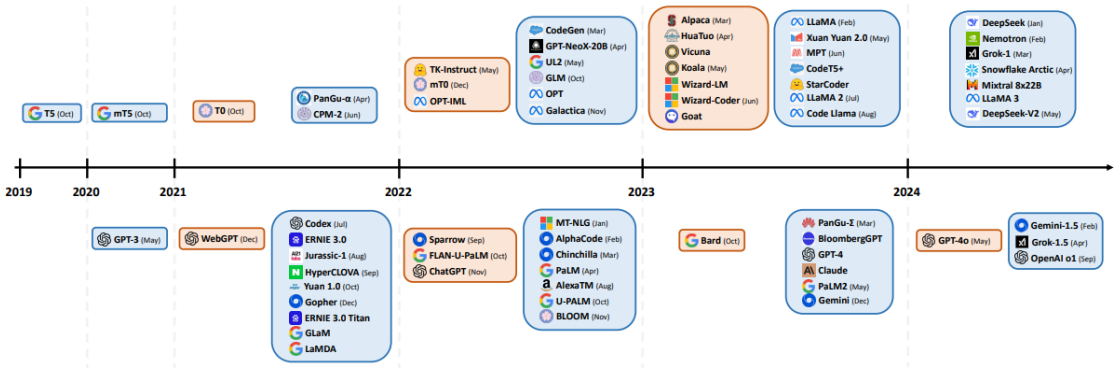

Figure 2.1: Affichage chronologique des versions des LLMs : les cartes bleues représentent les modèles pré-entraînés, tandis que les cartes orange correspondent aux modèles ajustés par instruction. Les modèles situés dans la moitié supérieure indiquent une disponibilité en open source, tandis que ceux dans la moitié inférieure sont en source fermée[10].

2.2.1 Architecture des LLMs

Larchitecture Transformer est considérée comme le bloc de construction fondamental des LLMs. Elle a été conçue pour permettre aux réseaux neuronaux de traiter efficacement les données séquentielles[10].

A. Encodeur-Décodeur

Cette architecture traite les entrées via un encodeur, qui transmet ensuite une représentation intermédiaire au décodeur pour générer la sortie. L'encodeur analyse toute la séquence en utilisant l'auto-attention, tandis que le décodeur traite la séquence progressivement, en implémentant une attention croisée.

L'attention est calculée par :

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

où :

- Q (Query), K (Key), V (Value) sont des matrices apprises

- d_k est la dimension des clés (facteur de normalisation)

Conçue par des chercheurs humains, cette approche permet de traiter efficacement des tâches comme la traduction automatique.

B. Décodeur

Cette architecture est généralement intégrée dans un cadre encodeur-décodeur plus large, mais elle peut également être utilisée de manière autonome dans certains modèles. Un exemple montrant ces concepts est présenté dans la Figure 2.2, qui montre les différentes configurations et leurs interactions.

- **Décodeur Causal** : il s'agit d'une architecture sans encodeur, où seul un décodeur est utilisé pour traiter les données et générer la sortie. Dans ce cas, chaque mot prédit ne dépend

que des mots précédents dans la séquence.

Équation d'attention masquée :

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \quad \text{où } M_{ij} = \begin{cases} 0 & \text{si } i \leq j \\ -\infty & \text{sinon} \end{cases} \tag{2.2}$$

Cette conception humaine est particulièrement adaptée aux tâches de génération de texte.

- **Décodeur Préfixe (Non-Causal)** : aussi appelé décodeur non-causal, il calcule l'attention sans se limiter strictement aux informations passées. L'attention est bidirectionnelle, permettant ainsi une meilleure prise en compte du contexte global.

Équation d'attention bidirectionnelle :

$$\text{Attention}_{\text{bidir}}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad \text{avec } K,V \in \mathbb{R}^{n\times d_k} \tag{2.3}$$

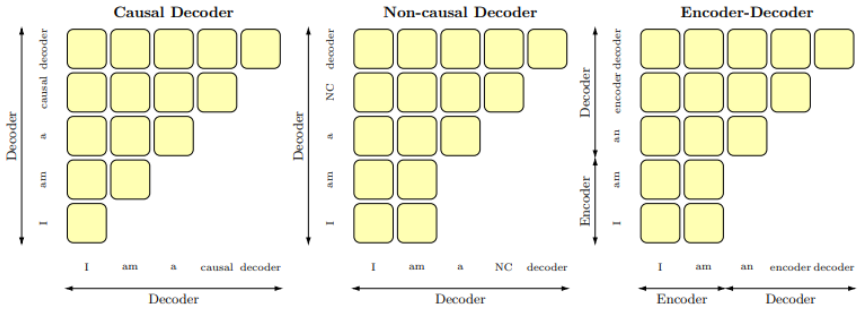

Figure 2.2: Modèles d'attention dans une architecture décodeur causal, décodeur non-causal et encodeur-décodeur[10].

C. Mixture-of-Experts (MoE)

Cette variante de larchitecture Transformer repose sur des experts indépendants organisés en parallèle, avec un routeur chargé de diriger les tokens vers les experts appropriés. Ces experts sont des couches feedforward placées après le bloc dattention. Développée par des ingénieurs humains, larchitecture MoE est une solution efficace et éparse qui offre des performances comparables aux modèles denses tout en augmentant la taille du modèle sans accroître proportion-

nellement le coût de calcul. En effet, seuls quelques experts sont activés à la fois, optimisant ainsi les ressources[10].

D. Paramètres clés dans un LLM

Les LLM ont plusieurs paramètres clés qui définissent leur architecture et leur comportement[10].

Taille du modèle

La taille du modèle correspond au nombre de paramètres présents dans le LLM. Un paramètre est une variable apprise par le modèle lors de son entraînement. Généralement mesurée en milliards ou en trillions, la taille dun modèle influence directement ses performances : plus il est grand, meilleures sont ses capacités, mais cela implique également un besoin accru en ressources informatiques pour lentraînement et lexécution[10].

La taille du modèle peut être exprimée comme suit :

$$\text{Taille du modèle} = \text{Nombre de Paramètres} \tag{2.4}$$

Par exemple, GPT-3 est un grand modèle avec 175 milliards de paramètres.

Taille du vocabulaire

La taille du vocabulaire correspond au nombre de tokens uniques (tels que des mots, de la ponctuation, etc.) sur lesquels le LLM est entraîné. Un vocabulaire plus étendu permet au modèle de mieux comprendre et de générer une plus grande variété de langage, mais nécessite également davantage de ressources informatiques pour son entraînement et son fonctionnement.

La taille du vocabulaire peut être exprimée comme suit :

$$\text{Taille du Vocabulaire} = \text{Nombre de Tokens Uniques} \tag{2.5}$$

Par exemple, GPT-2 a un vocabulaire de 1,5 milliard de tokens.

Température

La température est un paramètre qui régule le degré daléatoire dans les réponses du LLM. Une température élevée génère un texte plus créatif et varié, tandis quune température basse produit des réponses plus précises et factuelles.

Mathématiquement, la température peut être représentée comme suit :

$$\text{Probabilité de Sortie} = \frac{\exp(\text{Logit}/\text{Température})}{\sum \exp(\text{Logit}_i/\text{Température})} \tag{2.6}$$

Où Logit est la sortie brute du LLM avant lopération softmax, et Température est le paramètre qui échelle les logits.

Par exemple, si on met la température à 1,0, le LLM choisira surtout les mots les plus probables. Mais si on monte la température à 2,0, il pourra sélectionner des mots moins probables, ce qui rendra le texte plus créatif.

Fenêtre de contexte

La fenêtre contextuelle représente le nombre de mots que le LLM prend en compte pour générer du texte. Une fenêtre plus large permet de produire un texte plus cohérent et pertinent, mais elle nécessite également plus de ressources de calcul pour lapprentissage.

La fenêtre de contexte peut être exprimée comme suit :

$$\text{Fenêtre de Contexte} = \text{Nombre de Mots Considérés} \tag{2.7}$$

Par exemple, si la fenêtre de contexte est définie sur 2, le LLM prendra en compte les deux mots avant et après le mot actuel lors de la génération du mot suivant.

Top-k et Top-p

Ces techniques permettent de filtrer les jetons sélectionnés. Top-k retient les k jetons les plus probables, assurant ainsi une sortie de meilleure qualité. De son côté, Top-p fixe un seuil de probabilité cumulée et ne conserve que les jetons dont la probabilité totale dépasse ce seuil, favorisant ainsi la diversité. Tandis que Top-k évite les réponses absurdes, Top-p offre des résultats plus variés.

Les paramètres Top-k et Top-p peuvent être représentés comme suit :

$$\text{Top-k} = \text{Nombre de Tokens les Plus Probables Considérés} \tag{2.8}$$

$$\text{Top-p} = \text{Seuil de Probabilité Cumulée} \tag{2.9}$$

Par exemple, si vous définissez Top-k à 10, le LLM ne prendra en compte que les 10 mots suivants les plus probables. Si vous définissez le Top-p à 0,9, le LLM ne générera que des mots ayant une probabilité dau moins 0,9.

Ces paramètres, ainsi que dautres comme les séquences darrêt et les pénalités de fréquence et de présence, offrent la possibilité dajuster le comportement du LLM afin de répondre à des besoins et cas dutilisation particuliers.

2.3 Entraînement des LLMs

Lentraînement des LLMs repose sur plusieurs étapes essentielles, nécessaires pour garantir son efficacité. Le processus débute le plus souvent par la collecte et le prétraitement dune quantité considérable de données textuelles hétérogènes, telles que des livres, des articles, des sites web ou dautres sources textuelles. Le corpus ainsi constitué est alors traité pour constituer un ensemble de données servant à lentraînement du LLM. Lentraînement lui-même se déroule selon une méthode non supervisée, dans le cadre de laquelle le modèle doit apprendre à prédire le mot suivant dune séquence considérée, en fonction des mots précédents. Cette tâche est appelée modélisation du langage. Les LLM utilisent des architectures de réseaux neuronaux très sophistiquées, en général de type Transformer, qui leur permettent de réaliser des captures de formes et de dépendances linguistiques complexes. Lobjectif de lentraînement est doptimiser les paramètres du modèle de sorte à maximiser la probabilité dénoncer le mot correspondant à son contexte. Cette optimisation est généralement réalisée au moyen dun algorithme connu sous le nom de ń descente de gradient stochastique ż (SGD) ou lune de ses nombreuses variantes, couplé à une méthode de rétropropagation[10].

2.4 Évaluation des LLMs

Des métriques dévaluation automatique telles que la perplexité, BLEU et ROUGE sont couramment utilisées pour évaluer la qualité des prédictions générées par les modèles de langage[11]:

- **Perplexité** : couramment utilisée pour les modèles de langage, évalue la probabilité quun modèle attribue aux séquences de mots. La perplexité reflète la confiance du modèle dans ses prédictions, une valeur plus basse indiquant une meilleure performance.

Mathématiquement, la perplexité peut être représentée comme suit :

$$\text{Perplexité} = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 P(w_i)} \tag{2.10}$$

où : - N est le nombre total de mots dans la séquence, - $P(w_i)$ est la probabilité attribuée par le modèle au mot w_i.

Une perplexité plus basse indique que le modèle est plus sûr de ses prédictions.

- **BLEU (Bilingual Evaluation Understudy)** : est principalement utilisé pour évaluer les traductions automatiques. Cet outil compare le texte généré par le modèle avec un texte de référence en analysant les correspondances entre les groupes de mots (n-grams).

La formule générale de BLEU est donnée par :

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.11}$$

où : - BP est le facteur de pénalité pour la brièveté du texte généré, - p_n est la précision des n-grams, - w_n est le poids attribué à chaque n-gram.

BLEU mesure la correspondance entre le texte généré et le texte de référence, en se concentrant sur les n-grams partagés.

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** : est spécifiquement conçu pour évaluer les tâches de résumé. Il mesure la similarité entre le texte généré et le résumé attendu en se basant sur le chevauchement de phrases ou de mots.

La formule générale de ROUGE est donnée par :

$$\text{ROUGE} = \frac{\text{Nombre de mots/phrases communs}}{\text{Nombre total de mots/phrases dans le résumé de référence}} \tag{2.12}$$

ROUGE met laccent sur le rappel (recall), mesurant combien de contenu pertinent du résumé de référence est capturé dans le texte généré.

Lorsque les métriques automatiques ne parviennent pas à saisir toute la finesse dun texte produit par un LLM, lévaluation humaine devient une méthode complémentaire, voire essentielle. Elle permet notamment de :

- Apprécier la qualité globale du texte généré, en termes de fluidité, de cohérence et de pertinence.
- Évaluer la capacité du modèle à comprendre et restituer correctement le contexte.

- Repérer déventuels biais ou erreurs contextuelles que les outils automatisés pourraient ne pas détecter.

