République Algérienne Démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

École Nationale Polytechnique
Département Génie Industriel

Mémoire de Projet de Fin d'Études
Pour l'obtention du diplôme d'Ingénieur d'État en Génie Industriel
Option : Data Science & Intelligence Artificielle

Développement d'un assistant intelligent basé sur l'IA Générative

Application au sein de KPMG Deal Advisory

Auteurs :

KARTOBI Sofiane
OULD ABDALLAH Mohamed Riad

Présenté et soutenu publiquement le (08/07/2024)

Composition du Jury :

Présidente	Mme. BOUCHAFAA Bahia	MCA	ENP
Promotrice	Mme. AIT BOUAZZA Sofia	MAA	ENP
Examinateur	M. ARKI Oussama	MCA	ENP
Encadrante	Mme. BENDEMIRAD Lydia	Data Analyst	KPMG
Encadrante	Mme. SEMOUD Camelia	Consultante R&S	KPMG

ENP 2024

République Algérienne Démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

École Nationale Polytechnique
Département Génie Industriel

Mémoire de Projet de Fin d'Études
Pour l'obtention du diplôme d'Ingénieur d'État en Génie Industriel
Option : Data Science & Intelligence Artificielle

Développement d'un assistant intelligent basé sur l'IA Générative

Application au sein de KPMG Deal Advisory

Auteurs :

KARTOBI Sofiane
OULD ABDALLAH Mohamed Riad

Présenté et soutenu publiquement le (08/07/2024)

Composition du Jury :

Présidente	Mme. BOUCHAFAA Bahia	MCA	ENP
Promotrice	Mme. AIT BOUAZZA Sofia	MAA	ENP
Examinateur	M. ARKI Oussama	MCA	ENP
Encadrante	Mme. BENDEMIRAD Lydia	Data Analyst	KPMG
Encadrante	Mme. SEMOUD Camelia	Consultante R&S	KPMG

ENP 2024

Dédicaces

À mon père, ta guidance et ton soutien indéfectible ont forgé l'homme que je suis aujourd'hui. Tes principes et tes valeurs ont été pour moi une source constante d'inspiration, ta sagesse et ta force sont des phares qui éclairent mon chemin, et mon respect et mon admiration pour toi grandissent avec le temps.

À ma mère, partie trop tôt pour que je puisse me souvenir de toi, ton absence a laissé un vide que j'essaie de combler en vivant une vie qui te rendrait fière. Bien que je ne puisse me rappeler de ton visage ou de ta voix, ton amour continue de m'entourer et de m'inspirer et ta mémoire est une source de force et de motivation pour moi.

À mes sœurs et à mon frère, votre soutien inconditionnel a été une source de réconfort et de motivation tout au long de ce parcours. Vos encouragements et vos conseils ont été précieux, et je suis fier de vous avoir comme famille.

À mes tantes, véritables piliers de soutien qui ont comblé le vide laissé par ma mère avec leur amour inconditionnel, leur présence réconfortante, et leur éducation précieuse qui a façonné l'homme que je suis aujourd'hui.

À Moumouh, pour toujours être là, partager tous ces moments, et me soutenir à fond, un vrai G !

À mon cousin Hichem pour tous les bons souvenirs, les aventures que nous avons vécus ensemble.

À Sofiane, mon binôme Madridistas, compagnon de galères et de succès, celui avec qui les lignes de code deviennent des défis surmontés, des rires spontanés et des moments inoubliables, "Siempre fieles al blanco y al Madrid."

À mes amis, ma Gang, Oussama, Yanis, Akram, Malek et le grand Cherif des vrais G.

Aux étudiants du département Génie Industriel et aux membres de IEC avec lesquels j'ai passé de trés beaux et inoubliables moments. Tahya IEC !

À mes merveilleuses rencontres, Youcef l'arabisant, Walid les pauses Thé, Tamy ZZPhone et Aziz le doué.

Et enfin, À toutes les personnes qui ont marqué mon parcours et à ceux qui ont discerné en moi un potentiel prometteur, qui ont cru en mes capacités et en ma réussite

- Riyad

Dédicaces

À mes parents, piliers inébranlables de ma vie et de mes ambitions. Votre amour, votre patience et vos sacrifices m'ont porté jusqu'ici. Papa, ta rigueur et ton éthique de travail m'ont montré la voie. Maman, ta tendresse et ton soutien constant ont été mon refuge dans les moments difficiles. Ce mémoire est le fruit de votre dévouement et de votre foi en moi.

À mes sœurs, complices de toujours et sources inépuisables d'inspiration. Vos encouragements, votre humour et votre affection ont été des rayons de soleil dans ce parcours exigeant. Merci d'avoir été à la fois mes plus ferventes supportrices et mes critiques les plus honnêtes.

À mes grands-parents, gardiens de notre histoire familiale, de valeurs précieuses et surtout du patrimoine algérois de pure souche. Votre sagesse, vos récits et votre amour inconditionnel ont enrichi ma vie et nourri ma détermination.

À la mémoire de mon grand-père Sahnoune lah yrahmo, étoile bienveillante qui continue de briller sur mon chemin. Ton courage, ta curiosité intellectuelle et ta joie de vivre m'accompagnent à chaque instant. Ce mémoire est aussi un hommage à ton héritage. Tu aurais été fier, j'en suis sûr

À mes professeurs et mentors, qui m'ont guidé tout au long de ce parcours académique. Vos enseignements et votre sagesse resteront gravés en moi.

À mes frères et soeurs du comité 2023 IEC, avec qui j'ai partagé tant de moments de joie, de stress et de réussite : Aymen, Raihane, Mehdi, Sofiane, Samah, Sihem et Amira

À toute la famille Indus : Alice, Hadil, Serine, Manel, Inès, Younes, Amine AGG et tout le reste.

Au club IEC, Merci d'avoir été bien plus qu'une simple activité extrascolaire.

Vous avez été un lieu d'épanouissement, de découvertes et d'amitiés durables, une deuxième famille. Les compétences acquises et les expériences vécues au sein du club ont enrichi mon parcours académique et personnel. Une entité, je dirais même un feeling qui a rendu l'expérience Génie indus une aventure unique et inoubliable, transcendant largement le cadre académique traditionnel.Ce mémoire porte aussi l'empreinte des valeurs et de l'esprit d'équipe cultivés ensemble. Tahya IEC !

À mes amis et camarades de classe, sans qui ces années de spécialité n'auraient pas été les même : Hamza, Bilel, Riad (mon binome j'arrive)

A nos chers alumni qui nous ont tant marqués, tant appris et partager leur connaissances et savoirs faire

Mention spéciale pour l'équipe de stagiaire ESAssienne : Youcef 4 Bassins et Walid Ainaadja, Anaïs, Tamy Preure Le Tilleul et Yacine amateur des interviews Belaïli

Et biensur ! À mon binome Riad aka Ryoo, bien plus qu'un simple partenaire de travail. Toi et moi, c'était Forza en mode réalité. Drifts sur les virages serrés des deadlines, nitro dans les lignes droites des nuits blanches. "What color is your diploma ?" On l'a peint ensemble, mon pote. Andrew Tate dirait qu'on a brisé la matrice. Moi je dis qu'on l'a réécrite, ligne de code après ligne de code.

"Hasta el final, vamos Real !" Cette devise du Real Madrid a été le mantra qui m'a poussé à donner le meilleur de moi-même tout au long de ce parcours académique. À l'image des remontadas légendaires du club, j'ai appris à y croire jusqu'à la dernière minute, à ne jamais baisser les bras face aux défis. Cette mentalité de gagneur, cette foi inébranlable dans la victoire même quand tout semble perdu, a été ma source d'inspiration pour surmonter les obstacles et mener à bien ce mémoire.

Et pour conclure, comme l'a si bien dit le GOAT incontestable Cristiano Ronaldo dos Santos Aveiro :

"Muchas gracias afición, esto es para vosotros ! SIUUU !"

- Sofiane

Remerciement

Louange à Dieu seul, clément et miséricordieux

Tout d'abord, nous adressons nos sincères remerciements à Madame ***Sofia AIT BOUAZZA****, notre promotrice. Votre encadrement, vos conseils judicieux et votre disponibilité ont été déterminants dans l'aboutissement de ce travail.*

Nous remercions chaleureusement Monsieur ***Oussama ARKI*** *et Madame* ***Bahia BOUCHAFAA****, d'avoir accepté d'évaluer notre travail. Votre expertise et votre temps sont précieux, et nous sommes honorés de pouvoir vous présenter le fruit de nos recherches.*

Nous sommes également reconnaissants envers Mesdames ***Lydia BENDE-MIRAD****,* ***Camélia SEMMOUD*** *et* ***Nihad KERROUM****, pour leur soutien indéfectible, leurs précieux conseils et le partage généreux de leurs connaissances.*

Notre gratitude s'étend à l'ensemble de l'équipe pédagogique du département ***Génie Industriel****. Nous remercions particulièrement Messieurs Iskander ZOUAGHI, Ali BOUKABOUS, Oussama ARKI, Ahmed Farouk YEDDOU, Yassine AIT ALI YAHIA et Madame Bahia BOUCHAFAA, sans qui nous n'aurions pas eu les connaissances nécessaires pour réaliser ce mémoire.*

Nous exprimons notre reconnaissance à l'équipe ***KPMG****, en particulier à Mehdi BETTAHAR, Directeur du département Deal Advisory, pour la confiance qu'il nous a accordée et l'opportunité d'intégrer son équipe.*

Nous remercions chaleureusement Messieurs Lotfi ABDI, Yasser CHELGHAM, Abdelkrim BAHMED et Mesdames Nouzha MORSLI et Aicha BELOUZDAD d'avoir constamment été à l'écoute de nos questions et de nous proposer généreusement leur aide.

- Sofiane & Riyad

ملخص

الهدف من هذا العمل هو تطوير مساعد ذكي يستخدم الذكاء الاصطناعي التوليدي، بما في ذلك LLMs و RAG لتحسين أداء المستشارين في قسم الاستشارات التجارية لدى KPMG الجزائر، وهو لاعب رئيسي في مجال الاستشارات والتدقيق المالي. بصفتها شبكة دولية لشركات الخدمات المهنية، تعمل على دمج مثل هذا المساعد رافعة استراتيجية لتعزيز الميزة التنافسية لـ KPMG وتحسين كفاءة خدماتها في عمليات التكامل والاستحواذ والعناية الواجبة للشركات الكبرى.

الكلمات المفتاحية : عمليات الدمج والاستحواذ ، استشارات الصفقات ، الذكاء الاصطناعي التوليدي، LLM، RAG ، البحث عن المعلومات، مساعد، KPMG

Abstract

The objective of this work is to develop an intelligent assistant leveraging generative AI, including LLMs and RAG, to optimize the performance of consultants within the Deal Advisory department of KPMG Algeria, a major player in financial consulting and auditing. As an international network of professional services firms, the integration of such an assistant represents a strategic lever for reinforcing KPMG's competitive edge and enhancing the efficiency of its mergers and acquisitions and due diligence services for large corporations.

Keywords : Mergers and acquisitions, Deal advisory, Generative AI, RAG, LLM, Information retrieval, Assistant, KPMG

Résumé

L'objectif de ce travail est de développer un assistant intelligent exploitant l'intelligence artificielle générative, notamment les LLMs et le RAG, afin d'optimiser les performances des consultants au sein du département Deal Advisory de KPMG Algérie, acteur majeur du conseil et de l'audit financier. KPMG étant un réseau international de cabinets d'expertise, l'intégration d'un tel assistant constitue un levier stratégique pour renforcer son avantage concurrentiel et améliorer l'efficacité de ses services de fusions-acquisitions et de due diligence pour les grands groupes.