Récemment, lévaluation assistée par modèle (G-Eval) est une solution innovante qui utilise des LLMs, comme GPT-4, pour évaluer les textes générés par dautres modèles. Il repose sur la génération de chaînes de pensée (CoT) pour définir des étapes dévaluation détaillées, suivies dun système de notation formaté en remplissage de formulaire. Ce processus permet une évaluation plus fine et corrélée aux jugements humains, bien que des biais potentiels, comme une préférence pour les textes générés par LLMs, nécessitent une attention particulière[12].

En résumé, les méthodes automatiques telles que la perplexité, BLEU et ROUGE permettent dévaluer quantitativement les performances des modèles de langage. Lévaluation humaine apporte une perspective subjective, tandis que G-Eval exploite GPT-4 pour générer des métriques adaptées à chaque tâche, alignées sur les attentes humaines.

2.5 Domaines d'application

Lintégration des LLM dans diverses tâches en aval est devenue une tendance majeure, tant dans les communautés de recherche en intelligence artificielle que dans les industries. Bien que chaque domaine présente des défis uniques, la polyvalence des LLMs leur permet de fournir des contributions significatives grâce à leur capacité dadaptation[10].

Les LLM sont perçus comme des outils universels capables de comprendre, générer et manipuler du texte humain de manière contextuellement pertinente. Leurs applications couvrent un large éventail de tâches allant de la traduction linguistique et de la rédaction assistée à des opérations plus complexes comme la synthèse de documents, la génération de code ou l'analyse de grandes masses de données textuelles. Toutefois, leur performance dépend fortement de la qualité des données d'entrainement. Dans le domaine de l'éducation, les LLM transforment les pratiques en offrant des solutions polyvalentes : pour les étudiants, ils fournissent des contenus personnalisés et des exercices adaptés ; pour les enseignants, ils facilitent la création de plans de cours, la correction automatisée et le développement de matériel pédagogique inclusif. En apprentissage des langues, ils permettent des conversations réalistes et des corrections grammaticales, tout en améliorant l'accessibilité pour les malentendants ou les apprenants en difficulté.

Ainsi, les LLM répondent à des besoins variés et redéfinissent progressivement les contours de l'éducation moderne[10].

2.6 Exemples populaires de grands modèles de langage

Ces dernières années, l'évolution rapide des LLM a donné naissance à un écosystème diversifié. Cette analyse met en lumière les principaux modèles phares récemment développés, en se concentrant sur leurs caractéristiques générales, leur accessibilité, ainsi que les tendances actuelles observées depuis 2022 jusqu'à 2025[10]:

Table 2.1: LLM les plus populaires

Modèle	**Développeur**	**Année de lancement**	**Open Source ou Fermé**
PaLM	Google AI	2022	Fermé
BLOOM	Hugging Face	2022	Open Source
Gemini	Google AI	2022	Fermé
GPT-4	OpenAI	2023	Fermé
Claude	Anthropic	2023	Fermé
Quen	Alibaba Cloud	2023	Fermé
DeepSeek	DeepSeek Lab	2023	Open Source
Mistral	Mistral AI	2023	Open Source
ChatGLM	Zhipu AI	2023	Open Source
Llama 3	Meta	2024	Open Source
Falcon 2	TII	2025	Open Source

Parmi ces modèles, une diversité dapproches techniques et stratégiques se dégage, allant des architectures multimodales aux solutions optimisées pour lefficacité énergétique ou le déploiement local. Le tableau suivant présente une comparaison détaillée de ces modèles emblématiques, en mettant en avant leurs forces et limites techniques, ainsi que leur adaptabilité à différents contextes dutilisation. Cette analyse permet de saisir les dynamiques concurrentielles et collaboratives qui façonnent le paysage des LLMs à lorée de 2025[10]:

Table 2.2: Comparaison des modèles de langage avancés

Modèle	Avantages techniques	Inconvénients techniques
PaLM	✓ Très bonne compréhension linguistique ✓ Fort en raisonnement logique	✗ Modèle fermé ✗ Peu accessible
BLOOM	✓ Open source complet ✓ Multilingue (46 langues) ✓ Idéal pour recherche collaborative	✗ Moins performant sur tâches complexes
Gemini	✓ Multimodal (texte, image, audio) ✓ Fort raisonnement ✓ Intégration Google Workspace	✗ Modèle fermé ✗ Peu documenté
GPT-4	✓ Excellente performance générale ✓ Fort en dialogue ✓ Multimodal (GPT-4V)	✗ Modèle fermé ✗ Coût API élevé
Claude	✓ Contexte long (100k tokens) ✓ Sécurité renforcée ✓ Assistance conversationnelle	✗ Modèle fermé ✗ Accès géo-limité
Quen	✓ Performant sur textes chinois ✓ Intégré écosystème Alibaba	✗ Modèle fermé ✗ Limité hors Chine
DeepSeek	✓ Open source ✓ Fort en mathématiques ✓ Bon raisonnement	✗ Communauté limitée ✗ Peu benchmarké
Mistral	✓ Open source ✓ Optimisé vitesse ✓ Utilisable localement	✗ Pas multimodal ✗ Moins polyvalent
ChatGLM	✓ Open source ✓ Optimisé chinois ✓ Faible mémoire requise	✗ Faible sur langues européennes
LLaMA 3	✓ Open source ✓ Proche GPT-4 ✓ Bon pour fine-tuning	✗ Requiert ressources importantes
Falcon 2	✓ Open source ✓ Multilingue ✓ Bon en classification	✗ Support en développement

2.7 Inconvénients et challenges

Les LLM, comme GPT-4, ont transformé le NLP mais posent plusieurs défis majeurs : coût de calcul élevé, robustesse limitée, et difficulté dinterprétabilité. À mesure quils évoluent pour des tâches plus complexes, des problèmes dévolutivité, de confidentialité et de traitement en temps réel apparaissent. Les recherches explorent actuellement lintégration de la multimodalité, lapprentissage par transfert et lapprentissage continu pour adapter ces modèles à de nouvelles informations. Ces défis soulignent les complexités techniques et les enjeux futurs des LLM dans des applications concrètes[10].

- Coût computationnel : le coût de calcul pour entraîner les LLM est élevé, augmentant les dépenses et posant des problèmes environnementaux liés à la consommation énergétique. Bien que les performances saméliorent avec plus de ressources, ce gain diminue progressivement en raison de la loi des rendements décroissants, surtout lorsque la taille du modèle et des données reste constante.

- Biais et Équité : les LLM peuvent reproduire et amplifier les biais sociétaux présents dans leurs données dentraînement. Ces biais se reflètent souvent dans les résultats, soulevant des questions éthiques et des problèmes déquité.

- Surapprentissage : les LLM, malgré leurs capacités dapprentissage avancées, peuvent surapprendre des schémas parasites présents dans leurs données dentraînement, générant ainsi des réponses illogiques. Le défi réside dans léquilibre entre mémorisation, qui permet de reproduire des détails spécifiques, et généralisation, essentielle pour traiter des entrées inédites. Un excès de mémorisation risque de provoquer un surapprentissage, rendant le modèle rigide face à de nouvelles situations.

- Inégalité Économique et de Recherche : le coût élevé de formation et de déploiement des LLM risque de limiter leur développement aux grandes organisations bien financées, exacerbant ainsi les inégalités économiques et en matière de recherche en IA.

- Hallucinations : les LLM peuvent produire des ń hallucinations ż, des réponses incorrectes ou non alignées avec les données fournies, bien quelles semblent plausibles. Ces hallucinations se manifestent de trois manières : elles peuvent contredire les informations

fournies par les utilisateurs, être incohérentes avec le contexte ou les réponses précédemment générées, ou encore aller à lencontre des connaissances mondiales établies.

- Ingénierie des Prompts : les prompts, qui servent dentrées aux LLM, jouent un rôle essentiel dans la qualité des réponses grâce à leur syntaxe et leur sémantique. Des variations, parfois inattendues pour les humains, peuvent significativement influencer les résultats. Lingénierie des invites consiste à concevoir des requêtes en langage naturel pour guider efficacement les réponses des modèles.

- Connaissances limitées : les informations acquises lors du pré-entraînement peuvent devenir obsolètes avec le temps, et un ré-entraînement complet est coûteux. Pour garantir des réponses factuelles précises, on utilise des pipelines daugmentation de la récupération (comme RAG). Cependant, les modèles pré-entraînés ne sont pas directement optimisés pour ce type dapproche, nécessitant des adaptations spécifiques du pipeline dentraînement.

- Oubli Catastrophique : les LLM sont généralement pré-entraînés sur de vastes ensembles de données, puis affinés sur des données spécifiques à un domaine pour réduire les coûts de formation. Cependant, ils rencontrent des défis comme ladaptation au domaine et loubli catastrophique, qui compromettent la rétention des connaissances acquises lors de lapprentissage de nouvelles tâches.

- Interprétabilité et Explicabilité : la nature ń boîte noire ż des LLM complique la compréhension de leurs décisions, essentielle pour instaurer confiance et acceptation, notamment dans les domaines sensibles. Malgré leurs performances, ce manque de transparence limite leur fiabilité. Des efforts sont menés pour les rendre plus explicables, renforcer la confiance et assurer leur alignement avec les valeurs humaines et normes juridiques.

2.8 Stratégies d'adaptation des LLMs

Bien que les LLMs soient puissants, ils présentent des limites, notamment leur dépendance aux données dentraînement et leur manque de flexibilité dans des contextes spécifiques. Pour

répondre à ces défis, des approches innovantes ont été développées, telles que le RAG, qui enrichit les modèles avec des informations externes, le finetuning, qui les adapte à des tâches précises, et le prompt engineering, qui optimise leurs réponses via des instructions bien conçues.

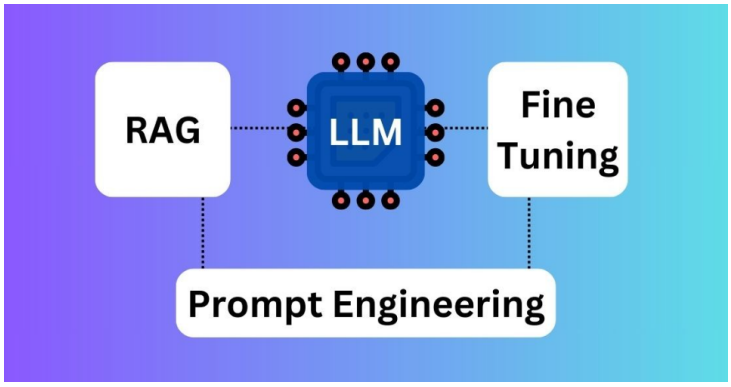

Figure 2.3: FineTuning vs. Prompting vs. RAG [19].

2.8.1 Finetuning

Le finetuning **(Ajustement fin)** des modèles de langage de grande taille consiste à ajuster les paramètres de ces modèles afin quils soient spécialisés sur des tâches déterminées ou sur un domaine donné. Même si la plupart des LLM actuels comme ChatGPT, Mistral ou Llama sont conçus pour être aussi généralistes que possible, ils naccèdent pas pour autant à une certaine spécialisation sur des sujets particuliers.

Lintérêt du finetuning est donc de tirer parti des grandes performances de ces LLM pour les appliquer sur un domaine précis, combinant alors efficience et personnalisation. Le LLM a alors une bien meilleure connaissance dun domaine spécifique, ce qui permet déviter de devoir créer son LLM dans un secteur particulier (assurance, pharmacie ou domaine de l'éducation, par exemple)[20].

Figure 2.4: Processus de finetuning[20].

En pratique, le finetuning se présente comme une solution indiquée lorsque lon veut accorder à un LLM générique une capacité de raisonnement spécialisée sur un domaine particulier, ou adoptant un vocabulaire de choix, lui étant propre.

L'ajustement fin des LLM est un processus d'apprentissage supervisé qui utilise un ensemble de données d'exemples étiquetés pour mettre à jour les pondérations des LLM et améliorer la capacité du modèle à effectuer des tâches spécifiques. Certaines méthodes les plus efficaces pour le finetuning sont à explorer :

Méthode 1: Instruction finetuning

Affinement des instructions est une méthode dadaptation dun modèle, consistant à apprendre à partir dexemples structurés en paires instruction-réponse. Ces exemples sont sélectionnés pour atteindre un objectif spécifique (ex. : générer un résumé ou traduire un texte). Le modèle apprend ainsi à répondre aux instructions en sinspirant des exemples du corpus utilisé pour son affinage[20].

Méthode 2: Full finetuning

Le finetuning complet consiste à ajuster tous les paramètres du modèle (ses poids) pour ladapter à une tâche spécifique. Ce processus aboutit à une version entièrement mise à jour du modèle, avec des paramètres optimisés pour la nouvelle cible. Cependant, comme lors du pré-entraînement, cette méthode nécessite des ressources conséquentes (mémoire GPU/TPU, puissance de calcul) pour gérer les gradients, les optimiseurs et les autres éléments mis à jour

pendant lentraînement[20].