Mots-Clés : Fusions acquisitions, Deal advisory, Intelligence Artificielle Générative, RAG, LLM, Recherche d'information,Assistant, KPMG.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

- ANN : Approximate Nearest Neighbors
- API : Application Programming Interface
- BDD : Buyer Due Diligence
- BERT : Bidirectional Encoder Representations from Transformers
- BLEU : Bilingual Evaluation Understudy
- BM25 : Best Match 25
- BPMN : Business Process Model and Notation
- CAC : Commissaire Aux Comptes
- CoT : Chain-of-Thought
- CSV : Comma-Separated Values file
- D&A : Deal & Analytics
- EBITDA : Earnings Before Intrests, Taxes, Depreciation, and Amortization
- ERNIE : Enhanced Representation through Knowledge Integration
- EY : Ernst & Young
- F&A : Fusions & Acquisitions
- FAISS : Facebook AI Similarity Search
- FDD : Financial Due Diligence
- FEC : Fichier des Ecritures Comptables
- GAN : Generative Adversarial Network

- GenAI : Generative Artificial Intelligence
- GL : Grand Livre
- GPT : Generative Pre-trained Transformers
- GPU : Graphics Processing Unit
- GRU : Gated Recurrent Unit
- HMM : Hidden Markov Model
- HSNW : Hierarchical Navigable Small Worlds
- HTML : HyperText Markup Language
- IA : Intelligence Artificielle
- IAG : Intelligence Artificielle Générale
- IHM : Interface Homme-Machine
- KNN : K Nearest Neighbors
- KPMG : Klynveld Peat Marwick Goerdeler
- LCEL : LangChain Chain Expression Language
- LLM : Large Language Model
- LSTM : Long Short Term Memory
- M&A : Mergers & Acquisitions
- MoE : Mixture of Experts
- MTEB : Massive Text Embedding Benchmark
- NLG : Natural Language Generation
- NLP : Natural Language Processing
- NLU : Natural Language Understanding
- PDF : Portable Document Format
- PDG : Président Directeur Général.

- POC : Proof Of Concept
- PwC : PricewaterhouseCoopers
- Q&A : Question & Answer
- R&S : Research & Strategy
- RAFT : Retrieval Augmented Fine-Tuning
- RAG : Retrieval Augmented Generation
- RAGET : RAG Evaluation Toolkit
- RNN : Recurrent Neural Network
- RoBERTa : Robustly Optimized BERT Pre-training Approach
- ROUGE : Recall-Oriented Understudy for Gisting Evaluation
- RPA : Robotic Process Automation
- SEC : Securities and Exchange Commission
- Seq2Seq : Sequence-to-sequence
- SiS : Streamlit in Snowflake
- SMID : Small and Medium Industrial/Institutional Deals
- SPA : Société Par Action
- SWOT : Strengths, Weaknesses, Opportunities, and Threats
- TS : Transaction Services
- URL : Uniform Resource Locator
- VDD : Vendor Due Diligence

Introduction Générale

Au cours de la dernière décennie, l'intelligence artificielle (IA) est devenue un pilier majeur de l'innovation, stimulant une transformation significative des opérations et des modèles d'affaires. Cette dernière est en train de marquer un tournant important, notamment avec l'avenu des Transformers et LLMs (Large Language Models), en passant de l'IA traditionnelle, qui avait un champ d'action assez restreint principalement destinées aux tâches prédictives, vers une nouvelle forme prometteuse à savoir l'IA Générative, ouvrant ainsi de tout nouveaux horizons permettant aux entreprises d'améliorer leurs flux de travail internes et enrichir leurs produits et services.

Dans cette nouvelle ère numérique, la capacité d'accéder rapidement à des informations pertinentes et précises est essentielle pour améliorer la productivité et prendre des décisions éclairées. Avec un volume toujours croissant de données numériques, être capable de trouver la bonne information est devenu une tâche prioritaire.

Le secteur de l'audit et du conseil connaît une croissance continue. Il est largement dominé par quatre grands cabinets : KPMG, Deloitte, PricewaterhouseCoopers (PwC) et Ernst & Young (EY). Ce marché est extrêmement lucratif et reste stable, en grande partie grâce aux obligations légales imposées aux entreprises, qui doivent constamment justifier leur conformité par des documents officiels. Les cabinets de conseil et d'audit peuvent se spécialiser dans divers domaines, notamment la stratégie, la comptabilité et la finance, la logistique, le management, le marketing, ainsi que la data et l'intelligence artificielle (IA). Par exemple, les MBB (McKinsey & Company, Bain & Company, Boston Consulting Group) sont connus pour leur expertise en stratégie. Des organisations comme QuantumBlack, une filiale de McKinsey, se concentrent sur la data science et l'IA. D'autres cabinets, tels qu'Accenture et Capgemini, intègrent des services de conseil en technologie, aidant les entreprises à exploiter leurs données et à adopter des technologies avancées pour améliorer leur performance et compétitivité.

Parmi ces cabinets, on retrouve KPMG Algérie SPA, qui est l'une des branches de KPMG où notre stage a eu lieu. Nous avons eu l'opportunité de travailler en étroite collaboration avec les équipes du département Deal Advisory, nous permettant ainsi d'acquérir une compréhension approfondie des enjeux et des défis spécifiques auxquels font face les consultants en finance et conseil.

Les analystes financiers doivent régulièrement rechercher des informations sur les entreprises et les secteurs, résumer et analyser ces informations, puis raisonner. Ce processus complexe et chronophage est essentiel pour orienter les décisions d'investissement, élaborer des stratégies financières et mener à bien la Due Diligence.

Face aux contraintes liées au respect des politiques internes de sécurité et de confidentialité des données lors de leur traitement et/ou stockage, ainsi qu'à l'aspect "boîte noire" des solutions commerciales comme ChatGPT qui limitent le contrôle sur les modèles utilisés et les données d'entraînement, les entreprises doivent choisir entre adopter des solutions sur mesure adaptées à leurs pratiques locales ou utiliser des modèles commerciaux. Cette décision est cruciale pour tirer pleinement parti de la valeur ajoutée offerte par ces technologies de pointe.

C'est dans cette optique là que s'inscrit notre projet de fin d'études, en tentant de répondre à une question aussi technique que stratégique, à savoir : **« Comment concevoir un Assistant intelligent permettant d'améliorer la performance des consultants deal advisory au sein de KPMG ? »**

Afin de répondre à notre problématique, nous nous sommes appuyés sur l'IA générative, une approche novatrice qui utilise des modèles de fondation pour générer du contenu textuel de manière autonome. Cette approche a été complétée par l'intégration de diverses technologies avancées et méthodologies comme le RAG pour développer un assistant intelligent efficace et performant. Cette combinaison nous a permis de concevoir une solution innovante et adaptée aux besoins spécifiques des consultants en TS au sein de KPMG.

Notre travail a été articulé autour de trois chapitres et structurés de la façon suivante :

Premier chapitre : État des lieux

Dans le premier chapitre, nous nous sommes focalisés sur notre cas d'étude : KPMG Algérie, en présentant le cabinet avec ses différents départements. Nous y avons conduit une analyse SWOT afin de déceler les différentes forces, opportuni-

tés, faiblesses et menaces du cabinet de conseil pour bien cerner la problématique et ainsi l'axe principal sur lequel nous devions travailler afin que notre passage ait le plus de valeur ajoutée. Nous avons ainsi trouvé en la conception d'un assistant intelligent un réel potentiel d'avantage concurrentiel, de réduction des coûts et des délais.

Deuxième chapitre : État de l'art

Lors de ce second chapitre, nous avons entrepris de présenter, introduire et définir de manière exhaustive les concepts clés indispensables pour la conception de notre projet et la compréhension de notre démarche de résolution. Parmi ces concepts figurent l'Intelligence Artificielle Générative (GenAI), les modèles de langage (LMs) et en particulier les LLMs (Large Language Models) et les GPTs (Générative Prétrained Models), et enfin le RAG (Retrieval Augmented Generation).

Troisième chapitre : Résolution de la problématique

Dans ce dernier chapitre, nous avons présenté en détail l'assistant intelligent que nous avons développé. Nous avons exploré les différentes approches et stratégies employées pour sa conception, ainsi que les évaluations effectuées pour garantir son efficacité et sa pertinence.

Nous clôturons notre travail par une conclusion résumant l'ensemble du projet, de ses apports et de sa démarche.

Chapitre 1

Etat des lieux

Dans ce chapitre, nous nous focalisons sur le cabinet de conseil KPMG plus particulièrement sur le cabinet de conseil sur lequel s'est porté notre étude : KPMG Algérie. Nous allons présenter ce dernier avec les différents départements et équipes qui le constituent, en l'occurrence la structure qui nous a accueillie à savoir le Deal Advisory. Par la suite, nous procéderons à la description et modélisation du processus de Due Diligence qui est le cœur de ce métier, suivi par un diagnostic interne et externe afin que nous puissions détecter les différents dysfonctionnements survenus lors du déroulement de ce dernier. Enfin, après avoir cerné au mieux notre problématique, nous finirons par énoncer cette dernière.

1.1 Présentation de KPMG Global

Klynveld Peat Marwick Goerdeler, ou KPMG, l'un des quatre plus grands cabinets de conseil et d'audit dans le monde, aussi appelés Big Four, d'origine anglo-néerlandaise créé en 1987, dont le siège social est situé aux Pays Bas, à Amsterdam. KPMG est un réseau mondial de prestations de services d'audit (Audit), fiscaux (Tax) et de conseil (Advisory). Le cabinet accompagne ses clients à travers une offre pluridisciplinaire composée de divers métiers. Les secteurs d'activité s'étendent des technologies, au secteur public, en passant par les banques, les assurances ou encore les énergies et l'immobilier et autres.

FIG. 1.1: Quelques chifres sur KPMG Global, source : KPMG intranet

Le cabinet est présent dans 143 pays et emploie plus de 265 000 personnes à travers le monde comme le montre la figure KPMG chiffres et références est un des réseaux globaux d'audit et de conseil les plus étendus. Le nom et le logo KPMG sont des marques utilisées sous licence par les cabinets indépendants membres de l'organisation mondiale KPMG.

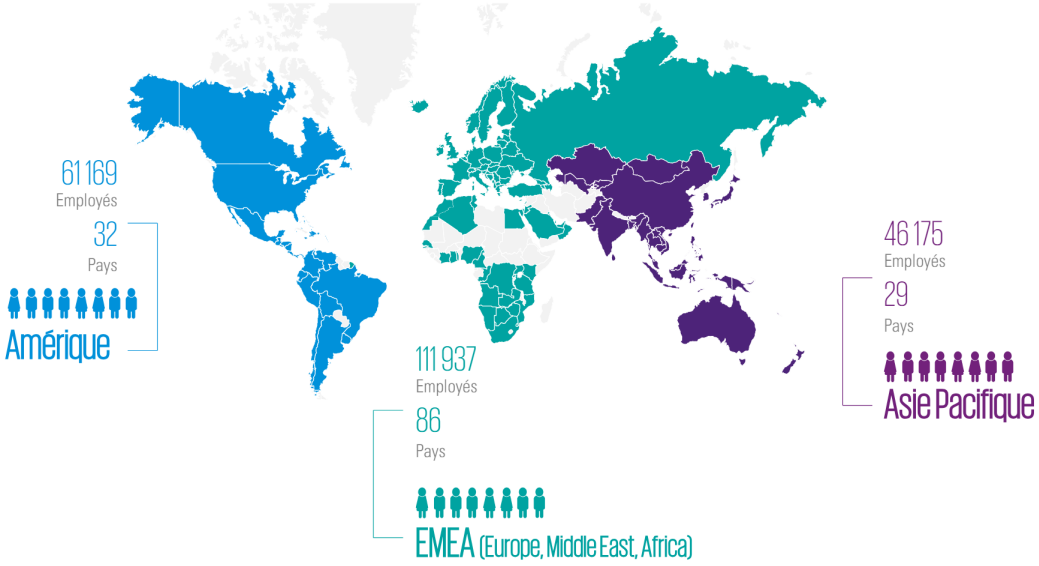


FIG. 1.2: Présence de KPMG dans le monde, Source : KPMG intranet, 2022

1.1.1 L'activité du réseau KPMG : Métier

KPMG propose une large gamme de services aux entreprises et organisations dans les domaines suivants : Advisory pour le conseil en entreprise, Audit et Tax & Legal.