Méthode 3: Parameter-efficient finetuning (PEFT)

Les méthodes dadaptation paramétrique efficace, telles que LoRA , QLoRA , Adapters Layers , et Prefix/Prompt Tuning, visent à ajuster les LLMs avec un coût computationnel et une empreinte mémoire réduits. Ces techniques modifient uniquement une fraction des paramètres du modèle initial :

- **LoRA** insère des matrices de faible rang pour approximer les mises à jour des poids;
- **QLoRA** combine quantification et LoRA pour optimiser davantage les ressources;
- **Adapters** ajoutent de petites couches légères entre les modules existants;
- **Prefix/Prompt Tuning** introduit des tokens apprenables en début de séquence pour guider la génération.

Ces approches permettent de spécialiser les LLMs pour des tâches spécifiques tout en préservant la majorité des paramètres préentraînés, rendant le finetuning accessible même avec des modèles extrêmement volumineux (ex : Llama 65B). Elles sont particulièrement utiles dans des contextes à ressources limitées ou pour éviter le surapprentissage sur de petits jeux de données[20].

Reinforcement learning from human feedback (RLHF)

Lapprentissage par renforcement à partir du feedback humain est une approche innovante qui vise à améliorer les modèles de langage en intégrant des retours humains au processus dentraînement. Grâce à cette méthode, les modèles apprennent à produire des réponses plus précises et adaptées au contexte en sajustant aux préférences et corrections fournies par des évaluateurs humains. En exploitant lexpertise humaine, le RLHF permet non seulement une optimisation continue des performances du modèle, mais aussi une adaptation dynamique aux exigences réelles des utilisateurs, garantissant ainsi une efficacité accrue et une meilleure fidélité aux attentes humaines[20].

Cependant, le finetuning présente des limites. Si la tâche ou le domaine cible diffèrent significativement des données de préentraînement, le modèle peut avoir du mal à s'adapter efficacement. De plus, l'ajustement fin peut entraîner des oublis catastrophiques : le modèle oublie une partie de ses connaissances générales au profit de la tâche spécialisée.

2.8.2 Prompt engineering

Lingénierie des prompts consiste à concevoir et structurer soigneusement les questions ou exemples fournis à un modèle de langage pour obtenir les réponses ou comportements souhaités. Cette méthode repose sur le fait que la formulation et lorganisation des prompts influencent directement les résultats générés par le modèle[19].

Certaines techniques dingénierie rapides courantes incluent :

- **Zero-Shot learning :** donner une tâche au modèle sans fournir d'exemples.
- **Few-shot learning :** fournir few examples ou prompts démontrant le format ou les schémas de sortie attendus, permettant au modèle de sadapter à partir de ces exemples.
- **Chain-of-Thought (CoT) :** les prompts en chaîne de pensée aident les LLM à réussir à résoudre des tâches complexes, à la fois en choisissant la manière de formuler l'invite et en fournissant au modèle le temps ou les étapes nécessaires pour générer la bonne réponse.
- **Prompt templates :** créer des structures de modèles d'invite réutilisables et flexibles, facilement adaptables à diverses tâches ou entrées.
- **Prompt mining :** identifier de manière systématique des invites ou combinaisons efficaces pour générer les résultats souhaités.
- **Augmentation contextuelle (contextual augmentation) :** cette méthode enrichit les invites avec des informations contextuelles pour augmenter la pertinence des réponses.

Lingénierie des prompts sest révélée précieuse pour orienter les modèles linguistiques vers des tâches ou des comportements spécifiques, sans nécessiter de réentraînement ou dajustement approfondi. Cependant, ce processus peut être long et itératif, car la recherche des prompts optimaux repose souvent sur des essais et erreurs.

2.8.3 RAG (Retrieval-Augmented Generation)

Le RAG est une technique qui améliore les réponses des modèles en intégrant des informations provenant de documents externes[23]. Elle se déroule en deux étapes principales :

- **Récupération** : Le modèle utilise une requête, généralement basée sur la question ou la tâche de l'utilisateur, pour rechercher et extraire des informations pertinentes dans une base de données. Cette base peut inclure des textes variés, comme des livres, des articles ou des sites web.
- **Génération augmentée** : les informations récupérées sont ensuite utilisées comme contexte supplémentaire pour générer une réponse plus précise, informative et fondée sur des données concrètes, au lieu de se limiter aux connaissances acquises lors de l'entraînement.

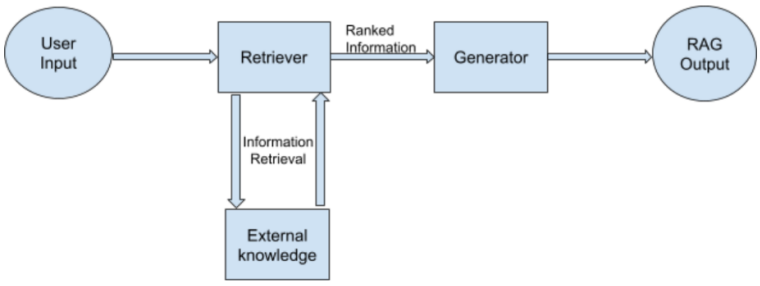

Figure 2.5: Flux de base du système RAG avec ses composants[22].

L'approche RAG est particulièrement adaptée pour des tâches nécessitant des réponses détaillées ou des informations spécifiques qui ne figurent pas nécessairement dans le corpus d'entraînement du modèle. Elle enrichit les requêtes pour améliorer la qualité des réponses générées, mais n'a pas d'impact sur les capacités inhérentes de raisonnement du LLM[21] :

1. Mécanismes de récupération dans le système RAG

Dans les systèmes RAG, l'outil de récupération est essentiel pour extraire des documents pertinents d'un corpus externe. Une récupération efficace garantit que les résultats du modèle reposent sur des informations précises. Les mécanismes de récupération sont classés en quatre catégories selon leurs méthodes et leurs sources: Sparse Retrieval, Dense Retrieval, Internet Retrieval, and Hybrid Retrieval[22].

a. **Sparse Retriever :** méthode historique pour la gestion des connaissances, se distingue par sa simplicité algorithmique et sa compatibilité avec les systèmes d'indexation classiques. En s'alignant sur le raisonnement humain, il assure une grande adaptabilité.

b. **Dense Retriever :** utilise une architecture de double encodeur pour générer des plongements denses (requêtes/documents). Il inclut trois approches : Word Embedding, Multimodal Retrieval et Data Distillation.

c. **Internet Retriever :** (ex. Bing) propose une approche "plug-and-play", évitant les mises à jour en temps réel mais nécessitant un filtrage strict contre les données erronées. Méthodes comme Komeili et al. (2021) génèrent des URLs via des APIs de recherche, croisant ces liens avec des données publiques ou Wikipedia pour améliorer la fiabilité des résultats.

d. **Hybrid Retriever :** combine stratégiquement sparse et dense pour renforcer lefficacité et la robustesse de LLM (Lazaridou et al., 2022). Des approches comme Hu et al. (2023) intègrent un filtrage basé sur le dual-encoder et les poids des termes, éliminant les contenus non pertinents. Boytsov et al. (2016) hybrident kNN et BM25 avec un modèle de traduction pour capturer les relations sémantiques entre termes. Ces méthodes compensent les limites individuelles, optimisant ainsi la précision dans des contextes variés.

2. Mécanismes de génération dans le système RAG

Dans les systèmes RAG, le mécanisme du générateur intègre les informations récupérées par le retriever avec la requête utilisateur pour produire des réponses cohérentes et contextuellement

adaptées. Appuyé sur un LLM, il garantit la fluidité, la précision et lalignement du texte généré avec les données externes et la requête initiale[22] :

a. **T5 (Text-to-Text Transfer Transformer) :** T5 est un modèle central dans les systèmes RAG , convertissant toutes les tâches NLP en format texte-à-texte. Il permet un fine-tuning versatile (réponse à questions, résumé) et surpasse GPT-3 et BART sur des benchmarks comme Natural Questions . Sa capacité à gérer des tâches multi-objectifs en fait un choix clé pour les systèmes RAG nécessitant une gestion complexe des connaissances.

b. **BART (Bidirectional and Auto-Regressive Transformer) :** BART est un modèle génératif intégré aux systèmes RAG pour sa capacité à traiter des données bruitées (résumé, QA) via une architecture de denoising autoencoder . Associé à un retriever , il améliore la précision factuelle des réponses en sappuyant sur des connaissances externes. Ses variantes RAG dominent des benchmarks dans des tâches comme la génération de dialogues et le résumé dactualités.

3. Différentes modalités de génération

a. **Modèles RAG basés sur du texte :** s'appuient sur des architectures Transformer (BERT , T5) pour intégrer retrieval et génération de texte. Le retrieval dense surpasse les méthodes sparces (TF-IDF) grâce à des représentations sémantiques. Des modèles comme REALM unifient les deux phases en un pipeline end-to-end , optimisant la précision et la fluidité des réponses.

b. **Modèles RAG basés sur l'audio :** appliquent lapproche RAG à laudio via des embeddings générés par Wav2Vec 2.0 . Ils prennent en charge des applications comme la reconnaissance vocale et les assistants conversationnels en intégrant retrieval et génération à partir de données audio.

c. **Modèles RAG basés sur la vidéo :** utilisent des embeddings vidéo (ex. I3D , TimeSformer) pour intégrer des données visuelles et textuelles, améliorant des tâches comme la génération de légendes et le retrieval grâce à la capture des dynamiques temporelles et spatiales.

d. **Modèles RAG multimodaux :** fusionnent texte, audio, vidéo et images pour des applications comme la génération dimages ou le retrieval croisé entre modalités. Des modèles comme Flamingo et des approches comme "retrieval as generation" (Wang et al., 2024) exploitent des bases dimages annotées pour accélérer la création visuelle et permettre des dialogues basés sur des entrées multimodales[22].

4. Paradigmes de recherche RAG

Le paradigme de recherche RAG est en constante évolution et se classe en trois étapes distinctes : RAG naïf, RAG avancé et RAG modulaire[23]. La Figure 2.6 illustre cette progression.

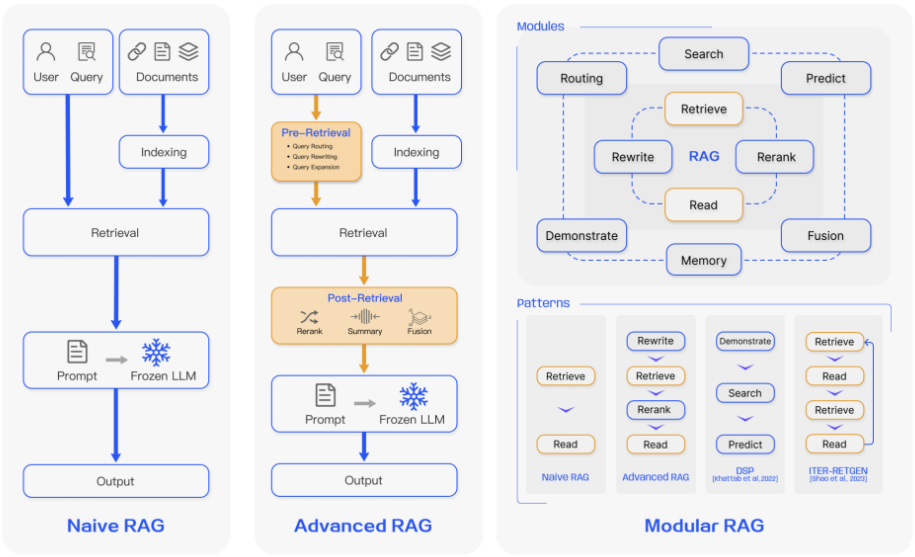

Figure 2.6: Trois paradigmes du RAG[23].

a. **RAG naïf (naive RAG)** : le RAG Naïf est la première méthodologie, basée sur un processus en trois étapes : indexation, récupération et génération, souvent appelé *Retrieve-Read*.

- Indexation : les données brutes (PDF, HTML, etc.) sont nettoyées, segmentées en petits blocs, converties en vecteurs via un modèle d'encodage, et stockées dans une base de données vectorielle pour permettre des recherches efficaces.
- Récupération : une requête utilisateur est transformée en vecteur, puis comparée aux segments indexés pour récupérer les K plus pertinents, utilisés comme contexte étendu.

- Génération : la requête et les documents sélectionnés sont combinés pour formuler une réponse cohérente à l'aide d'un LLM.

b. RAG avancé (advanced RAG) : le RAG Avancé apporte des améliorations spécifiques pour surmonter les limitations du RAG Naïf, en se concentrant sur la qualité de la récupération grâce à des stratégies pré-recherche et post-recherche.

- **Pré-recherche :** cette étape optimise la structure dindexation et la requête initiale. Lobjectif est daméliorer la qualité des données indexées (granularité fine, métadonnées, segmentation) et de clarifier la requête utilisateur via des techniques comme la reformulation ou lexpansion de la question.
- **Post-recherche :** une fois le contexte pertinent récupéré, il est crucial de lintégrer efficacement. Les principales méthodes incluent :
  - Re-classement : réorganiser les segments pour prioriser les plus pertinents.
  - Compression du contexte : sélectionner les informations essentielles, réduire la redondance et maintenir le focus sur les détails clés.

c. RAG modulaire (modular RAG) : le RAG modulaire repousse les limites des approches Naïf et Avancé en introduisant une architecture flexible et adaptable.Il intègre de nouveaux modules spécialisés (recherche, mémoire, routage, prédiction) et permet leur reconfiguration pour répondre à des besoins spécifiques. Cette modularité supporte des stratégies hybrides, des interactions dynamiques entre modules et une intégration facile avec dautres technologies comme le fine-tuning. Comparé au fine-tuning, qui adapte profondément un modèle mais nécessite des ressources importantes, le RAG excelle dans les environnements dynamiques grâce à son accès en temps réel à des sources externes, bien quil puisse être plus lent et soulever des questions éthiques. Les deux approches peuvent se compléter pour optimiser les performances selon les besoins dune application donnée.

5. Processus RAG

Cependant, pour mettre en place un système RAG, il est essentiel de suivre une série d'étapes clés qui permettent de combiner efficacement la récupération d'informations et la génération de réponses[21] :

Étape 1 : Collecte des données

La première phase de la Génération Augmentée par Récupération consiste à collecter des informations pertinentes issues de multiples sources (bases de données, articles scientifiques, sites web, etc.). Elle repose sur la définition de requêtes précises, conçues pour extraire des données ciblées et garantir leur cohérence avec le contexte de la tâche et les objectifs du modèle dIA. Cette étape clé assure une base solide pour les étapes suivantes, en alignant les connaissances externes sur les besoins spécifiques de lapplication.

Étape 2 : Fragmentation des données (Chunking)

Le chunking consiste à découper un texte long ou un document en segments plus petits et facilement exploitables, appelés **chunks**. Chaque chunk constitue un segment logique et autonome du texte, pouvant être analysé de manière isolée, et centré sur un thème précis. Lorsquune information est extraite de la source, elle devient ainsi directement pertinente pour répondre à une requête utilisateur, en limitant linclusion dinformations extrinsèques ou redondantes issues de lensemble du corpus. Cette approche améliore lefficacité du traitement et la pertinence des résultats.

Étape 3 : Vectorisation des documents – Embeddings

Après avoir fragmenté les données en parties plus petites, elles doivent être converties en représentations vectorielles. Cette étape transforme le texte en embeddings, des formats numériques qui capturent la signification sémantique du contenu.

En résumé, les embeddings transforment les mots ou phrases en vecteurs numériques dans un espace à haute dimension, capturant leurs significations sémantiques et relations syntaxiques. Des mots proches sémantiquement se retrouvent ainsi géométriquement voisins dans cet espace, permettant de modéliser facilement analogies et associations linguistiques via des calculs vectoriels.

Étape 4 : Indexation des embeddings dans une base de données vectorielle

Une base de données vectorielle est un système spécialisé conçu pour stocker des informations sous forme de vecteurs à haute dimensionnalité, représentant des caractéristiques ou propriétés spécifiques. Ces vecteurs peuvent comporter des dizaines à des milliers de dimensions, selon la complexité et la richesse des données traitées.

Les données, quil sagisse de textes, dimages, de fichiers audio ou vidéo, sont converties en vecteurs numériques grâce à des modèles dapprentissage automatique, des embeddings linguistiques ou des méthodes dextraction de caractéristiques.

Contrairement aux bases de données classiques, qui organisent les données scalaires (nombres, textes simples) en lignes et colonnes, les bases vectorielles manipulent directement des vecteurs. Cette approche modifie en profondeur leur architecture doptimisation et leurs techniques de requêtage, permettant notamment des recherches par similarité sémantique ou des analyses basées sur des proximités géométriques dans lespace multidimensionnel.

Étape 5 : Traitement des requêtes utilisateur

Lorsqu'une requête utilisateur est reçue, elle est convertie en une représentation vectorielle (embedding) à l'aide du même modèle utilisé pour les documents, assurant ainsi une compatibilité entre les deux.

Une fois la requête transformée, le système compare son embedding avec ceux des fragments de documents. Il identifie les segments les plus pertinents en mesurant leur similarité, à l'aide de techniques comme la similarité cosinus ou la distance euclidienne. Ces fragments sélectionnés sont alors considérés comme les plus adaptés pour répondre à la requête de l'utilisateur.

Étape 6 : Génération de réponses avec un LLM

Les fragments de texte extraits, combinés à la requête initiale de l'utilisateur, sont transmis à un modèle de langage. Ce dernier utilise ces éléments pour produire une réponse claire et cohérente, délivrée à l'utilisateur via l'interface de chat.

6. Évaluation des systèmes RAG

Dans les systèmes de type RAG, l'évaluation se structure autour de deux phases clés : la récupération de documents pertinents et la génération de réponses à partir de ces documents. Ces étapes nécessitent des métriques adaptées pour mesurer à la fois la qualité du retrieval (pertinence des documents) et celle de la generation (fidélité et cohérence des réponses). Les métriques principales utilisées dans ce cadre, sont[24]:

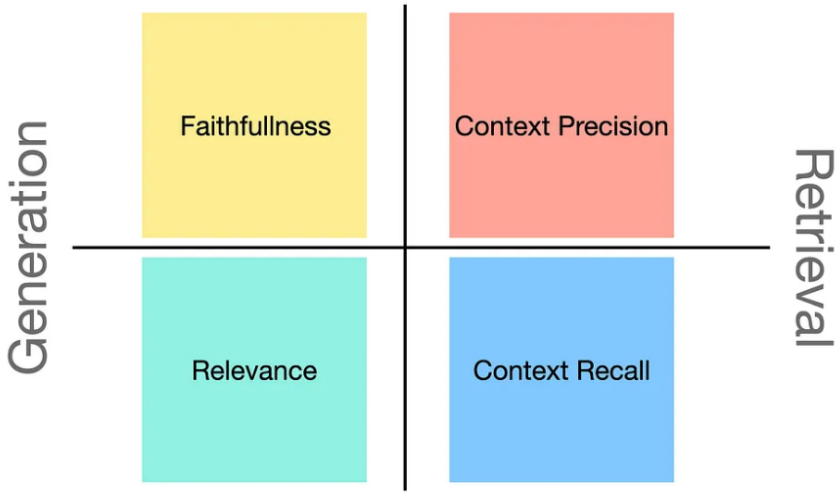

Figure 2.7: Métriques d'évaluation du RAG[25].

6.1 Mesures de récupération (Retrieval Metrics)

Context Precision @K : mesure la proportion des documents pertinents globaux qui figurent parmi les **K premiers documents récupérés**.

Formule :

$$\text{ContextPrecision@}K = \frac{|\mathscr{R}_K \cap \mathscr{D}_{\text{relevant}}|}{K} \tag{2.13}$$

Où :

- $\mathscr{R}_K$: Ensembles des K premiers documents récupérés.

- $\mathscr{D}_{\text{relevant}}$: Ensemble des documents pertinents.

Plus K est petit, plus cette métrique est sensible à la précision immédiate des résultats.

Context Recall @K : mesure la proportion des **documents pertinents globaux** qui figurent parmi les **K premiers résultats**. Elle indique donc si le système réussit à capturer une grande partie des documents utiles dès les premières positions.

Formule :

$$\text{ContextRecall@}K = \frac{|\mathscr{R}_K \cap \mathscr{D}_{\text{relevant}}|}{|\mathscr{D}_{\text{relevant}}|} \tag{2.14}$$

Où :

- $\mathscr{R}_K$: Ensembles des K premiers documents récupérés.

- $\mathscr{D}_{\text{relevant}}$: Ensemble des documents pertinents.

Une valeur élevée signifie que le système couvre efficacement les documents pertinents dans ses premiers résultats.

6.2 Mesures de génération (Generation Metrics)

- Faithfulness (Fidélité) (Réponse $\leftrightarrow$ Documents pertinents) : examine si la réponse générée s'appuie correctement sur les informations présentes dans les documents récupérés. Elle mesure ainsi le niveau de cohérence et dalignement entre la réponse produite et les sources documentaires utilisées.

- Relevance (Pertinence) (Réponse $\leftrightarrow$ Requête) : évalue dans quelle mesure la réponse générée correspond à lintention et au contenu de la requête initiale. Elle assure que la réponse traite effectivement du sujet demandé et satisfait les attentes exprimées dans la requête.

7. Les défis et limites actuels du RAG

Cette section vise à mettre en évidence les défis et les limites actuels du RAG compte tenu de lévolution récente des architectures et des méthodes utilisées du système, ce qui permettra d'orienter les recherches futures dans ce domaine[23].

- Efficacité et passage à léchelle : les modèles RAG peinent à sadapter aux grandes masses de données dynamiques, en raison de leur dépendance à des bases externes. Cela nécessite des algorithmes de recherche performants, mais aussi des ressources élevées en calcul et en mémoire, limitant leur utilisation en temps réel ou sur des systèmes peu puissants.

- Pertinence des documents récupérés : la qualité des réponses générées dépend directement de la pertinence des informations extraites. Or, les moteurs de recherche utilisés peuvent parfois renvoyer des contenus obsolètes ou hors sujet, surtout dans les cas de génération complexe ou longue.

- Biais et équité : les biais présents dans les sources de données peuvent être amplifiés par le système RAG, entraînant des réponses discriminatoires. Il est donc essentiel de mettre en place des stratégies pour atténuer ces biais tout au long du processus.

- Cohérence du contenu généré : malgré leurs capacités, les modèles RAG peuvent intégrer imparfaitement les connaissances récupérées, ce qui peut produire des réponses incohérentes ou erronées notamment des hallucinations factuelles.

- Interprétabilité et transparence : le fonctionnement des systèmes RAG reste souvent opaque, ce qui limite leur adoption dans des domaines critiques comme la santé ou le droit. Développer des outils dexplicabilité est donc primordial pour renforcer la confiance des utilisateurs.

2.8.4 RAG vs Fine-Tuning vs prompt engineering:

Les trois techniques (finetuning, prompt engineering et RAG) sont des méthodes pour adapter les LLMs à des tâches spécifiques. Elles entraînent essentiellement le LLM à mieux performer dans un domaine particulier (Table 2.3) : la meilleure technique dépend des besoins du projet, mais généralement, pour :

- Des tâches simples avec des ressources limitées, le prompt engineering est le choix optimal.

- Lorsque la précision et la personnalisation sont cruciales, le fine-tuning savère nécessaire.

- Tandis que, quand on a besoin dun équilibre entre performance et efficacité, le RAG offre une solution intéressante.

Table 2.3: Tableau résumant les principales différences entre ces trois techniques

Technique	Description	Forces	Limitations
RAG	Combine la génération de modèle linguistique avec la récupération de sources de données externes	Exploite les connaissances externes, améliore la précision factuelle	Nécessite une source de données externe, peut introduire des informations non pertinentes.
Fine-tuning	Entraînement supplémentaire d'un modèle linguistique préentraîné sur une tâche ou un ensemble de données spécifique	S'adapte à des domaines/tâches spécialisés, exploite les connaissances préentraînées, efficace sur le plan informatique	Potentiel d'oubli catastrophique, limité par la distribution des données de préentraînement
Prompt Engineering	Conception minutieuse des invites pour guider le comportement du modèle linguistique	L'orientation du modèle sans réentraînement, flexible, peut être combinée avec d'autres techniques	Prend du temps, nécessite des essais et des erreurs, peut ne pas bien se généraliser.

2.9 Travaux relatifs

Avec les avancées récentes de l'IA, plusieurs travaux de recherche soulignent l'évolution des techniques d'amélioration des LLM, en mettant particulièrement l'accent sur l'ingénierie des prompts, le réglage fin et la RAG. L'ingénierie rapide a gagné en popularité en tant que méthode permettant d'améliorer les performances LLM dans diverses tâches NLP. La contribution dans [26] fournit une étude complète des techniques adaptées à différentes applications et souligne sa polyvalence et son adoption généralisée.

Dans [27], les auteurs soulignent que le RAG peut améliorer considérablement les systèmes de questions-réponses médicales en intégrant des bases de connaissances externes ; cependant, ils reconnaissent également les limites des scénarios complexes nécessitant plusieurs cycles de recherche d'information, qu'ils abordent par une approche RAG itérative (i-MedRAG). Ce processus itératif permet aux LLM de poser des questions de suivi en fonction des interactions précédentes, affinant ainsi le processus de récupération et améliorant la pertinence des réponses.

Dans le cadre de tâches spécifiques telles que Text-to-SQL [28], les auteurs examinent les méthodes d'ingénierie rapide parallèlement aux stratégies de réglage fin, illustrant les rôles

complémentaires que ces approches jouent dans l'amélioration de la précision du modèle.