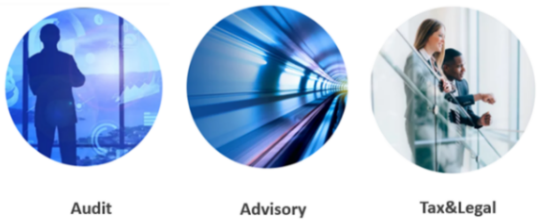


FIG. 1.3: Activités de KPMG

Advisory : Le service Advisory de KPMG propose des services de conseil stratégique et opérationnel pour accompagner les entreprises face aux défis complexes auxquels elles sont confrontées. Qu'il s'agisse d'élaborer de nouvelles stratégies commerciales, d'optimiser l'efficience opérationnelle, de gérer les risques ou d'explorer de nouvelles pistes de croissance, KPMG Advisory travaille en étroite collaboration avec ses clients afin de leur apporter des solutions innovantes et pragmatiques.

Audit : KPMG audit propose des services d'audit de premier ordre, aidant les entreprises à préserver la confiance des investisseurs, des autorités de régulation et du grand public. Le service Audit vise à examiner les états financiers et s'assurer de la conformité avec les normes comptables internationales et identifier les risques potentiels. Les équipes d'audit de KPMG allient expertise financière, méthodologie éprouvée et outils numériques avancés pour délivrer à ses clients un audit fiable et une validation indépendante de leurs comptes.

Tax Legal : Le service Tax Legal de KPMG offre des conseils fiscaux et juridiques intégrés pour aider les clients à naviguer dans le paysage fiscal et réglementaire complexe. Ce service fournit des conseils avisés sur la planification fiscale, la conformité réglementaire, les restructurations d'entreprise, les transactions internationales et bien plus encore pour les clients à minimiser les risques fiscaux, à optimiser leur efficacité fiscale et à atteindre leurs objectifs commerciaux.

1.1.2 KPMG Advisory

KPMG Advisory est un leader mondial dans la prestation de conseils stratégiques aux entreprises confrontées à des défis complexes et en évolution rapide,

KPMG Advisory propose un service de conseil en risque, en Deal et en management.

Deal Advisory Le Deal Advisory de KPMG propose une expertise de pointe dans l'accompagnement des opérations de transactions, fusions-acquisitions, évaluations d'entreprises et gestion des processus transactionnels. Ce service prodigue des conseils stratégiques à chaque étape clé, de la réalisation des audits d'acquisition initiaux jusqu'à la finalisation des accords. Ils œuvrent aux côtés des clients pour maximiser la valeur de leurs opérations de rapprochement et les aider à atteindre leurs objectifs business.

Risk Consulting KPMG Risk Consulting aide les entreprises à gérer et à atténuer les risques dans un environnement commercial en offrant des solutions personnalisées pour identifier, évaluer et gérer les risques opérationnels, financiers, réglementaires et technologiques, garantissant la résilience des clients et leur capacité à saisir les opportunités de croissance.

Management Consulting Le service de Management Consulting de KPMG offre des conseils stratégiques pour aider les entreprises à améliorer leur performance opérationnelle, leur efficacité organisationnelle et leur transformation numérique en développant des stratégies sur mesure, mettre en œuvre des initiatives de changement et optimiser les processus métier, permettant une croissance durable et une compétitivité accrue.

1.2 KPMG en algérie

KPMG Algérie S.P.A, société par actions, membre de l'organisation mondiale KPMG, constituée de cabinets indépendants affiliés à KPMG International Limited, une société de droit anglais (« private company limited by guarantee »), le cabinet propose des services d'audit, de conseil et d'expertise comptable. En tant que membre du réseau KPMG International, KPMG Algérie agit en tant que filiale de KPMG France, avec laquelle elle collabore pour déléguer certaines missions afin de renforcer l'équipe algérienne. Cette collaboration lui permet d'optimiser les coûts de production et d'acquérir un avantage concurrentiel sur le marché.

KPMG Algérie existe sous la forme d'une société de droit Algérien depuis 2002. En mai 2009, pour se rapprocher de ses clients de l'ouest, KPMG Algérie a ouvert un bureau à Oran. Le cabinet a bien évolué pendant ces dernières années et compte plus de 300 experts, formés aux normes du label KPMG, dans les disciplines de la finance, du conseil, des systèmes d'information et du droit/fiscalité des entreprises. KPMG Algérie est sollicitée par les investisseurs pour environ un sur deux de leurs

FIG. 1.4: Quelques chiffres sur KPMG Algérie, Source : KPMG intranet

projets d'investissement.

1.2.1 Organigramme

Le cabinet a son siège principal à Alger, mais dispose également d'une petite division à Oran. Cette antenne a été créée dans le but d'étendre les activités de la société dans la région ouest du pays. Elle fournit des services d'expertise comptable et fiscale pour le compte du siège d'Alger, l'organigramme est bien représenté sur la figure 1.5. Cependant, comme notre stage s'est déroulé au sein de la structure basée à Alger, nous nous concentrerons principalement sur celle-ci.

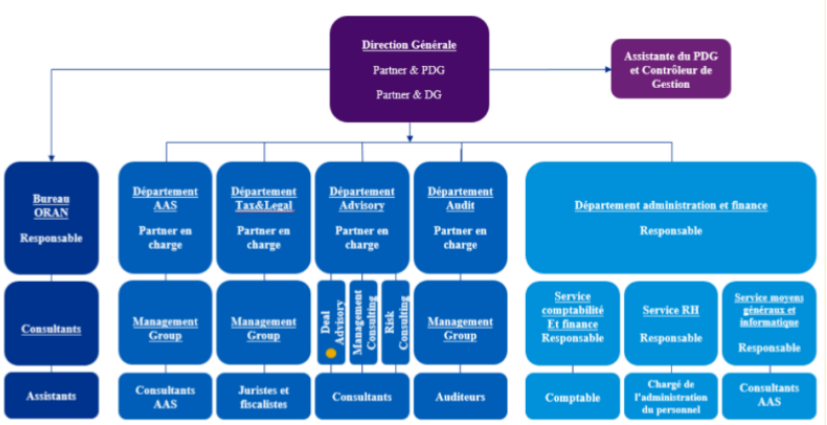


FIG. 1.5: Organigramme de KPMG Algérie, source : intranet KPMG interne

1.2.2 Présentation du département Deal Advisory

Le département Deal Advisory a été créé en avril 2017. Son rôle est de gérer les transactions de fusions et acquisitions pour les activités internationales de KPMG Algérie SPA, en collaboration avec KPMG France, ainsi que pour les missions locales.

L'équipe Deal Advisory accompagne les entreprises tout au long des différentes étapes des opérations suivantes : recherche de cibles, évaluation financière, due diligence, élaboration de business plans et revue des contrats d'acquisition ou de cession. De plus, cette équipe gère les situations de fraudes et de litiges pouvant affecter le bon déroulement d'une transaction, que ce soit du côté de l'acquéreur (Buy Side) ou de l'entreprise cible (Sell Side). Les missions sont classées en trois catégories selon la taille de l'entreprise cliente ou son type d'activité :

- Missions CORE : Elles concernent les grandes entreprises nécessitant un temps important et un volume de données conséquent. Ces missions peuvent s'étendre sur plusieurs mois.
- Missions SMID (Small and Medium Industrial/Institutional Deals) : Elles s'adressent aux startups, petites et moyennes entreprises. La durée de ces missions est d'environ deux semaines.
- Missions Real Estate : Elles regroupent l'ensemble des missions liées au secteur immobilier.

Ce service est réparti comme le montre la figure 1.6 en quatre divisions principales : Transaction services (TS), Recherche & stratégie(R&S), Deal Analytics (D&A) et Deal Tech.

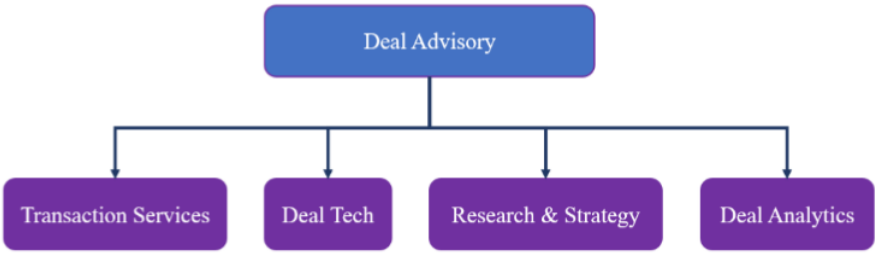


FIG. 1.6: Présentation des équipes Deal

Transaction Services (TS) L'équipe Transaction Services (TS) se concentre sur les aspects financiers et opérationnels des transactions. Leur objectif est de

minimiser les risques et de maximiser la valeur pour les clients. Les services offerts comprennent :

- Due Diligence : Analyse approfondie des états financiers et opérationnels d'une entreprise cible pour identifier les risques potentiels.
- Evaluation Financière : Détermination de la valeur d'une entreprise ou d'un actif spécifique, en utilisant des méthodes d'évaluation standards et personnalisées.
- Structuration des Transactions : Assistance dans la conception de la structure de la transaction pour optimiser les avantages financiers et fiscaux.

Recherche Stratégie (RS) L'équipe Recherche Stratégie (RS) est dédiée à l'analyse stratégique et à la recherche de cibles ou d'opportunités d'investissement. Leurs services incluent :

- Analyse Stratégique : Evaluation des opportunités de marché, des tendances sectorielles et des défis concurrentiels.
- Recherche de Cibles : Identification et évaluation de cibles potentielles pour des fusions ou acquisitions.
- Développement de Business Plans : Création de business plans détaillés pour soutenir les décisions d'investissement et les stratégies de croissance.

Deal Analytics (DA) L'équipe Deal Analytics (DA) utilise des outils analytiques avancés pour fournir des insights basés sur les données, améliorant ainsi la prise de décision tout au long du processus de transaction. Leurs services comprennent :

- Analyse de Données : Utilisation de l'analyse des données pour identifier des tendances, des risques et des opportunités cachées.
- Modélisation Financière : Création de modèles financiers complexes pour évaluer l'impact des différentes options stratégiques.
- Tableaux de Bord : Développement de tableaux de bord interactifs pour suivre les indicateurs clés de performance en temps réel.

Deal Tech L'équipe Deal Tech est spécialisée dans l'intégration de technologies de pointe pour améliorer l'efficacité et la précision des transactions. Leurs services incluent :

- Intégration Technologique : Mise en œuvre de solutions technologiques pour automatiser les processus transactionnels et améliorer la collaboration.
- Sécurité des Données : Garantir la sécurité des informations sensibles tout au long de la transaction.
- Innovations Technologiques : Adoption de nouvelles technologies comme l'intelligence artificielle et la blockchain pour optimiser les transactions.

1.3 La Deal Stratégie au sein de KPMG

1.3.1 Définition de la deal stratégie

Les opérations de fusion-acquisition (F&A) constituent un outil stratégique essentiel pour favoriser la croissance d'une entreprise. Les F&A permettent d'accéder plus rapidement à de nouveaux segments de marché, à des canaux de vente et de distribution inédits, ainsi qu'à des compétences nouvelles. Elles offrent également l'opportunité d'améliorer les infrastructures et processus, tout en réduisant les dépenses. Néanmoins, il est crucial que toute entité acquise s'aligne sur les objectifs stratégiques globaux et les soutienne, afin d'assurer une croissance pérenne.

L'équipe KPMG spécialisée en stratégie transactionnelle peut proposer une approche globale qui remet en question les idées reçues. Elle s'appuie sur une expertise sectorielle et des analyses fondées sur les données pour appréhender, interroger et concrétiser l'adéquation stratégique et la valeur des décisions d'investissement à diverses étapes du cycle de vie d'une transaction.

1.3.2 Importance de la deal stratégie

Pour optimiser la valeur et maximiser les synergies des F&A, il est nécessaire d'élaborer une stratégie appropriée, en phase avec les objectifs globaux. Une méthodologie efficace associe une connaissance approfondie du secteur et des insights pertinents, à des analyses du potentiel du marché, de l'environnement concurrentiel et du positionnement. On y ajoute un examen objectif du business plan de l'entreprise pour répondre à des questions stratégiques cruciales telles que :

- Quelle est la stratégie la plus adaptée pour pénétrer le marché ?
- L'entreprise doit-elle développer en interne, nouer des partenariats ou procéder à une acquisition ?

- La cible est-elle en adéquation stratégique avec l'activité existante et les ambitions de croissance ?
- Le marché sur lequel évolue la cible présente-t-il un potentiel de croissance intéressant ?
- Le modèle économique de la cible est-il viable et durable sur le long terme ?
- L'entreprise devrait-elle se séparer de divisions non stratégiques ou sous-performantes ?

1.3.3 Dans quels cas la deal stratégie est-elle pertinente ?

Les services de stratégie transactionnelle sont généralement mis en œuvre à deux moments clés du processus de F&A :

1. **Accompagnement pre-deal** : KPMG assiste dans l'identification et la sélection des marchés potentiels à conquérir. On combine une analyse stratégique du marché avec une compréhension de la stratégie pour identifier et évaluer les options stratégiques envisageables pour l'entrée sur le marché.
2. **Accompagnement durant le deal** : Une fois la cible identifiée, on peut fournir un niveau d'analyse supplémentaire pour challenger l'hypothèse de création de valeur et d'adéquation stratégique, et réaliser une due diligence pour garantir la solidité de la stratégie transactionnelle. Les services peuvent inclure la due diligence commerciale côté acquéreur, le benchmarking et les études de cas, l'examen du business plan et les évaluations de création de valeur.

1.3.4 La phase Pre-Deal

L'évaluation pré-transaction implique une analyse initiale approfondie d'une entreprise cible afin d'identifier les principaux risques et obstacles potentiels à la conclusion de l'affaire. Cette phase utilise principalement des informations disponibles publiquement et se concentre sur les éléments suivants :

1. **Business Overview (Aperçu de l'entreprise)**
    - Description de l'activité : Une description détaillée des opérations et des activités principales de l'entreprise.
    - Principales étapes historiques : Les jalons clés et les développements historiques de l'entreprise.

- Implantation géographique : La présence géographique de l'entreprise et ses marchés principaux.
- Offre de produits : Les produits et services proposés par l'entreprise.

2. **Market Overview (Aperçu du marché)**

   - Tendances du marché : Les tendances actuelles du marché dans lequel l'entreprise opère.
   - Évolution de la taille du marché : La croissance et les changements dans la taille du marché.
   - Facteurs de croissance : Les moteurs et les facteurs influençant la croissance du marché.

3. **Benchmark (Analyse comparative)**

   - Principaux concurrents : Identification des principaux concurrents de l'entreprise cible.
   - Positionnement des concurrents : Le positionnement stratégique des concurrents sur le marché.

4. **Performance financière**

   Analyse des performances financières passées et actuelles de l'entreprise cible, incluant les revenus, les bénéfices, et d'autres indicateurs financiers pertinents.

5. **Évaluation**

   - Analyse comparative : Comparaison avec des entreprises similaires pour évaluer la valeur de l'entreprise cible.
   - Analyse transactionnelle : Étude des transactions similaires récentes pour évaluer la valeur potentielle de l'entreprise cible.

Cette analyse est particulièrement pertinente pour les investisseurs ou acquéreurs potentiels qui souhaitent obtenir une compréhension initiale de l'entreprise cible avant de décider d'engager une due diligence complète

La figure 1.7 illustre la phase pre-deal dans le cycle de vie des services de transaction. Elle montre également les différents livrables (discussion documents) à être réalisé par les consultants.

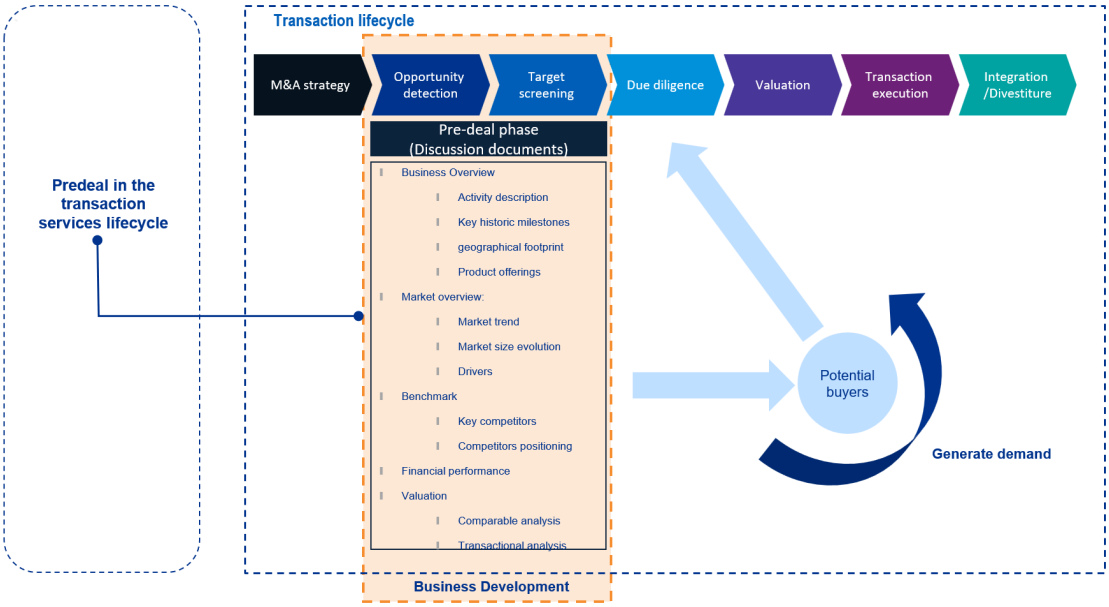


FIG. 1.7: Phase pre-deal dans le cycle de vie des services de transaction, source : document interne

1.3.5 Pourquoi se concentrer sur la création de valeur ?

L'efficacité des initiatives de création de valeur dans les opérations de capital-investissement est directement liée à leur mise en œuvre précoce. Des procédures de due diligence plus approfondies, particulièrement en amont de la transaction, permettent d'identifier et d'évaluer les principaux facteurs de croissance et de coûts qui étayent une thèse d'investissement, facilitant ainsi la prise de décision quant à la poursuite ou non de l'opération. L'enquête M&A de KPMG 2024 révèle que 52% des acteurs du capital-investissement interrogés estiment que le contexte économique actuel a renforcé l'importance de la due diligence et de l'évaluation des risques.

Dans l'éventualité où la transaction se concrétise, l'élaboration de plans de création de valeur élaborés - impliquant non seulement le partenaire de la transaction mais aussi le partenaire opérationnel responsable de la stratégie - permet la mise en place d'actions spécifiques dès la finalisation de l'accord. En initiant le processus plus tôt et en mettant l'accent sur la question de la valeur dès le départ, on assure l'alignement de toutes les parties prenantes sur les objectifs à atteindre, permettant ainsi une mise en œuvre immédiate dès le jour de la conclusion de l'accord.

L'identification précoce des facteurs de création de valeur permet aux fond

d'investissments de prioriser et de planifier une action immédiate, comme illustré dans le figure 1.8.

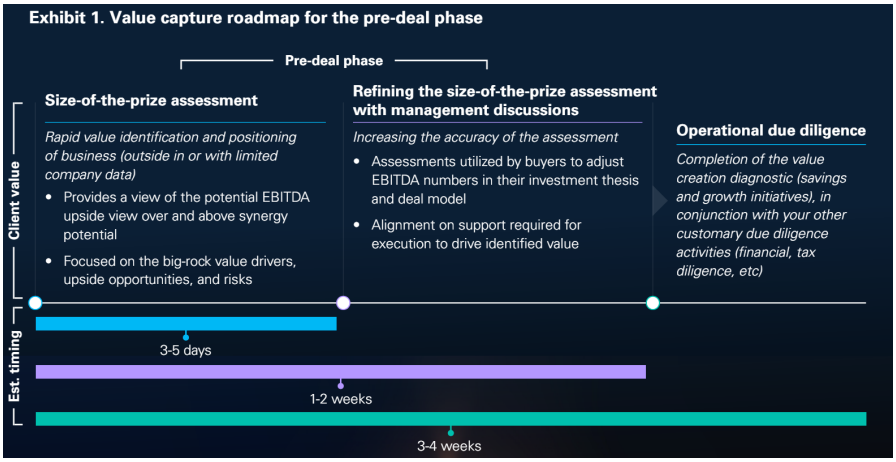


FIG. 1.8: Feuille de route de capture de valeur pour la phase de pré-transaction, source document KPMG

1.3.6 La phase de Due Diligence

L'évaluation financière constitue l'élément central de l'étude d'opportunité lors d'une fusion ou d'une acquisition. Cette phase débute par le processus de Due Diligence, dont KPMG propose deux variantes principales

Chez KPMG, la Due Diligence a pour objectif de fournir à l'acquéreur potentiel des informations cruciales sur les cibles envisagées. Cette analyse approfondie couvre les aspects financiers, juridiques, opérationnels et les risques inhérents à l'activité de la cible. Au terme de ce processus, une valorisation de l'entreprise ciblée est établie, prenant en compte divers paramètres, ce qui facilite la conclusion de la transaction avec l'accord de toutes les parties impliquées.

Parmi les sources de données communément utilisées permettant d'extraire de telles informations sur la santé financière d'une entité, on trouve les rapports annuels de cette dernière. Hormis le fait que ces rapports soient publics, ils comportent tout de même diverses informations stratégiques et financières.

KPMG réalise généralement deux types de due diligence :

- Pour les clients vendeurs, appelée Vendor Due Diligence (VDD)

- Pour les clients acquéreurs, appelée Buyer Due Diligence (BDD), également connue sous le nom de Financial Due Diligence (FDD).

1. Vendor Due Diligence (VDD) :

Dans le cadre de la VDD, l'analyse se concentre sur l'activité commerciale et le processus est mandaté par le vendeur.

Elle se caractérise par son objectivité et sa conformité aux engagements pris avec le client. Elle englobe les aspects financiers, commerciaux et opérationnels critiques de l'entreprise, adoptant la perspective d'un acquéreur potentiel. Les rapports sont validés par la direction pour garantir leur fiabilité, et le vendeur est tenu informé de l'avancement du processus. Cette approche facilite la formulation d'offres par les acquéreurs potentiels.

2. Buyer Due Diligence (BDD) :

La FDD, souvent appelée Financial Due Diligence, est le moyen privilégié pour un acquéreur, qui la mandate auprès de KPMG, d'obtenir des informations essentielles sur une entreprise en vue de son acquisition.

Elle revêt une importance capitale pour trois raisons principales :

1. Elle fournit des informations essentielles pour l'évaluation financière, offrant une vision claire des sources de revenus et de création de valeur de la cible. Elle identifie des métriques clés telles que la qualité des revenus, illustrée notamment par l'EBITDA, et établit le lien entre l'EBITDA et les flux de trésorerie pour affiner le modèle d'évaluation financière.

2. Elle permet d'évaluer la santé financière de la cible en analysant ses sources de revenus.

3. Elle offre une compréhension approfondie de la relation entre l'entreprise et ses capitaux propres, permettant ainsi d'évaluer sa solvabilité.

Ces trois aspects contribuent à l'évaluation des synergies potentielles entre l'acquéreur et sa cible, dans l'optique de les rendre aussi explicites que possible et de créer de la valeur pour les deux parties impliquées.

1.3.7 Modélisation du processus de Due Diligence

Les activités du département Transaction Services durant la phase de due diligence peuvent être résumées par le processus suivant, conçu à l'aide de l'outil

Camunda pour modéliser les processus d'affaires avec la méthode BPMN 2.0. Ce processus illustre les deux types de due diligence expliquées précédemment.