Dans le contexte d'applications pratiques, les contributions dans [29] démontrent comment RAG peut être combiné avec des techniques et des agents de conversion de texte en SQL pour soutenir les flux de travail de gestion des contrats, en mettant l'accent sur le fait que la précision des réponses est améliorée grâce à l'orchestration dynamique des processus de récupération et de génération. De même,

Les contributions de [30] explorent l'optimisation des LLM sur les données financières et les réglementations du gouvernement indonésien pour soutenir la prise de décision, indiquant que l'ajustement spécifique à un domaine peut traiter efficacement des ensembles de données complexes et dynamiques.

L'étude dans [31] compare l'ingénierie rapide, le RAG et le réglage fin pour l'analyse de texte sur la santé mentale, révélant que si le réglage fin permet d'obtenir la plus grande précision, l'ingénierie rapide et le RAG offrent des alternatives plus flexibles et économes en ressources, avec des niveaux de performance modérés.

L'application du RAG est particulièrement importante pour relever les défis liés à la rareté des données et aux hallucinations, comme le démontrent les auteurs de [32] qui introduisent HDLCoRe, un cadre qui exploite la génération augmentée par récupération pour atténuer les hallucinations dans la génération de code HDL. Cette approche illustre comment RAG peut être intégré à l'ingénierie rapide pour améliorer la fiabilité des sorties LLM dans des domaines spécialisés.

De plus, le réglage fin reste une technique critique, en particulier lorsque de grands ensembles de données spécifiques à un domaine sont disponibles. Les auteurs de [33] montrent comment le réglage fin des LLM sur de vastes ensembles de données scientifiques, combiné à une ingénierie rapide et à de nouvelles mesures de perte comme la distance cosinus inverse (ICD), peut améliorer considérablement l'extraction de données structurées et la synthèse scientifique. Cela indique une tendance vers des approches hybrides qui exploitent à la fois le réglage fin et l'ingénierie rapide pour optimiser les performances du modèle pour des tâches de données complexes.

Le déploiement et le développement pratiques de systèmes basés sur LLM impliquent

également des considérations relatives à la conception rapide et à la mise au point. [34] explorent les perspectives des praticiens, en soulignant l'importance de comprendre les priorités et les défis du monde réel dans la création d'applications LLM, qui reposent souvent sur des techniques d'ingénierie rapides pour une adaptation et un déploiement rapide.

En résumé, la littérature reflète une interaction nuancée entre l'ingénierie rapide, le réglage fin et le RAG dans l'avancement des capacités LLM. Ensemble, ces stratégies forment une boîte à outils à multiples facettes pour optimiser les LLM dans diverses applications, explorant leur intégration et leurs adaptations spécifiques au domaine [26], [29], [33], [34], [32], [31].

2.10 Conclusion

Tout au long de ce chapitre, nous avons défini les fondements théoriques essentiels à la compréhension et à la mise en uvre des systèmes dIA générative. Nous avons présenté les LLMs, leurs capacités, ainsi que les méthodes clés permettant de les adapter et de les optimiser pour des applications spécifiques, telles que le finetuning et le prompt engineering. Par ailleurs, nous avons détaillé lapproche RAG, qui constitue une solution performante pour enrichir les réponses générées par les LLMs grâce à lintégration de connaissances externes.

Ces concepts constituent la base solide sur laquelle sappuie nos solutions pratiques développées dans le chapitre suivant. Ce dernier sera consacré à la présentation de nos implémentations établie sur lIA générative, illustrant comment ces technologies sont appliquées dans un cas réel dexploitation de données externes pour la génération de résultats attendues.

CHAPITRE 3

CONCEPTION ET DÉVELOPPEMENT DE LA SOLUTION

3.1 Introduction

Ce chapitre traite des défis précédemment évoqués liés aux limitations des LLM à fournir des réponses précises sur des informations académiques spécialisées. Pour y remédier, nous développons un système innovant intégrant les techniques de génération augmentée contextuellement et de fine-tuning, afin de permettre au modèle daccéder à des sources de connaissances externes et de générer des réponses plus cohérentes avec les contextes des requêtes du domaine éducatif. La méthodologie détaillée sera présentée, depuis la construction dune base de connaissances externe, en passant par lamélioration des performances du modèle grâce à son ajustement sur des données spécifiques, via une stratégie de recherche classique, de recherche augmentant la génération et de finetuning, jusquà lévaluation des résultats à laide de métriques standardisées, incluant la précision linguistique, la fiabilité et la préservation du contexte. Ces expérimentations apportent des solutions pratiques sur lefficacité de ces intégrations, contribuant à améliorer la qualité des résultats dans un domaine académique spécialisé.

3.2 Architecture de la solution

Cette section vise à expliquer de manière claire l'architecture générale du système et son mode de fonctionnement. La Figure 3.1 fournit un aperçu global des stratégies utilisées dans notre solution.

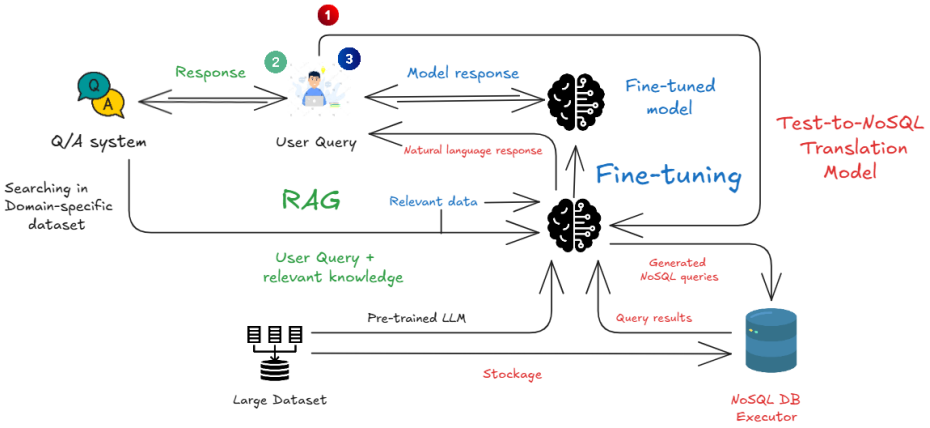

Figure 3.1: Architecture globale du système.

3.3 Description de l'ensemble de données

Cet ensemble de données contient des avis et des métadonnées de professeurs provenant des 100 meilleures universités des États-Unis, collectés sur RateMyProfessors.com[1]. Les données comprennent des informations détaillées sur les professeurs, les avis des étudiants et les métadonnées universitaires couvrant la période de 1999 à 2025. Son exploitation va répondre aux objectifs suivants :

- Permettre une analyse avancée par apprentissage automatique des modèles de retours des étudiants.
- Suivre l'évolution temporelle de l'efficacité pédagogique et de la satisfaction des étudiants.
- Fournir des données structurées pour la recherche et l'analyse pédagogiques.

[1] Kaggle Dataset - RateMyProfessor T100 Universities Reviews

- Créer un historique des indicateurs de performance pédagogique.
- Fournir des informations basées sur les données sur la qualité de l'enseignement supérieur.

Les caractéristiques importantes des données à explorer sont en deux gatégries :

Informations sur le professeur

- Nom et titre
- Département et établissement
- Note globale
- Nombre total d'avis
- Pourcentage de personnes prêtes à recommencer
- Score de difficulté

Détails de l'avis

- Notes individuelles des avis
- Horodatage des avis
- Informations sur le cours
- Note reçue (si fournie)
- Commentaires détaillés des étudiants
- Votes utiles/inutiles

3.4 Formulation du problème

3.4.1 Problème de la Recherche simple sémantique

Étant donné une requête Q_{NL} exprimée en langage naturel et un dataset D constitué de documents structurés dans une base de données NoSQL, l'objectif est de traduire cette requête en une syntaxe NoSQL exécutable pour faire retourner des réponses pertinents, a suivre :

$$R_{NL} = T_{NoSQL-to-NL}(R_{NoSQL}) \tag{3.1}$$

Où :

$$R_{NoSQL} = E_{NoSQL}(Q_{NoSQL}, D) \tag{3.2}$$

$$Q_{NoSQL} = T_{Text-to-NoSQL}(Q_{NL}) \tag{3.3}$$

Schéma global de la simple génération de réponses :

$$R_{NL} = T_{NoSQL-to-NL}\left(E_{NoSQL}\left(T_{Text-to-NoSQL}(Q_{NL}), D\right)\right) \tag{3.4}$$

Avec :

- Q_{NL} : Requête utilisateur en langage naturel,
- $T_{Text-to-NoSQL}$: Modèle de traduction texte $\rightarrow$ NoSQL,
- Q_{NoSQL} : Requête NoSQL générée,
- E_{NoSQL} : Exécuteur de requêtes NoSQL,
- D : Base de données cible,
- R_{NoSQL} : Résultats de la requête NoSQL,

- $T_{NoSQL-to-NL}$: Module de conversion NoSQL $\rightarrow$ langage naturel,
- R_{NL} : Réponse finale en langage naturel.

3.4.2 Problème de la génération guidée par Prompting (Zero/Few-shot)

Étant donné une requête q, un ensemble de documents pertinents $K_q \subset D$ obtenu par recherche sémantique, et une base de prompts P constituée d'instructions ou d'exemples (Zero-shot ou Few-shot), le second objectif est de guider un modèle de langage pré-entraîné M_{base} dans la génération d'une réponse appropriée.

La réponse est générée via :

$$R_{prompt}(q) = M_{base}(p_q) \tag{3.5}$$

Où, prompt enrichi est :

$$p_q = \text{BuildPrompt}(q, K_q, P) \tag{3.6}$$

Avec :

- K_q : ensemble des documents pertinents récupérés,
- P : ensemble de prompts ou gabarits (Zero/Few-shot),
- **BuildPrompt(.)** : fonction de construction de prompt à partir de q, K_q et P,
- M_{base} : modèle de langage pré-entraîné.

3.4.3 Problème de ladaptation au domaine par fine-tuning

Étant donné un ensemble de paires (requête, réponse) annotées et représentatives d'un domaine cible D spécifique en l'occurrence, l'enseignement supérieur et l'activité des enseignants le troisième objectif est d'adapter un modèle de langage généraliste M_{base} afin de renforcer sa spécialisation, sa pertinence lexicale et sa robustesse dans ce contexte spécifique.

Soient :

- $D_{\text{spécifique}} = \{(q_i, r_i)\}_{i=1}^{n}$: un corpus de paires requête/réponse dans un dom
- M_{base} : modèle de langage généraliste,
- M_{fine} : modèle adapté.

L'adaptation est réalisée par optimisation de la fonction de perte (ex. cross-entropy) :

$$M_{\text{fine}} = \arg\min_{\theta} \sum_{i=1}^{n} L(M_{\text{base}}(q_i; \theta), r_i) \tag{3.7}$$

où :

- θ sont les paramètres du modèle à optimiser,
- L est la fonction de perte supervisée.

3.5 Pipeline de la solution adoptée

Cette section présente de manière synthétique lorganisation de la solution, les approches et le pipeline mis en uvre. Elle explique comment les différentes composantes interagissent, depuis lacquisition des données jusquà leur analyse finale. La Figure 3.2 illustre lensemble du processus.

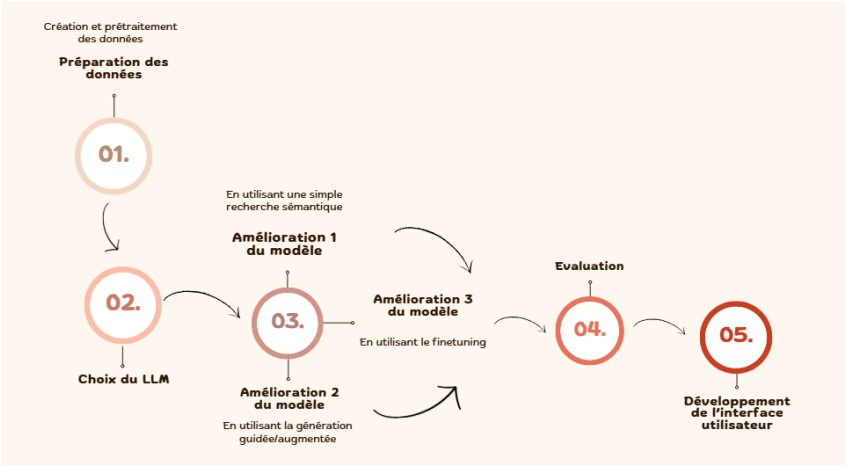

Figure 3.2: Schématisation de pipeline de la solution.

Pour garantir un développement méthodique et efficace de la solution, le processus passe par cinq étapes clés. Chaque étape joue un rôle crucial dans latteinte des objectifs finaux, depuis la préparation initiale des données jusquà la mise en uvre finale de linterface utilisateur. Voici un aperçu détaillé des étapes principales :

1. **Préparation de données :** initialement, nous avons collecté et prétraité des données issues de RateMyProfessor, incluant les commentaires et évaluations des étudiants sur leurs professeurs. Après nettoyage et structuration, ces données ont été utilisées pour guider et entraîner notre modèle.