Ce macro-processus engendre plusieurs processus, illustrés dans la figure 1.9, et couvrant les étapes suivantes :

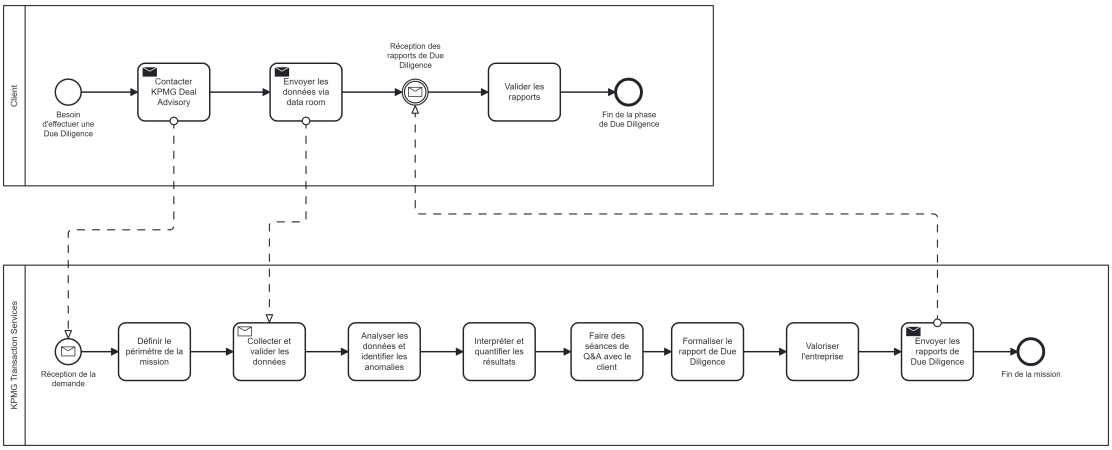


Fig. 1.9: Processus de Due Diligence

1. Définition du périmètre de la mission :

Cette phase revêt une importance particulière car elle établit clairement les contours de l'étude de la transaction pour répondre aux attentes du client. Comprendre le client, son marché et ses activités est essentiel. Ensuite, définir le contexte de la transaction et de la mission permet d'organiser le travail en interne et de répartir les tâches entre les parties prenantes concernées.

2. Collecte et validation des données :

Après avoir défini le scope de la mission, le client se charge d'envoyer l'ensemble des données nécessaires à la mission de Due Diligence via une data room qui est une plateforme en ligne sécurisée permettant de stocker et partager des informations confidentielles, afin de faciliter la collecte et l'accès aux données. Les données envoyées peuvent se présenter sous forme FEC (Fichier des Ecritures comptables), GL(Grand livres) ,états financiers, de rapports annuels, de rapports CAC, d'entretiens, d'enquêtes ou même d'études de marché. Ces données doivent être validées par un auditeur pour garantir son exactitude, son exhaustivité et sa fiabilité. Sans cela, aucune analyse ne peut être effectuée, car le moindre "Red Flag" signalé dans le rapport d'audit bloquerait la mission.

3. Analyse de données et identification des synergies :

Une fois la phase préliminaire achevée, avec des résultats validés par le client, l'évaluation des risques et la quantification des synergies attendues par les parties de la transaction seront prioritaires. Un suivi rigoureux de la qualité du travail est essentiel, en conformité avec les référentiels internes définis par l'organisation KPMG.

4. Interprétation et quantification des résultats :

Une fois la phase précédente achevée, il sera crucial d'interpréter les résultats obtenus en mesurant de manière précise l'impact des synergies entre les deux entités. Cela implique la création de modèles prédictifs basés sur divers indicateurs et objectifs convenus préalablement par les parties. Forte de son expérience, KPMG utilisera ses référentiels internes pour évaluer ces indicateurs et développer des modèles alignés sur ses travaux antérieurs, tout en s'adaptant au contexte du marché.

5. Séances Q&A avec le client

Les sessions de questions-réponses (Q&A) sont essentielles pour approfondir la compréhension de l'entreprise cible. Elles permettent à l'acquéreur de valider sa thèse d'investissement, d'identifier les risques potentiels, d'évaluer la qualité du management, de clarifier les données financières et opérationnelles, de négocier des conditions plus favorables et de faciliter la planification de l'intégration. En posant des questions ciblées, l'acquéreur peut découvrir des informations cruciales sur les responsabilités juridiques, les défis opérationnels ou les incertitudes du marché, qui pourraient influencer l'évaluation ou la structure de l'opération. Ces sessions contribuent ainsi à une prise de décision plus éclairée, à une meilleure atténuation des risques et à la préparation d'une intégration post-acquisition réussie.

6. Formalisation du rapport de due diligence

Après avoir mené plusieurs phases d'étude de l'opportunité d'investissement potentiel et validé celui-ci, la phase de due diligence peut commencer. Elle englobe plusieurs aspects tels que le commercial, l'environnemental, le financier et le fiscal, entre autres. La due diligence financière revêt une importance primordiale dans ce processus, car elle fournit des indicateurs financiers essentiels sur l'entreprise concernée, impactant ainsi l'évaluation et la décision d'acquisition en fonction de la santé financière de l'entité.

1.4 Diagnostic

Suite à la présentation de l'entreprise, une analyse interne et externe a été réalisée afin de détecter les forces et faiblesses de l'entreprise en interne, ainsi que les opportunités et menaces en externe.

1.4.1 Analyse interne

Forces :

- Compétences du personnel : KPMG Algérie opérant en off-shore et en étroite collaboration avec KPMG France et KPMG International, elle bénéficie de toute l'expertise internationale de ses consultants qui ont la possibilité de se former constamment au cours des différents échanges lors des missions.
- L'ambiance de travail au sein de KPMG Algérie est particulièrement joviale et conviviale, ce qui a pour effet de créer une synergie de groupe rendant les collaborateurs très efficaces.
- Ressources financières : En tant que cabinet d'audit et de conseil de premier plan, KPMG dispose de ressources financières considérables qui peuvent être investies dans l'innovation et le développement de nouvelles technologies
- Utilisation de technologies de pointe : KPMG est connu pour adopter des technologies innovantes pour améliorer ses services telle que la RPA.

Faiblesse :

- Un manque de ressources humaines entraîne le rejet de nombreuses opportunités de missions obtenues par les partenaires du cabinet. Cette menace est particulièrement préoccupante dans le modèle économique des cabinets de conseil, où le capital humain est la principale source de production.
- La survenue de turnover important au sein des équipes à certaines périodes accentue le manque de ressources disponibles, attribuable en partie à la charge de travail élevée et à la nature rébarbative des tâches. Cette conjoncture contribue à aggraver la pression sur les effectifs disponibles
- Les coûts parfois élevés de la phase prospection pre deal
- Le risque d'erreurs est élevé en raison de l'implication de juniors et de stagiaires sur des données de grande importance et de grande ampleur. Cela nécessite une supervision constante des seniors pour effectuer une vérification supplémentaire, entraînant ainsi un double travail.

1.4.2 Analyse externe

Menaces :

- La concurrence face aux cabinets de conseil, notamment les big four, est très forte en Algérie ainsi qu'ailleurs dans le monde, pour cette raison KPMG doit se démarquer par sa proposition de valeur.
- L'émergence de nouvelles technologies pourrait perturber les services de conseil traditionnels si KPMG ne s'adapte pas assez rapidement.
- Risques liés à la confidentialité et à la sécurité des données : les consultants seraient susceptibles de recourir à des solutions externes pouvant créer une faille de sécurité par rapport aux données clients.

Opportunités :

- KPMG pourrait se positionner en tant que pionnier dans le domaine du conseil en fusion-acquisition, renforçant ainsi sa position concurrentielle sur le marché.
- Émergence d'un ensemble de théories et techniques pour améliorer les processus et services du Deal Advisory .
- Le développement de services innovants basés sur la technologie peut aider KPMG à se différencier et à stimuler sa croissance.

1.4.3 Synthèse de l'Analyse SWOT :

En conclusion, l'analyse SWOT du département Deal Advisory de KPMG met en lumière ses points forts, tels que sa réputation de marque solide et son expertise en conseil financier, ainsi que ses défis, notamment la dépendance à l'expertise humaine et les retards potentiels dans le traitement des données. Cependant, cette analyse révèle également des opportunités significatives, telles que la croissance du marché des fusions-acquisitions et la demande croissante de solutions basées sur les dernières technologies. Toutefois, ces opportunités s'accompagnent de menaces telles que la concurrence accrue et les risques liés à la sécurité des données. La figure 1.10 récapitule l'ensemble de ces points sous forme de matrice SWOT.

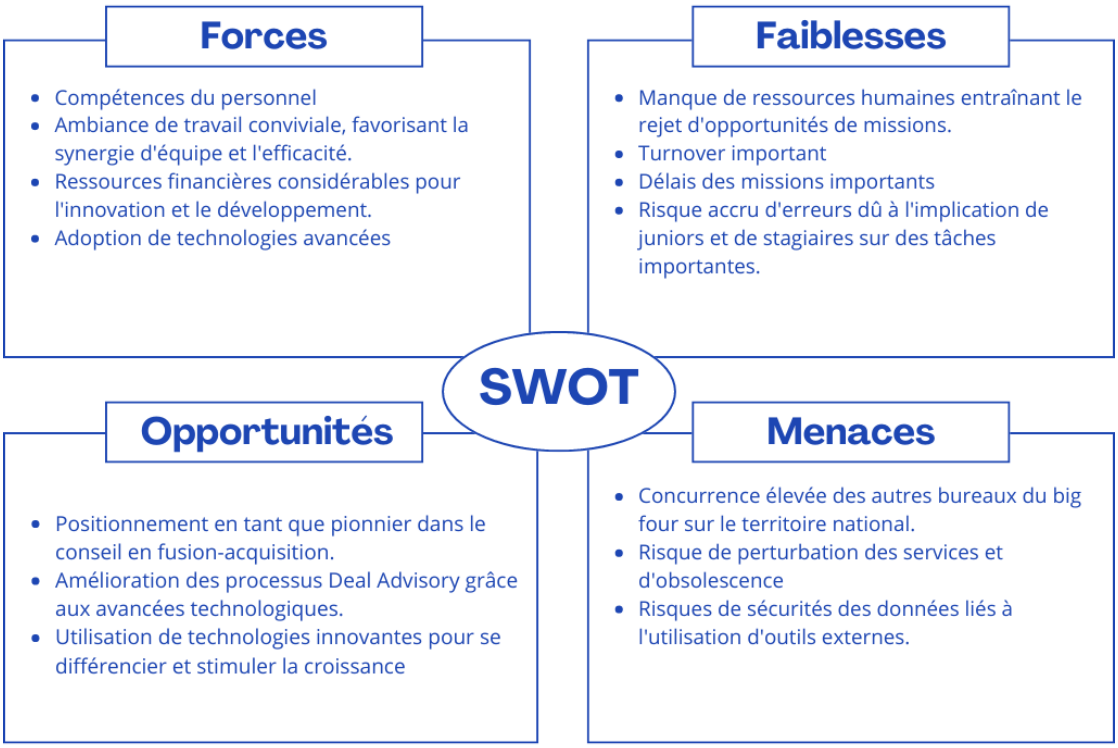


Fig. 1.10: Matrice SWOT

1.5 Constat et dysfonctionnements

Après avoir réalisé le diagnostic externe et interne, nous sommes arrivés à faire un constat et une synthèse des dysfonctionnements au sein de KPMG, il est clair que l'entreprise fait face à plusieurs défis qui entravent son efficacité opérationnelle et sa compétitivité sur le marché.

Ces lacunes se manifestent à différents niveaux :

- Des difficultés à traiter de gros amas de données structurées et non structurées
- Des coûts élevés engendrés par le staffing de juniors, notamment si le deal n'aboutit par la suite
- Délais conséquents de réalisation de la due diligence et l'analyse financière

- Nécessité parfois de l'engagement de nombreux seniors et managers en plus des juniors
- Complexité croissante des produits financiers

Ces dysfonctionnements révèlent un besoin pressant de modernisation des pratiques opérationnelles, notamment à travers l'adoption de solutions innovantes telles qu'un assistant IA spécialisé qui répondra aux questions complexes, recherchera des insights pertinents, conseillera sur les meilleures stratégies à adopter et trouvera des leviers de création de valeur, le tout en dialoguant naturellement avec les utilisateurs.

Un tel outil pourrait non seulement optimiser les tâches répétitives et chronophages, libérant ainsi les ressources humaines pour des tâches à plus forte valeur ajoutée, mais il pourrait également ingérer et analyser en d'immenses quantités de données financières, juridiques et commerciales, structurées et non structurées, afin d'en extraire des informations et ainsi améliorer la précision et la rapidité des analyses permettant à l'entreprise de prendre des décisions éclairées et proactives.

En conclusion, le développement d'un assistant IA apparaît comme une réponse stratégique indispensable pour surmonter les défis opérationnels actuels et positionner l'entreprise sur une trajectoire de croissance durable et compétitive dans un environnement commercial en constante évolution.

1.6 Problématique

Dans le contexte des fusions-acquisitions, les missions de due diligence revêtent une importance capitale pour les décisions des investisseurs. Au sein de KPMG, le département Deal Advisory est chargé de ces missions, bénéficiant d'une expertise métier approfondie dans le domaine des grandes transactions.

Au sein du Deal Advisory, les équipes de KPMG Algérie s'engagent à évoluer vers de nouveaux horizons afin de s'aligner avec les meilleures pratiques mondiales. C'est dans cette optique que notre intervention s'est déroulée au sein de ce service.

Après avoir identifié la nécessité de développer un assistant IA spécialisé, la problématique centrale devient alors : **« Comment concevoir un assistant intelligent basé sur l'IA générative permettant d'améliorer la performance et l'efficacité opérationnelle des Consultants lors des missions de Deal Advisory au sein de KPMG ? »**

Cette problématique se répartie sur plusieurs champs d'action et converge ainsi vers d'autres sous questions, à savoir :

- Quelle approche utiliser pour développer un assistant intelligent ?
- Comment évaluer la performance et la fiabilité du modèle ?
- Quels sont les challenges auxquels il faudra s'y opposer ?

Pour répondre à ces questions, il est indispensable d'adopter une approche méthodique et systématique.

1.7 Conclusion

Ce chapitre visait à contextualiser notre étude en présentant le cabinet KPMG, son environnement et ses activités.

Nous avons débuté en exposant KPMG International, en passant par KPMG Algérie SPA, puis en détaillant le département Deal Advisory, impliquant les équipes R&S, TS, D&A et Deal Tech. Ensuite, nous avons réalisé un diagnostic à travers une analyse SWOT pour identifier les dysfonctionnements de la firme et prioriser les interventions nécessaires. Cette analyse a mis en lumière les processus liés à la deal stratégie, notamment la phase pre-deal et la due diligence, comme étant les plus problématiques.

À travers des entretiens avec l'équipe TS et l'observation des activités des missions de value creation et de due diligence, nous avons obtenu une compréhension détaillée de celles-ci. Nous avons identifié des dysfonctionnements et des tâches à caractère complexe et chronophage, notamment la lecture et l'analyse des documents tels que les rapports financiers.

Suite à une étude sur l'optimisation potentielle, nous avons conclu qu'une intervention était nécessaire, conduisant à l'amélioration de la performance des consultants lors des missions de Deal Advisory. Nous avons également défini le contexte de l'étude pour proposer une solution axée sur l'implémentation d'un chatbot basé sur l'IA générative, répondant aux besoins du cabinet en termes d'efficacité.

Pour résoudre notre problématique, une revue de littérature exhaustive sur les travaux existants abordant des problèmes similaires sera abordée dans le prochain chapitre.

Chapitre 2

Etat de l'art

Ce second chapitre, crucial pour notre projet de fin d'études, définit les termes et concepts utilisés tout au long de notre travail. Il offrira une présentation détaillée des notions clés et terminologies nécessaires pour comprendre les étapes ultérieures. Bien que non exhaustive, la recherche bibliographique sera suffisamment approfondie pour garantir une compréhension cohérente du projet.

Nous mettrons l'accent sur les éléments suivants : l'intelligence artificielle classique et générative, le NLP, les modèles de fondation, en particulier les LLMs, et les approches d'amélioration comme le RAG.

2.1 L'évolution de l'intelligence artificielle

2.1.1 Intelligence Artificielle

Nous nous appelons Homo sapiens, "l'homme sage", en raison de l'importance de notre intelligence. Depuis des millénaires, nous cherchons à comprendre notre pensée. L'intelligence artificielle (IA) va plus loin, en visant à créer des entités intelligentes. Ce domaine, né après la Seconde Guerre mondiale et nommé en 1956, est l'un des plus récents en science et ingénierie. Contrairement à la physique, qui a déjà ses grands noms, l'IA offre encore de nombreuses opportunités pour des innovateurs. Elle couvre divers sous-domaines, de la perception à des applications spécifiques comme jouer aux échecs, écrire de la poésie, conduire une voiture et diagnostiquer des maladies. L'IA est pertinente pour toute tâche intellectuelle, en faisant un domaine véritablement universel [11].

Le tableau 2.1 présente huit définitions de l'IA selon deux dimensions : les proces-

sus de pensée et le raisonnement en haut, et le comportement en bas. À gauche, le succès est mesuré par la fidélité à la performance humaine, et à droite, par la rationalité, une mesure idéale de performance. Un système est rationnel s'il fait la "bonne chose" selon ses connaissances. Historiquement, ces quatre approches de l'IA ont été développées par différentes personnes avec diverses méthodes. L'approche centrée sur l'humain est une science empirique impliquant des observations et des hypothèses sur le comportement humain. L'approche rationaliste combine mathématiques et ingénierie. Ces groupes ont à la fois critiqué et aidé les uns les autres [11].

Penser comme un humain :	**Penser rationnellement :**
"Le nouvel effort passionnant pour faire penser les ordinateurs... des machines avec des esprits, au sens plein et littéral." [13] "L'automatisation des activités que nous associons à la pensée humaine, telles que la prise de décision, la résolution de problèmes, l'apprentissage..." [14]	"L'étude des facultés mentales à travers l'utilisation de modèles computationnels." [15] "L'étude des calculs qui permettent de percevoir, raisonner et agir." [16]
Agir comme un humain :	**Agir rationnellement :**
"L'art de créer des machines qui accomplissent des fonctions nécessitant de l'intelligence lorsqu'elles sont réalisées par des humains." [17] "L'étude de la manière de faire faire aux ordinateurs des choses que, pour le moment, les humains font mieux." [18]	"L'intelligence computationnelle est l'étude de la conception d'agents intelligents." [19] "L'IA s'intéresse au comportement intelligent dans les artefacts." [20]

Tab. 2.1: Définitions de l'Intelligence Artificielle

1. Le Machine Learning

Le machine learning (apprentissage automatique en français) est un domaine de l'intelligence artificielle qui permet aux systèmes informatiques d'apprendre et de s'améliorer à partir de données, sans être explicitement programmés. Au lieu

d'écrire des instructions pour chaque tâche, on fournit au système des exemples à partir desquels il peut détecter des modèles et faire des prédictions ou prendre des décisions sans intervention humaine supplémentaire.

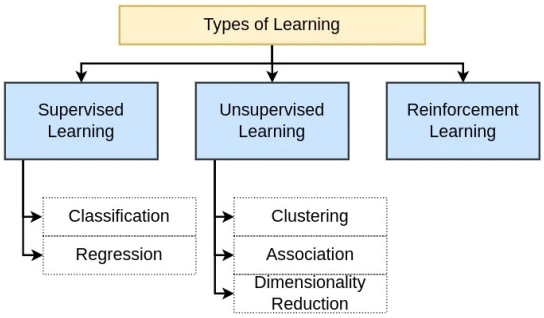


FIG. 2.1: Les types d'apprentissage Machine

Comme le montre la figure 2.1 Il existe plusieurs types d'apprentissage dans le machine learning :

1. Apprentissage supervisé : On fournit au système des exemples étiquetés (inputs et outputs désirés). Il apprend alors à faire une prédiction ou une classification sur de nouveaux exemples. C'est utilisé par exemple pour la reconnaissance d'images, la détection de spam, etc.
2. Apprentissage non supervisé : On ne fournit que des inputs au système qui doit alors trouver lui-même une structure dans les données, sans labels. Utilisé pour le clustering, la réduction de dimensionnalité, la détection d'anomalies, etc.
3. Apprentissage par renforcement : Le système apprend par essais/erreurs en interagissant avec un environnement pour maximiser une récompense. Utilisé en robotique, dans les jeux, etc.

Le machine learning a de très nombreuses applications dans tous les secteurs, tant qu'il y a des données à analyser et des décisions à prendre de façon intelligente.

2. Deep Learning

Le Deep learning ou apprentissage profond est l'une des technologies principales du Machine learning. Avec le Deep Learning, nous parlons d'algorithmes capables de mimer les actions du cerveau humain grâce à des réseaux de neurones artificielles. Les réseaux sont composés de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente.

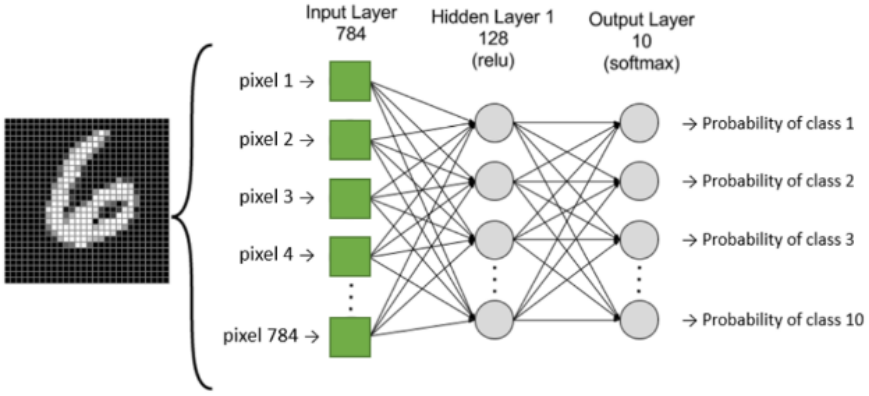


FIG. 2.2: Architecture des modèle de Deep Learning : Réseau de neurones

2.1.2 Intelligence Artificielle Générative

L'IA générative désigne une branche de l'intelligence artificielle qui se concentre sur la création de modèles et d'algorithmes capables de générer du contenu nouveau et original, comme des images, des textes, de la musique et même des vidéos. Contrairement aux modèles d'IA traditionnels, qui sont entraînés pour accomplir des tâches spécifiques, les modèles d'IA générative visent à apprendre et à imiter les motifs des données existantes pour produire des créations uniques [22].

La figure 2.3 illustre les différents domaines de l'IA, en l'occurence la GenAI.

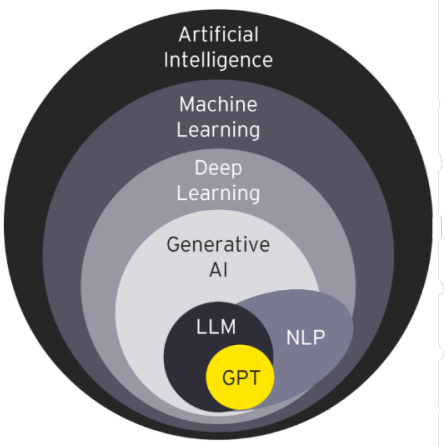


FIG. 2.3: Domaines de l'IA, source : [10]

1. Modèles de fondation (IA Générale)

Les modèles de fondation, également connus sous le nom d'IA générale (IAG), ou encore en anglais GPAI (General-Purpose AI), forment la base de nombreuses applications, y compris ChatGPT d'OpenAI, Bing de Microsoft, et de nombreux chatbots de sites web. Ils soutiennent également de nombreux outils de génération d'images comme Midjourney ou les outils de remplissage génératif d'Adobe Photoshop [23].