2. **Choix du LLM :** face à la variété des modèles disponibles, nous avons sélectionné un LLM open source, librement téléchargeable et utilisable localement. Ce choix sest porté sur des critères daccessibilité, de simplicité dutilisation et de capacité à être déployé sans infrastructure lourde. Nous avons ensuite testé ses performances sur un échantillon de nos données afin dévaluer sa capacité à répondre de manière exacte et cohérente à nos besoins.

3. **Choix de la méthode damélioration du LLM :** pour optimiser les performances du modèle, trois approches complémentaires ont été adoptées, choisies pour leur efficacité et leur compatibilité avec nos contraintes techniques. La première, une simple recherche sémantique, génère des réponses pertinentes en fonction de la requête de lutilisateur. La seconde est la génération guidée, une extension de la génération classique de résultats, qui enrichit systématiquement les prompts avec des données externes pertinentes, assurant ainsi des réponses plus précises et mieux contextualisées. Ensemble, ces méthodes forment une solution flexible, évolutive et performante pour améliorer les capacités du modèle (LLM). La troisième est une méthode de finetuning qui ajuste uniquement des paramètres de faible rang, permettant dadapter le modèle avec peu de ressources tout en maintenant la qualité des générations.

4. **Évaluation des résultats :** une fois les méthodes mises en uvre, une évaluation rigoureuse et linterprétation des résultats obtenus sont cruciales pour déterminer lapproche la plus efficace et adaptée à nos besoins.

5. **Développement de linterface utilisateur :** nous allons concevoir linterface graphique qui facilitera linteraction avec les utilisateurs.

Dans ce travail, nous explorons une collection de stratégies pour optimiser l'efficacité et la pertinence des LLM. Ces approches permettent dintégrer des connaissances externes et dadapter les modèles à des domaines ciblés.

3.6 Implémentation de la solution

Dans cette section, nous détaillons et présentons les étapes parcourues pour la mise en pratique de ces solution, les résultats obtenus ainsi que linterface utilisateur :

3.6.1 Génération sémantique simple de réponses

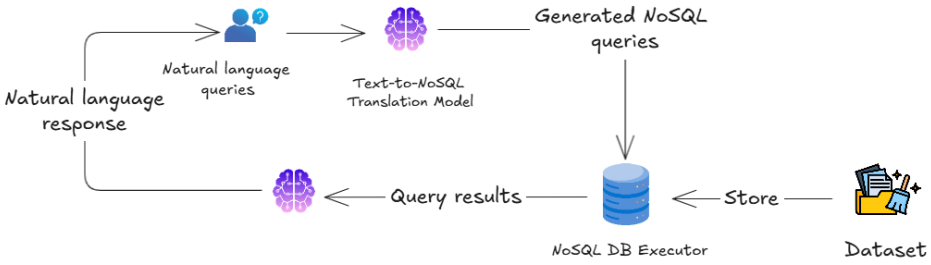

Figure 3.3: Schématisation de la solution "recherche sémentique simple".

Les étapes suivies pour cette approche sont les suivantes :

1. De RateMyProfessor, nous avons construit un corpus externe à partir de données textuelles préalablement nettoyées et structurées.
2. Une invite efficace (y compris le schéma de collection et des exemples de requêtes NoSQL) est construite.
3. Générer de requêtes NoSQL.

4. Integration de la requête pour générer la réponse finale.

5. Reformulation de la réponse générée en langage naturel.

Pour évaluer ce premier modèle développé, nous avons adopté une approche simple consistant à tester le modèle sur une seule question clé. Cette démarche nous a permis dobserver le comportement naturel du modèle sans aucune intervention externe ni instruction structurée, et dévaluer la pertinence de ses réponses dans un cas concret et ciblé :

Table 3.1: Résultats des métriques d'évaluation pour une question seule

Question	**Context Precision**	**Context Recall**	**Faithfulness**	**Answer Relevance**
Notes moyennes de difficulté et clarité par professeur ?	33.3%	33.3%	9.1%	100.0%

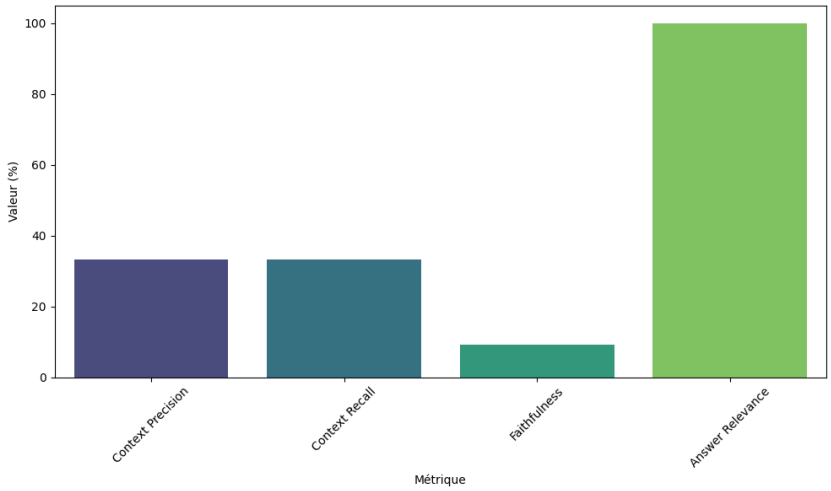

Figure 3.4: Performances des Métriques pour la Question Spécifiée.

Les résultats obtenus, de l'approche de la Recherche sémantique simple, traduisent que :

- Les scores de Context Precision et Context Recall sont relativement faibles, avec des valeurs proches de 30%, indiquant des difficultés à extraire et retrouver correctement les réponses pertinentes.
- La métrique Faithfulness affiche un score très bas (environ 10%), ce qui suggère que les réponses générées contiennent probablement des hallucinations ou des réponse non fidèles aux données sources.

- En revanche, la métrique Answer Relevance atteint un score parfait de 100%, montrant que les réponses générées sont bien alignées avec la question posée, au moins syntaxiquement.

3.6.2 Génération guidée/augmentée contextuellement

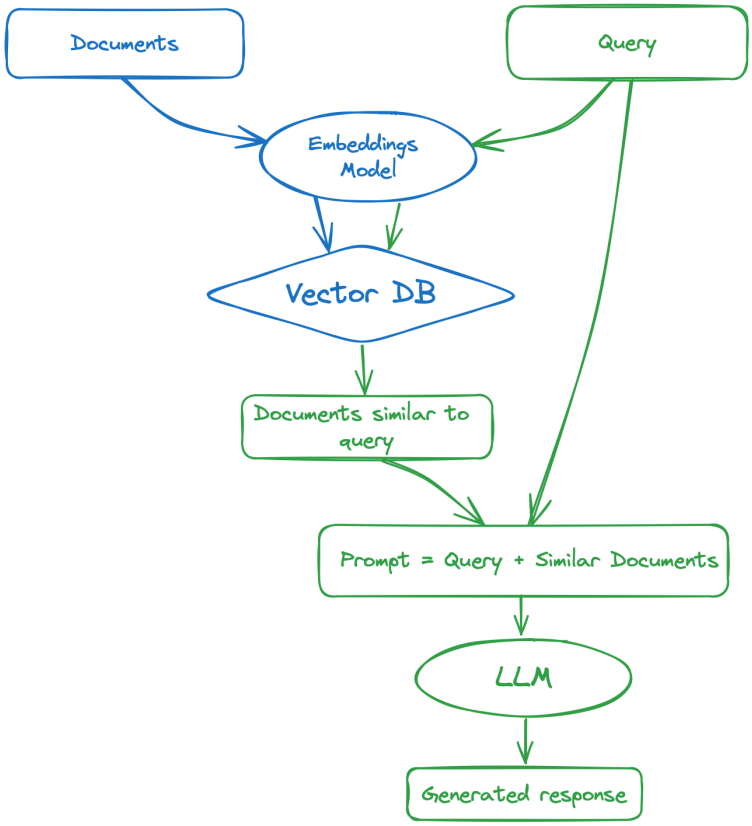

Figure 3.5: Schématisation de la solution "génération augmentée".

Le processus de la génération de réponses, dans un contexte éducatif, se résument en une séquence d'étapes à suivre:

1. La base de connaissances externe est construite à partir des informations de l'espace RateMyProfessor. Le prétraitement est effectué en suivant les bonnes pratiques du NLP.

2. A l'aide de la BD opérationnelle, les données tabulaires, y compris les intégrations migrent, de la machine locale vers lespace BD. Une fois les données migrées, une connectivité à la BD est établie et laccès sera assuré par les différents programmes.

3. Les embeddings générés seront utilisées pour intégrer à la fois les documents et les requêtes. Les embeddings sont des représentations vectorielles de données textuelles, facilitant la compréhension sémantique et les comparaisons de similarité. Elles améliorent lefficacité et la précision des systèmes de recherche dinformations en codant les relations sémantiques entre les documents et les requêtes.

   Dans cette étape, le modèle d'embedding de Sentence-Transformer est utilisé pour encoder les documents et les requêtes en vecteurs sémantiques. Les vecteurs générés sont ensuite stockés dans une base de données vectorielle.

4. A cette étape, les documents les plus similaires à la requête sont récupérés à partir de la base de données vectorielle.

5. Cette étape, une réponse à la requête en utilisant un LM est générée, après avoir récupéré les documents pertinents depuis la base de données vectorielle. Cette phase assure la transformation des informations récupérées en réponses cohérentes et adaptées au contexte.

   Pour guider le LM dans cette tâche, un prompt structuré est construit et il inclut :

   - une définition du rôle de lassistant (aide pour les données universitaires, professeurs, avis).
   - une instruction claire : modèle structuré d'exemples accompagnés de réponses, à partir du contexte fourni.

   Cette instruction permet dassurer une génération fluide et contextuelle de la réponse finale.

Pour évaluer le modèle développé, nous avons suivi deux stratégies de traitements : avec exemples, destinée à tester la capacité du modèle à exploiter le contexte fourni, et sans exemples, pour observer son comportement sans instructions explicites. Elles ont été testées dans deux configurations : Zero-shot et Few-shot, afin de mesurer limpact du prompt sur la qualité des réponses générées :

Table 3.2: Performances des modèles LLaMA sous différents types d'entraînement

Model	Training Type	Context Precision	Context Recall	Faithfulness	Answer Relevance
llama3.1:8b	Zero-shot	0.258	0.395	0.260	0.064
llama3.2:latest	Zero-shot	0.263	0.368	0.209	0.054
llama3.1:8b	Few-shot	0.721	1.000	0.187	0.701
llama3.2:latest	Few-shot	0.721	1.000	0.200	0.838

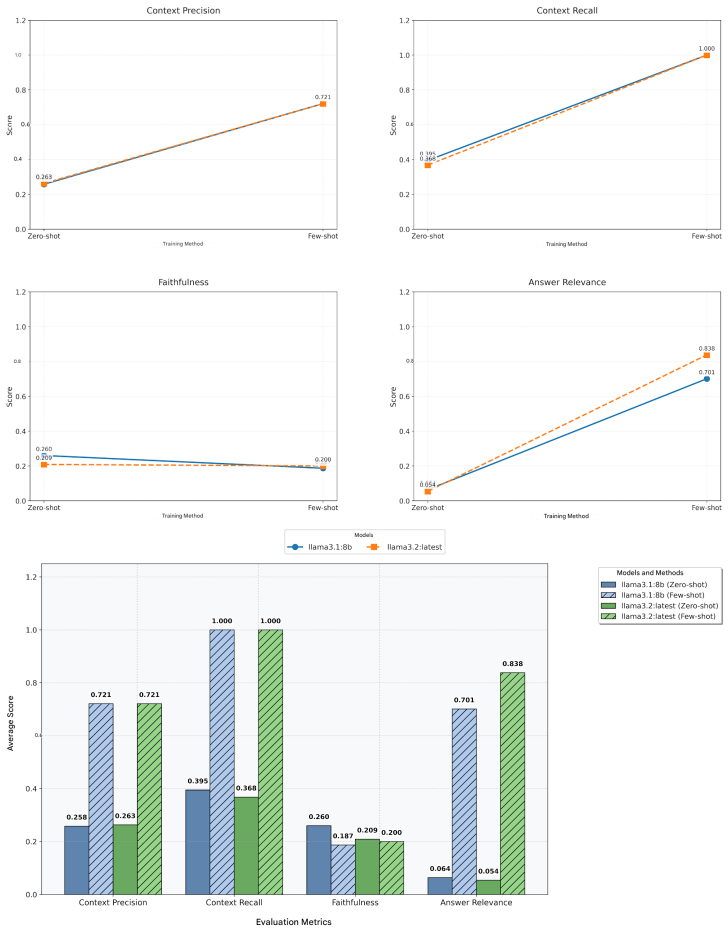

Figure 3.6: Performance comparée des modèles Llama par type d'entraînement.

Les résultats obtenus de cette approche montrent que :

- Le modèle llama 3.2 surpasse généralement le modèle llama 3.1 dans toutes les métriques évaluées.
- Les écarts sont particulièrement marqués dans les métriques Context Recall et Answer Relevance , où llama 3.2 atteint des scores plus élevés.
- L'utilisation de la méthode Few-shot améliore significativement les performances par rapport à Zero-shot pour tous les modèles et métriques.

3.6.3 Finetuning

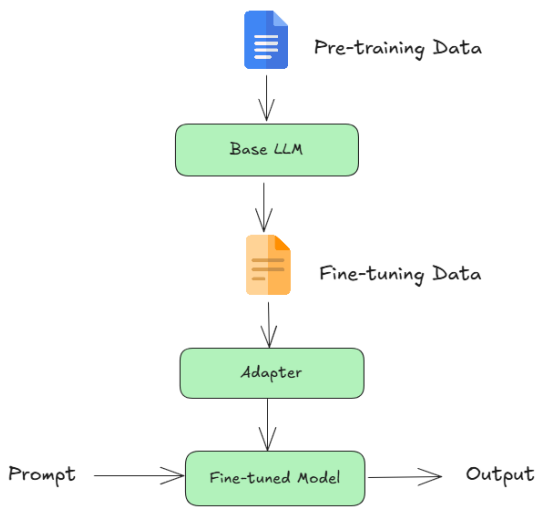

Figure 3.7: Schématisation de la solution "Finetuning"

Pour implémenter lapproche de Finetuning nous avons suivi les étapes suivantes:

1. choix du modèle : adapté à la tâche s'appuie sur deux critères principaux :

   - Coût, qualité et vitesse d'inférence : En général, plus un modèle est grand et précis, plus il est lent. Cest pourquoi, pour commencer, il est préférable dutiliser des modèles de taille moyenne comme 3B ou 8B.
   - Proximité du modèle avec le type de données : Le modèle choisi doit être adapté à la nature des données utilisées.

   Le modèle Llama convient bien à nos données textuelles et fait partie des modèles de langage les plus légers et efficaces.

2. Chaque modèle a une manière spécifique d'interpréter les données afin de les comprendre. Le modèle Llama s'attend à ce que les données soient structurées sous un format particulier pour être entraîné correctement. Ce format est le suivant :

   ```
   "message"
   {"role": "system", "content"} : explique son rôle en tant que système
   ```

`{"role": "user", "content"} : on lui pose une question`

`{"role": "assistant", "content"} : format de la réponse attendue`

Ce format est utilisé pour les exemples d'apprentissage (données d'entraînement).

3. Le modèle a été fine-tuné une technique qui consiste à modifier uniquement une petite partie des paramètres du modèle, réduisant ainsi la consommation de mémoire et les temps de calcul.

4. Une fois le modèle entrainé, il est testé et par la suite évaluer, en fonction de données de test nouvelles.

5. Enfin, le modèle entraîné est sauvegardé localement pour une utilisation future.

Ces étapes détaillent le processus méthodologique suivi pour le Finetuning.

Le tableau 3.3 résume les valeurs de perte dentraînement et de validation mesurées après chaque époque, reflétant ainsi lapprentissage et la généralisation du modèle :

Table 3.3: Résultats obtenus après chaque époque d'entraînement

Epoch	Training Loss	Validation Loss
0	1.006100	0.520065
1	0.955795	0.504463
2	0.908005	0.489329
3	0.862605	0.474649
4	0.819475	0.460410

La Figure 3.8 montre un graphique représentant la variation du loss dentraînement et du loss de validation.

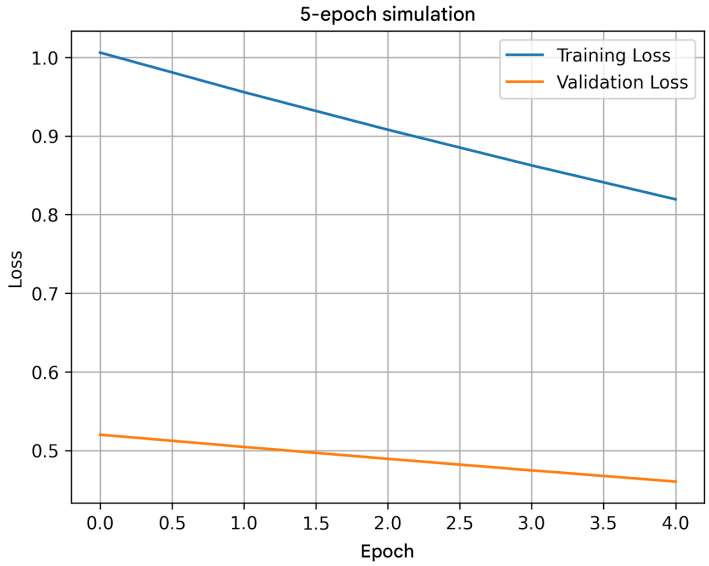

Figure 3.8: Graphique montrant la perte d'entraînement et de validation par époque.

Les résultats récoltés traduisent que:

1. La perte dentraînement diminue régulièrement au fil des époques, indiquant une stabilisation progressive du modèle sur les données dapprentissage.

2. La perte de validation suit une tendance similaire, mais avec une baisse plus lente, montrant une amélioration progressive de la généralisation. Lécart stable entre les deux courbes suggère une absence de surapprentissage.

Pour mesurer lefficacité du modèle après entraînement, nous avons conçu un protocole dévaluation combinant des questions tirées de notre propre base de données (Question 1 et 2), ainsi que des questions externes non vues par le modèle (Question 3 et 4). Cette approche permet de juger à la fois sa fidélité aux données dapprentissage et sa capacité à généraliser à de nouveaux contextes: (3.4, 3.9)

Table 3.4: Résultats des métriques par question

Question	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**Similarity**	**Hallucination**
Question 1	0.1507	0.09722	0.1233	0.5385	0.9231
Question 2	0.1342	0.0272	0.0805	0.5000	0.9406
Question 3	0.0725	0.0105	0.0518	0.5455	0.9589
Question 4	0.1390	0.0108	0.1070	0.6923	0.9400

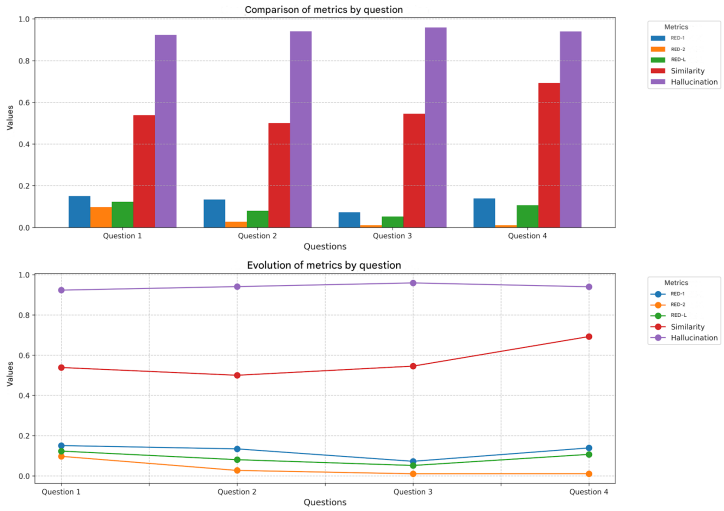

Figure 3.9: Graphique montrant la perte d'entraînement et de validation par époque.

Les résultats générés par finetuning sont traduisent que:

- Les performances du modèle varient selon les questions. Contrairement à une tendance solide, les scores ROUGE-1 restent relativement faibles, allant de 0.07 à 0.15 , ce qui indique que les réponses générées sécartent significativement des références attendues. Les scores ROUGE-2 sont encore plus bas (de 0.01 à 0.10), confirmant une faible correspondance au niveau des séquences de mots. Le score ROUGE-L , bien quun peu plus élevé (0.05 à 0.12), reste néanmoins insuffisant pour refléter une bonne similarité globale.

- La Question 1 obtient les meilleurs scores ROUGE, montrant un léger avantage par rapport aux autres, mais cela reste limité en termes de qualité de réponse.

- La similarité sémantique est modérément bonne, variant entre 0.50 et 0.69 , avec un pic sur la Question 4 . Cela montre que certaines réponses partagent des concepts proches avec les données sources, sans toutefois atteindre une adéquation optimale.

- Les scores dhallucination (0.920.96) sont remarquablement élevés, soulignant la fiabilité du modèle à produire des réponses ancrées dans des données vérifiables, avec une tendance minimale à générer des informations erronées.

Le modèle présente des limites importantes dans la génération de réponses précises et fidèles aux données, comme en témoignent les faibles scores ROUGE. Bien quil réussisse à

éviter certains biais avec des scores dhallucination élevés , cela ne compense pas la faible pertinence des réponses. Une amélioration notable est nécessaire, notamment au niveau de la compréhension contextuelle et de la fidélité à la source dinformation, afin dassurer une utilisation fiable du système dans un cadre pratique.

3.6.4 Comparaison des méthodes utilisées

Une comparaison a été réalisée entre les différentes approches utilisées dans la solution afin dévaluer leurs performances respectives et didentifier celle qui offre les meilleurs résultats. Les résultats sont résumés dans le tableau 3.5 et la figure 3.10, qui compare les performances des systèmes dévaluation sur plusieurs métriques clés telles que la précision contextuelle, le rappel contextuel, la fidélité et la pertinence des réponses.

Table 3.5: Comparaison des performances des approches utilisées

Approch	**Context Precision**	**Context Recall**	**Faithfulness**	**Answer Relevance**
Fine-tuning	N/A	N/A	Hallucination ≈ 0.08	ROUGE-L ≈ 0.123
Search	0.333	0.333	0.09	1.00
RAG	0.721	1.0	0.200	0.838

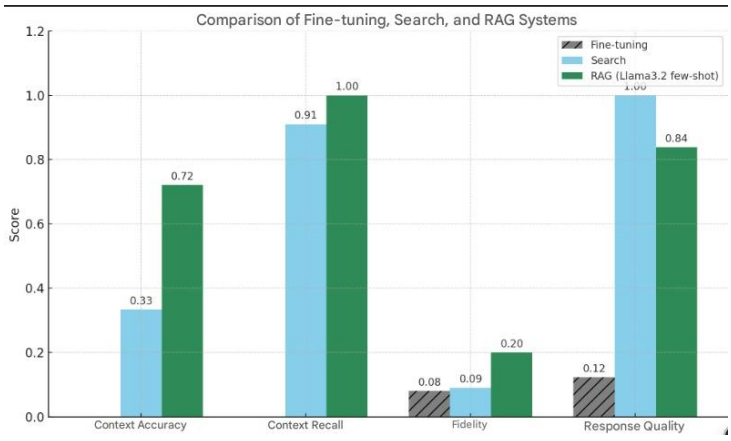

Figure 3.10: Graphique montrant la comparaison des performances des approches utilisées.

- Le *fine-tuning* se distingue par une bonne fidélité, avec peu dhallucinations. Cependant, il ne récupère pas de contexte de manière efficace, ce qui rend les métriques de précision et de rappel indisponibles. Les réponses fournies sont souvent génériques ou incomplètes, ce qui limite sa performance globale.

- La méthode *Search* excelle en termes de rappel. Elle retrouve la plupart des documents pertinents. Toutefois, sa précision est faible car elle récupère également des informations non pertinentes. Malgré cela, la qualité des réponses est excellente, probablement grâce à un filtrage effectué par lutilisateur final.

- *L*a génération augmentée textuellement combine les forces du bon rappel et dune précision correcte, ce qui en fait une approche bien équilibrée, supérieure à celle du fine-tuning. Les réponses générées sont de très bonne qualité, légèrement inférieure au search mais meilleure quavec le fine-tuning.

En résumé, le *fine-tuning* convient si lon cherche un modèle léger et stable, bien qu'il reste à le prouver en précision contextuelle pour un big data. Le *search* est idéale pour des tâches simples de question-réponse, offrant une grande fiabilité. Enfin, la méthode de *la génération augmentée* ou guidée contextuellement constitue la meilleure synergie entre accès au contexte et qualité des réponses, mais elle nécessite une vigilance accrue face aux risques d'hallucinations.

3.6.5 Présentation de linterface utilisateur

Dans cette partie, nous présentons les résultats de notre travail sous forme dexemples illustratifs, où des questions ont été posées à chacun des modèles, et les réponses générées ont été analysées afin de montrer lefficacité de chaque approche adoptée.