Le terme "modèles de fondations" a été popularisé en 2021 par des chercheurs de l'Institut Stanford pour l'IA centrée sur l'humain, en collaboration avec le Centre Stanford pour la Recherche sur les Modèles de Fondation [26]. Ces chercheurs ont défini les modèles fondamentaux comme des "modèles entraînés sur des données larges (généralement en utilisant l'auto-supervision à grande échelle) qui peuvent être adaptés à une vaste gamme de tâches en aval."[26] [23]

Les modèles fondamentaux sont des modèles d'IA conçus pour produire une grande variété de résultats. Ils sont capables d'exécuter diverses tâches et applications, comme la génération de texte, d'images ou d'audio. Ils peuvent fonctionner comme des systèmes autonomes ou servir de base pour de nombreuses autres applications.[23]

Les chercheurs suggèrent que la définition de "général" fait référence à la portée des capacités des modèles fondamentaux, à la diversité de leurs utilisations, à la variété des tâches qu'ils peuvent accomplir et aux types de sorties qu'ils peuvent produire. Certains modèles fondamentaux peuvent traiter des entrées dans une seule modalité, comme le texte, tandis que d'autres sont multimodaux et capables de traiter plusieurs modalités d'entrée simultanément, par exemple, le texte, les images, les vidéos, etc., et de générer plusieurs types de sorties (comme générer des images, résumer du texte, répondre à des questions) basées sur ces entrées.[23]

Par exemple, l'application populaire ChatGPT est construite sur les familles de modèles fondamentaux GPT-3.5 et GPT-4. Ces familles de modèles sont également utilisées comme base pour d'autres applications, telles que Bing Chat et Duolingo Max.

Une caractéristique distinctive des modèles fondamentaux est l'ampleur des données et des ressources informatiques nécessaires à leur construction. Ils nécessitent des ensembles de données contenant des milliards de mots ou des centaines de millions d'images récupérées sur internet. Les modèles fondamentaux reposent également sur le "transfer learning", c'est-à-dire l'application de schémas appris d'une tâche à une autre.[23][22]

2. Types de modèles de fondations

Les modèles de fondations sont variés et comprennent notamment les réseaux adverses génératifs (GANs), les modèles de langage de grande taille (LLMs) basés sur des transformateurs, les modèles en Vision par Ordinateur, et les modèles multimodaux [23]. Le tableau 2.2 illustre cela plus en détails.

Type de modèle fondamental	**Description**	**Exemples et cas d'utilisation principaux**
Modèles de vision par ordinateur (Computer Vision Foundation Models)	Modèles utilisés pour analyser et interpréter des images et des vidéos.	Exemple : Florence, utilisé pour des tâches comme la description d'image, le questionnement visuel, etc.
Modèles multimodaux (Multimodal Foundation Models)	Modèles combinant plusieurs types d'entrées, comme le texte et l'image, pour établir des corrélations sémantiques.	Exemple : UniLM, utilisé pour l'IA documentaire, combinant la vision par ordinateur et le traitement du langage naturel.
Réseaux adverses génératifs (GANs)	Modèles impliquant deux réseaux neuronaux en compétition pour générer de nouvelles données similaires aux données d'entraînement.	Exemples : Création d'images, génération de données synthétiques, amélioration d'images astronomiques.
Modèles de langage de grande taille basés sur les transformateurs (Transformer-Based Large Language Models, LLMs)	Modèles de langage utilisant des réseaux de neurones profonds pour traiter et générer du texte.	Exemples : GPT-3.5, GPT-4, utilisés dans des applications comme ChatGPT, Bing Chat, Duolingo Max.

Tab. 2.2: Tableau comparatif des types de modèles de fondations

3. GenAI vs Traditional AI

Comme son nom l'indique, l'IA générative se distingue des formes précédentes d'IA par sa capacité à générer de nouveaux contenus, souvent sous des formes non structurées (texte écrit, images). Les modèles de fondation, comme les transformateurs, en sont la base et sont entraînés sur des ensembles de données vastes et variés. En revanche, les modèles d'IA traditionnels sont généralement limités à des ensembles de données spécifiques et à des tâches uniques, comme la classification

d'objets ou la prédiction.[45]

Les modèles de fondation, tels que les grands modèles de langage, peuvent accomplir plusieurs fonctions et générer du contenu. Cette polyvalence permet aux entreprises d'utiliser un même modèle pour divers cas d'utilisation, ce qui n'est pas possible avec les anciens modèles d'apprentissage profond.[45]

Cependant, les modèles de fondation actuels présentent des limites, comme des réponses erronées ("hallucinations", un défault que l'on expliquera plus en détails par la suite) et un manque d'explicabilité. Ils ne sont pas encore adaptés à l'analyse de grandes quantités de données tabulaires ou à la résolution de problèmes d'optimisation complexes.[45]

Le tableau 2.3 résume les principaux points abordés lors de cette comparaison.

Aspect	**IA Traditionnelle**	**IA Générative**
Type de Contenu	Structuré (tableaux, lignes, colonnes)	Non structuré (texte écrit, images)
Modèles Utilisés	Modèles spécifiques	Modèles de fondation (transformers)
Apprentissage	Entraîné sur des ensembles de données spécifiques	Entraîné sur de vastes ensembles de données non structurées
Capacités	Classifier, prédire	Générer du contenu, classifier, prédire
Exemples d'Applications	Reconnaissance d'objets, prédictions	ChatGPT, DALL · E 2, Stable Diffusion
Polyvalence	Limitée à des tâches spécifiques	Polyvalente, multiples cas d'utilisation
Limites	Une seule tâche à la fois	Risque de "hallucinations", manque d'explicabilité

Tab. 2.3: Comparaison entre l'IA Classique et Générative

2.2 L'évolution vers les LLMs

2.2.1 Le Traitement du langage naturel (NLP)

1. Définition :

Le NLP pour Natural Language Processing ou Traitement du Langage Naturel est une discipline qui porte essentiellement sur la compréhension, la manipulation et la génération du langage naturel par les machines. Ainsi, le NLP est réellement à l'interface entre la science informatique et la linguistique. Il porte donc sur la capacité de la machine à interagir directement avec l'humain. [21]
Au sein du NLP, différentes sous-parties coexistent. Le Natural Language Processing englobe notamment :
- Le NLU ; Natural Language Understanding ou Compréhension du Langage Naturel en français. Le rôle du NLU est de comprendre en profondeur les échanges et les données, pour identifier les intentions des paroles ou des écrits humains.
- Le NLG ; Natural Language Generation ou Génération du Langage Humain en français. Son rôle est de créer et générer automatiquement des échanges dans une langue définie, grâce à l'intelligence artificielle. Les données se transforment alors en textes, et les entreprises peuvent automatiser certains processus manuels.

2. Domaines d'application du NLP

Aujourd'hui, le traitement du langage naturel automatisé NLP est utilisé dans de nombreux domaines :

a. Sentiment analysis
Aussi connue sous le nom de « Opinion Mining », l'analyse des sentiments consiste à identifier les informations subjectives d'un texte pour extraire l'opinion de l'auteur.

b. Traduction automatique
Le langage naturel étant par nature ambigu et variable, ces applications ne reposent pas sur un travail de remplacement mot à mot, mais nécessitent une véritable analyse et modélisation de texte, connue sous le nom de Traduction automatique statistique (Statistical Machine Translation en anglais).

c. Chatbots
Les méthodes NLP sont au cœur du fonctionnement des Chatbots actuels. Bien que ces systèmes ne soient pas totalement parfaits, ils peuvent aujourd'hui facilement gérer des tâches standards telles que renseigner des clients sur des produits ou services, répondre à leurs questions, etc. Ils sont utilisés par plusieurs canaux, dont l'Internet, les applications et les plateformes de messagerie. L'ouverture de

la plateforme Facebook Messenger aux chatbots en 2016 a contribué à leur développement.

3. Notions de base en NLP

- Token : éléments de base d'une phrase : « Vous », « trouverez », « en », « pièce », « jointe », « le », « document », « en », « question ».
- Sequence : Liste séquentielle de jeton : « Vous », « trouverez », « en pièce jointe », « le document », « en question ».
- Vocabulary : c'est une liste complète des jetons.
- Tokenisation, ou découpage du texte en plusieurs pièces appelées tokens. Exemple : « Vous trouverez en pièce jointe le document en question » ; « Vous », « trouverez », « en pièce jointe », « le document », « en question ».
- Stemming : un même mot peut se retrouver sous différentes formes en fonction du genre (masculin féminin), du nombre (singulier, pluriel), la personne (moi, toi, eux…) etc. Le stemming désigne généralement le processus heuristique brut qui consiste à découper la fin des mots dans afin de ne conserver que la racine du mot. Exemple : « trouverez » -> « trouv »
- Lemmatisation : cela consiste à réaliser la même tâche mais en utilisant un vocabulaire et une analyse fine de la construction des mots. La lemmatisation permet donc de supprimer uniquement les terminaisons inflexibles et donc à isoler la forme canonique du mot, connue sous le nom de lemme. Exemple : « trouvez » -> trouver

2.2.2 Modèle de langage (Language Models)

1. Définition

Un modèle de langue utilise l'apprentissage automatique pour établir une distribution de probabilité sur les mots utilisés afin de prédire le mot suivant le plus probable dans une phrase en fonction de l'entrée précédente. Les modèles de langue apprennent à partir de textes et peuvent être utilisés pour produire du texte original, prédire le prochain mot dans un texte, la reconnaissance vocale, la reconnaissance optique de caractères et la reconnaissance de l'écriture manuscrite. [1]

Un modèle de langue est une distribution de probabilité sur des mots ou des séquences de mots. En pratique, il donne la probabilité qu'une certaine séquence de mots soit "valide" [1]. La validité dans ce contexte ne se réfère pas à la validité grammaticale. Au lieu de cela, elle signifie que cela ressemble à la manière dont les gens écrivent, ce que le modèle de langue apprend. C'est un point important.

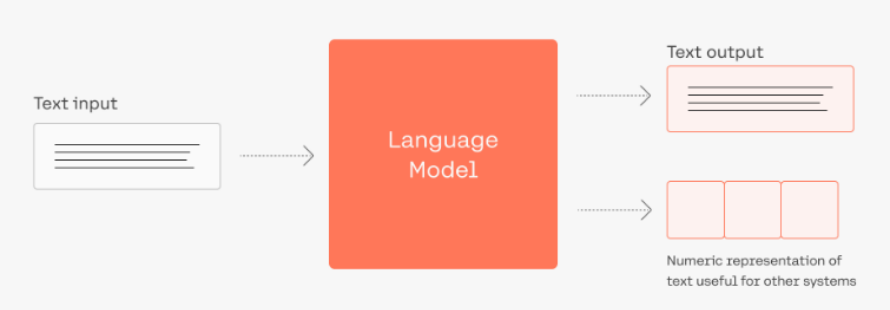


FIG. 2.4: un modèle de langage

Il n'y a pas de magie dans un modèle de langue comme dans d'autres modèles d'apprentissage automatique, en particulier les réseaux neuronaux profonds, c'est juste un outil pour incorporer une abondante information de manière concise et réutilisable dans un contexte hors échantillon.

2. Types des modèles de langage
Il existe deux catégories principales des modèles de langage :

a. Modèle de Langage Probabiliste

Ce sont un type de modèle qui utilisent les motifs statistiques des données pour faire des prédictions sur la probabilité de séquences spécifiques de mots. Une approche de base pour construire un modèle de langue probabiliste consiste à calculer les probabilités des **n-grammes**. [43]
Un n-gramme est une séquence de mots, où n est un nombre supérieur à zéro. Pour créer un modèle de langue probabiliste simple, on calcule la probabilité de différentes combinaisons de mots (n-grammes) dans un texte. Cela se fait en comptant le nombre de fois que chaque combinaison de mots apparaît et en le divisant par le nombre de fois que le mot précédent apparaît.
La formule du modèle n-gramme peut s'écrire comme suit :

$$P(w_n \mid w_1, w_2, w_3, \ldots, w_{n-1}) = P(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_{n-N+1})$$

Où :
- $P(w_n \mid w_1, w_2, w_3, \ldots, w_{n-1})$ est la probabilité du mot w_n étant donné les mots précédents.
- $w_1, w_2, w_3, ..., w_n - 1$ sont les mots précédents.
- $P(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_{n-N+1})$ est la probabilité du mot w_n étant donné les n-1 mots précédents $w_n - 1, w_n - 2, ..., w_n - N + 1$.