- Résultats obtenus avec le modèle de la simple recherche sémantique: la figure **??** illustre un exemple dinteraction avec le modèle utilisant la technique de recherche sémantique simple. On y retrouve une requête utilisateur, suivie de la réponse générée par le modèle après récupération dinformations externes pertinentes :

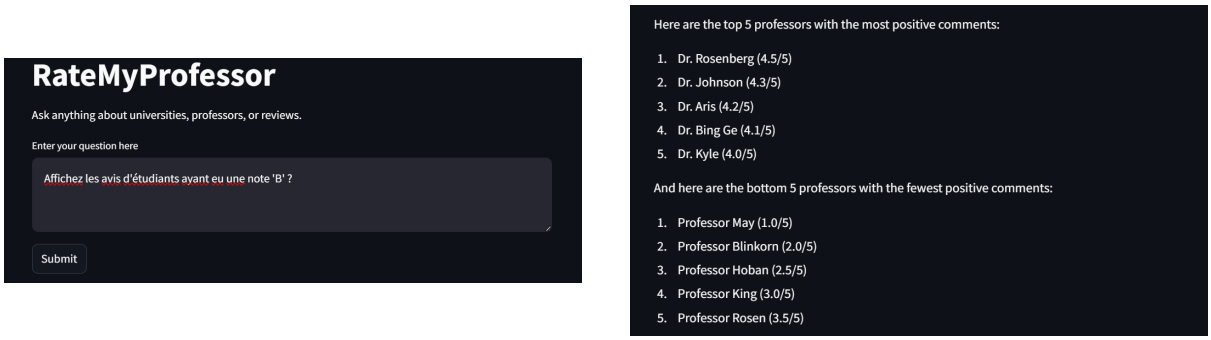

Figure 3.11: Exemple de réponses générées par le modèle 1

- Résultats obtenus avec le modèle la génération guidée/augmenter: la figure 3.12 présente un exemple dinteraction avec le modèle utilisant la technique de génération augmentée. On y retrouve une requête utilisateur, suivie de la réponse générée par le modèle après récupération dinformations externes pertinentes :

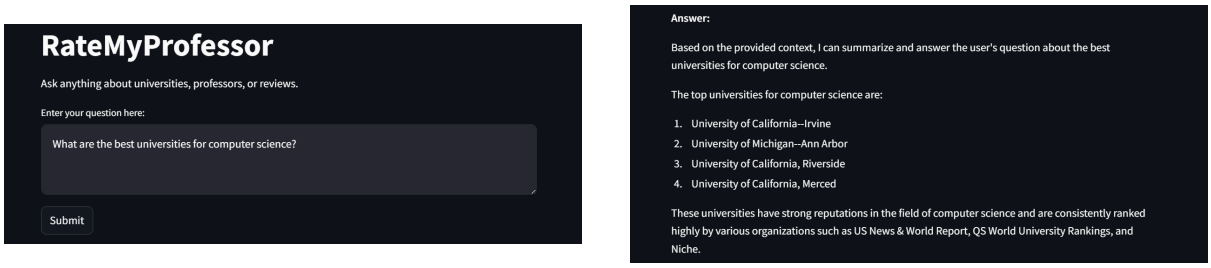

Figure 3.12: Exemple de réponses générées par le modèle 2

- Résultats obtenus avec le modèle Finetuning: la figure 3.13 illustre un exemple de sortie générée par le modèle après avoir appliqué la méthode Finetuning . Ici, le modèle a été adapté à laide de données spécialisées, ce qui permet dobtenir des réponses plus précises dans le domaine ciblé.

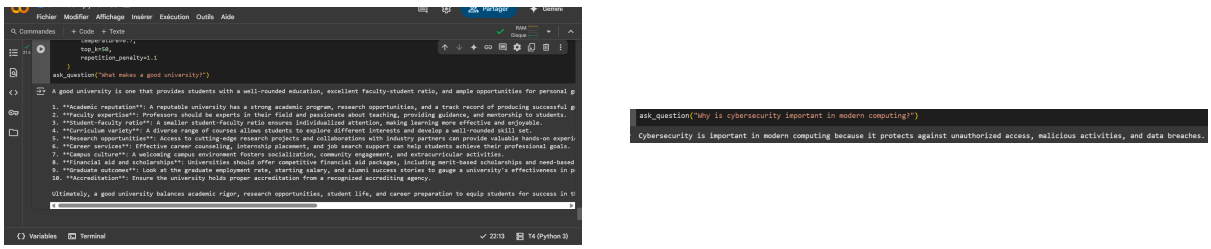

Figure 3.13: Exemple de réponses générées par le modèle 3

3.7 Conclusion

Les expérimentations menées dans ce chapitre confirment l'efficacité des approches pour générer des réponses précises et contextualisées dans le domaine éducatif. Les résultats démontrent une nette amélioration par rapport à un LLM de base, avec une réduction significative des hallucinations et une meilleure adéquation aux requêtes spécifiques. Cependant, certaines limites persistent, notamment la dépendance à la qualité des données externes et la complexité du déploiement. Ces éléments de compréhension clés ouvrent des perspectives prometteuses, comme l'optimisation des prompts, l'intégration de mécanismes de vérification en temps réel ou encore le couplage avec des systèmes multi-agents.

BIBLIOGRAPHY

[1] Nilsson, N. J. (2010). *The Quest for Artificial Intelligence.* Cambridge University Press. Available at: https://www-cs-faculty.stanford.edu/~nilsson/QAI/qai.pdf

[2] Robert, Jérémy. *Study On Machine Learning Algorithms. ResearchGate, 2021*. Available at: https://www.researchgate.net/publication/354306269_Study_On_Machine_Learning_Algorithms

[3] Farhad Mortezapour Shiri, , Thinagaran Perumal1, Norwati Mustapha1, and Raihani Mohamed. *A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU* .arXiv, 2023. Available at: https://arxiv.org/pdf/2305.17473

[4] Haton, Jean-Paul.*L'intelligence artificielle générative. Techniques de l'Ingénieur, 2023*. Available at: https://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/intelligence-artificielle-42679210/l-intelligence-artificielle-generative-h3760/

[5] Diksha Khurana, Aditya Koli, Kiran Khatter, Sukhdev Singh. *Natural Language Processing: State of The Art, Current Trends and Challenges*. arXiv, 2017. Available at: https://arxiv.org/abs/1708.05148

[6] Ezin, Eugène. *Intelligence Artificielle et ses domaines d'applications. ResearchGate, 2023*. Available at: https://www.researchgate.net/profile/Eugene-Ezin/publication/367525222_INTELLIGENCE_ARTIFICIELLE_ET_SES_DOMAINES_D'APPLICATIONS/links/63d6a66d64fc860638f8a1e2/INTELLIGENCE-ARTIFICIELLE-ET-

[7] Jurafsky, Daniel, et James H. Martin. *Speech and Language Processing, chapitre 3 :*

"N-Gram Language Models". Available at: https://web.stanford.edu/~jurafsky/slp3/3.pdf

[8] Katy Fokou. *PNL et modèles linguistiques*. mai 2018. Available at: https://www.smalsresearch.be/nlp-modeles-de-langue/#:~:text=Quest%2Dce%20qu,%C3%A0%20une%20s%C3%A9quence%20de%20mots.

[9] Vaswani, Ashish et al. *Attention Is All You Need*. arXiv, 2017. Available at: https://arxiv.org/pdf/1706.03762

[10] Humza Naveeda , Asad Ullah Khanb, , Shi Qiuc, , Muhammad Saqibd,e, , Saeed Anwarf,g, Muhammad Usmanf,g, Naveed Akhtarh,j , Nick Barnesi , Ajmal Mianj. *A Comprehensive Overview of Large Language Models*. 17 Oct 2024. Available at: https://arxiv.org/pdf/2307.06435

[11] Taojun Hu, Xiao-Hua Zhou. *Unveiling LLM Evaluation Focused on Metrics: Challenges and Solutions*. April 16, 2024. Available at: https://arxiv.org/pdf/2404.09135

[12] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu et Chenguang Zhu. *G-EVAL: NLG Evaluation using GPT-4 with Better Human Alignment*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. Available at: https://aclanthology.org/2023.emnlp-main.153.pdf

[13] Arvind Padmanabhan . *NLP is a subset of AI and uses ML/DL techniques. Source: Sathiyakugan 2018*. Devopedia Images, 2019. Available at: https://devopedia.org/natural-language-processing

[14] Kashif Sultan ,Kashif Sultan et Kashif Sultan . "Big Data Perspective and Challenges in Next Generation Networks". *Future Internet*, vol. 10, no. 7, p. 56, 2023. Available at: https://www.mdpi.com/1999-5903/10/7/56

[15] Manika Nagpal. *8 Deep Learning Architectures Data Scientists Must Master*. 28 Oct 2024. Available at: https://www.projectpro.io/article/deep-learning-architectures/996.

[16] Minority Programmers. *Exploiter la puissance des LLM : explorer les frontières de la recherche en IA avec de grands modèles linguistiques*. 18 mai 2023. Available at: https://medium.com/minority-programmers/harnessing-the-power-of-llms-exploring-ai-research-frontiers-with-large-language-models-eef2bcd31dd8

[17] E-learning company "Data Science Dojo". *What Is N-Gram and How Does It Work?*. Available at: https://botpenguin.com/glossary/n-gram

[18] Amr Mousa, Hong-Kwang Kuo, Lidia Mangu et Hagen Soltau . *Morpheme-based Feature-Rich Language Models Using Deep Neural Networks for LVCSR of Egyptian Arabic*. ResearchGate, 2014. Available at: https://www.researchgate.net/publication/261334587_Morpheme-based_feature-rich_language_models_using_Deep_Neural_Networks_for_LVCSR_of_Egyptian_Arabic/figures?lo=1

[19] Pranab Sahoo , Ayush Kumar Singh , Sriparna Saha , Vinija Jain2, , Samrat Mondal1 and Aman Chadha. *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications*. arXiv, 2024. Available at: https://arxiv.org/pdf/2402.07927

[20] Cheonsu Jeong. *Fine-tuning and Utilization Methods of Domain-specific LLMs.*. arXiv, 2024. Available at: https://arxiv.org/pdf/2401.02981

[21] Natassha Selvaraj. *What Is Retrieval-Augmented Generation (RAG)?*. Mar 14, 2025. Available at: https://www.datacamp.com/blog/what-is-retrieval-augmented-generation-rag

[22] Zhao, Liang et al. *When to Do What: A Taxonomy of LLM Prompting Methods*. arXiv preprint arXiv:2410.12837, 2024. Available at: https://arxiv.org/pdf/2410.12837

[23] Luo, Hongyin et al. *MathCoder: Seamless Code Integration in LLMs for Enhanced Mathematical Reasoning*. arXiv preprint arXiv:2312.10997, 2023. Available at: https://arxiv.org/pdf/2312.10997

[24] Hao Yu , Aoran Gan , Kai Zhang , Shiwei Tong , Qi Liu , et Zhaofeng Liu. *Evaluation of Retrieval-Augmented Generation: A Survey* . arXiv preprint arXiv:2405.07437, 2024. Available at: https://arxiv.org/pdf/2405.07437

[25] Cobus Greyling. *RAG Evaluation*. Sep 1, 2023. Available at: https://cobusgreyling.medium.com/rag-evaluation-9813a931b3d4

[26] VATSAL, Shubham ; DUBEY, Harsh. *A Survey of Prompt Engineering Methods in Large Language Models for Different NLP Tasks*. arXiv:CS.CL, 2024. Available at: https://arxiv.org/abs/2402.17152

[27] XIONG, Guangzhi et al. *Improving Retrieval-Augmented Generation in Medicine with Iterative Follow-up Questions*. arXiv:CS.CL, 2024. Available at: https://arxiv.org/abs/2405.15845

[28] SHI, Liang et al. *A Survey on Employing Large Language Models for Text-to-SQL Tasks*. arXiv:CS.CL, 2024. Available at: https://arxiv.org/abs/2405.05082

[29] SEABRA, Antony et al. *Contrato360 2.0: A Document and Database-Driven Question-Answer System Using Large Language Models and Agents*. arXiv:CS.AI, 2024. Available at: https://arxiv.org/abs/2405.15681

[30] FEBRIAN, Gilang Fajar et al. *KemenkeuGPT: Leveraging A Large Language Model on Indonesia's Government Financial Data and Regulations to Enhance Decision Making*. arXiv:CS.AI, 2024. Available at: https://arxiv.org/abs/2405.17427

[31] KERMANI, Arshia et al. *A Systematic Evaluation of LLM Strategies for Mental Health Text Analysis: Fine-tuning Vs. Prompt Engineering Vs. RAG*. arXiv:CS.CL, 2025. Available at: https://arxiv.org/abs/2501.12345

[32] PING, Heng et al. *HDLCoRe: A Training-Free Framework for Mitigating Hallucinations in LLM-Generated HDL*. arXiv:CS.CL, 2025. Available at: https://arxiv.org/abs/2502.06789

[33] IBN AHAD, Jawad et al. *Empowering Meta-Analysis: Leveraging Large Language Models for Scientific Synthesis*. arXiv:CS.CL, 2024. Available at: https://arxiv.org/abs/2406.05802

[34] MAILACH, Alina et al. *Themes of Building LLM-based Applications for Production: A Practitioner's View*. arXiv:CS.SE, 2024. Available at: https://arxiv.org/abs/2406.12345

[35] Cheng, Anzhe; Zhang, Peiyu; Duan, Shukai; Kanakaris, Nikos; et al. *A Comprehensive Survey of Retrieval-Augmented Generation*. arXiv preprint arXiv:2306.03901, 2023. Available at: https://arxiv.org/pdf/2306.03901