Cette idée repose sur un concept appelé l'hypothèse de Markov, qui stipule que la probabilité d'une combinaison de mots (le futur) dépend uniquement du mot précédent (le présent) et non des mots qui l'ont précédé (le passé).

Il existe différents types de modèles de n-grammes tels que :
- Unigrammes qui évaluent chaque mot indépendamment.
- Bigrammes qui considèrent la probabilité d'un mot donné le mot précédent.
- Trigrammes qui considèrent la probabilité d'un mot donné les deux mots précédents ; et ainsi de suite.

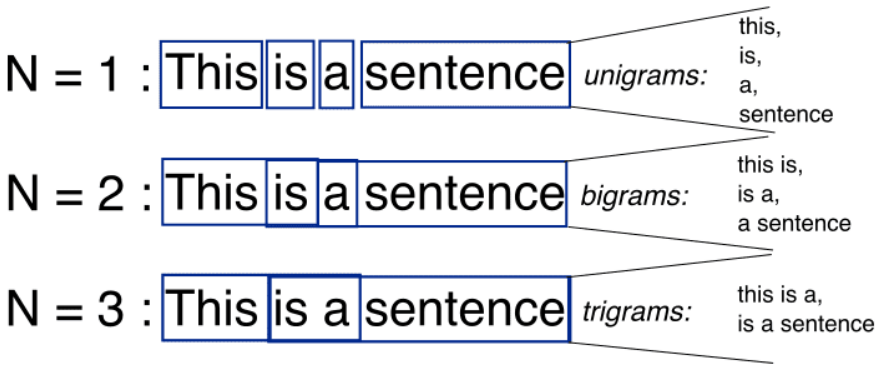


FIG. 2.5: Types des modèles de n-grammes, source [43]

Les n-grammes sont relativement simples et efficaces, mais ils ne prennent pas en compte le contexte à long terme des mots dans une séquence.

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

Les modèles de langue neuronaux, comme leur nom l'indique, utilisent des réseaux neuronaux pour prédire la probabilité d'une séquence de mots. Ces modèles sont entraînés sur un large corpus de données textuelles et sont capables d'apprendre la structure sous-jacente de la langue. Ils peuvent gérer de grands vocabulaires et traiter des mots rares ou inconnus en utilisant des représentations distribuées.[33]

Les architectures de réseaux neuronaux les plus couramment utilisées pour les tâches de traitement du langage naturel (NLP) sont les réseaux neuronaux récurrents (RNN) et les réseaux de transformateurs (nous les aborderons dans la section suivante) Les modèles de langue neuronaux sont capables de mieux capturer le contexte que les modèles statistiques traditionnels. De plus, ils peuvent gérer

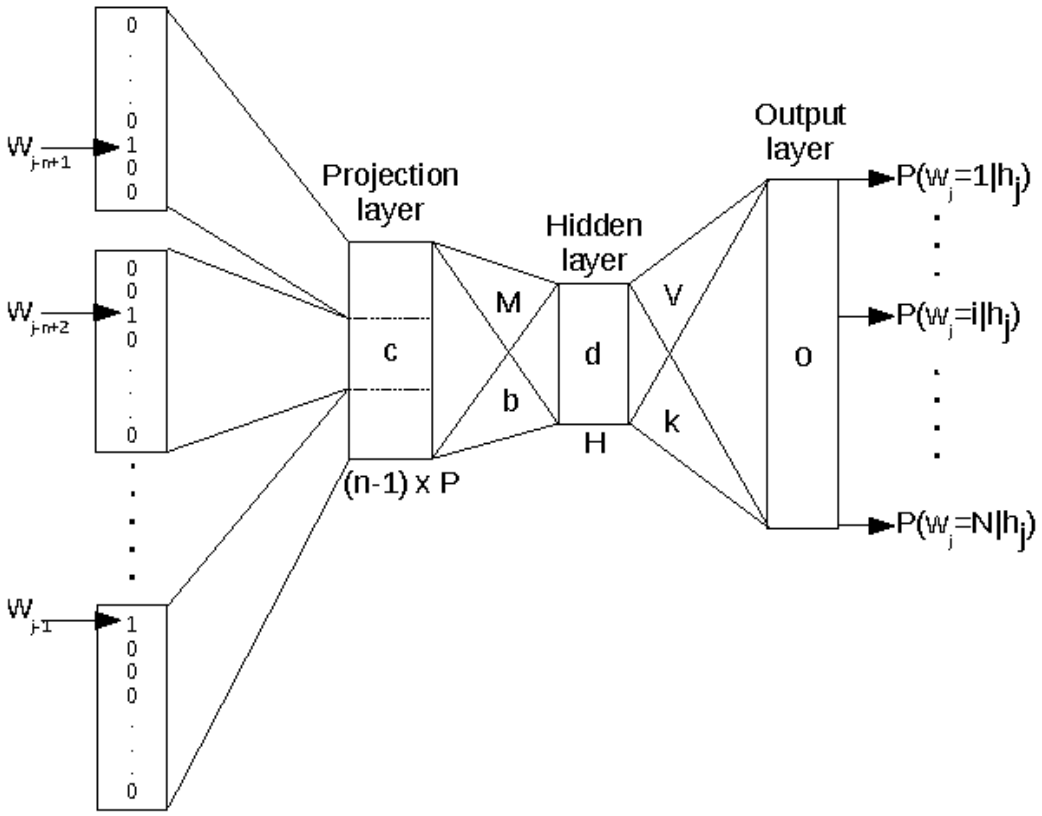


FIG. 2.6: Modèles de langage modernes basés sur les réseaux de neurones, sources : [33]

des structures linguistiques plus complexes et des dépendances plus longues entre les mots.
La formule du modèle de langage basé sur un réseau neuronal peut s'écrire comme suit :

$$P(w_n|w_1, w_2, w_3, ..., w_n - 1) = f(w_n - 1, h_n)$$

Où :
- $P(w_n|w_1, w_2, w_3, ..., w_n - 1)$ est la probabilité du mot w_n étant donné les mots précédents $w_1, w_2, w_3, ..., w_n - 1$
- f est une fonction qui prend la sortie du mot précédent étape $w_n - 1$ et l'état caché du modèle h_n et génère la probabilité du mot suivant w_n.

Les modèles de langage neuronal ont d'abord été basés sur des RNN et des intégrations de mots. Puis le concept de LSTM, GRU et Encoder-Decoder est apparu. L'avancée récente est la découverte de Transformers, qui a radicalement changé le

domaine de la modélisation du langage.

3. Modèles N-grammes vs. Modèles de réseaux neuronaux : Une analyse comparative

Le tableau 2.4 présente une comparaison complète entre les modèles N-grammes et les modèles de réseaux neuronaux dans le contexte du traitement du langage naturel.
Ce tableau 2.4 compare les caractéristiques de chaque type.

Caractéristique	**Modèles N-grammes**	**Modèles de réseaux neuronaux**
Entraînement	Peut être entraîné sur de petits ensembles de données	Nécessite de grands ensembles de données pour un entraînement efficace
Exigence de mémoire	Nécessite moins de mémoire	Nécessite beaucoup plus de mémoire
Vitesse	Plus rapide à entraîner et à prédire	Plus lent à entraîner et à prédire
Compréhension contextuelle	Compréhension contextuelle limitée	Peut capturer des relations contextuelles complexes
Performance	Précision inférieure par rapport aux modèles de réseaux neuronaux	Précision supérieure par rapport aux modèles N-grammes
Application	Convient pour des tâches de modélisation de langage plus simples	Convient pour des tâches de modélisation de langage complexes telles que la traduction automatique et la génération de langage naturel

Tab. 2.4: Comparaison entre les modèles N-grammes et les modèles de réseaux neuronaux

Il met en évidence les différences clés en termes d'exigences de formation, de consommation de mémoire, de vitesse, de compréhension contextuelle, de performances et d'applications. Ces informations sont essentielles pour comprendre les forces et les limites de chaque approche, vous permettant de prendre des décisions

éclairées concernant le meilleur choix de modèle pour vos tâches spécifiques de traitement du langage.
Les transformateurs constituent les éléments de base des nouveaux modèles de langage neuronal. Le concept d'apprentissage par transfert est introduit, ce qui constitue une avancée majeure.

2.2.3 Transformers

1. Définiton

L'architecture Transformer est le bloc de construction fondamental de tous les modèles de langage avec Transformers (LLMs). L'architecture Transformer a été introduite dans l'article "Attention is all you need," [24] publié en décembre 2017 . La version simplifiée de l'architecture Transformer ressemble à ceci :

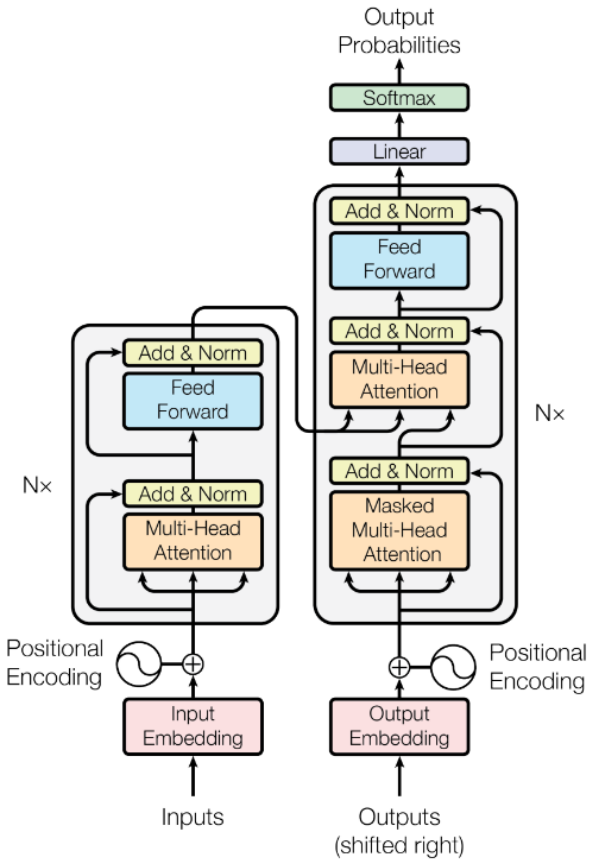


Fig. 2.7: Le transformer - Architecture du modèle, source : [24]

Bien que le diagramme de l'article original soit un peu intimidant, l'innovation derrière les Transformers se résume à trois concepts principaux :
- Encodages Positionnels.
- Le mécanisme d'Attention.
- L'Auto-Attention.

- Encodages Positionnels

Commençons par le premier concept, les encodages positionnels. Supposons que nous essayons de traduire un texte de l'anglais au français. Rappelons-nous que les RNN, l'ancienne méthode de traduction, comprenaient l'ordre des mots en traitant les mots séquentiellement. Mais c'est aussi ce qui les rendait difficiles à paralléliser.

Transformers contournent cet obstacle grâce à une innovation appelée encodages positionnels. L'idée est de prendre tous les mots de votre séquence d'entrée – une phrase en anglais, dans ce cas – et d'ajouter à chaque mot un numéro indiquant son ordre. Ainsi, vous fournissez à votre réseau une séquence comme :

[("Dale", 1), ("says", 2), ("hello", 3), ("world", 4)]

Conceptuellement, on peut considérer cela comme le déplacement de la charge de compréhension de l'ordre des mots de la structure du réseau neuronal vers les données elles-mêmes.

Au début, avant que le Transformer n'ait été entraîné sur des données, il ne sait pas comment interpréter ces encodages positionnels. Mais à mesure que le modèle voit de plus en plus d'exemples de phrases et de leurs encodages, il apprend à les utiliser efficacement.

- Le mécanisme d'Attention

L'attention est un mécanisme qui permet à un modèle de texte de "regarder" chaque mot de la phrase originale lorsqu'il prend une décision sur la façon de traduire les mots dans la phrase de sortie. [24]

Pratiquement Les mécanismes d'attention permettent aux vecteurs (mots) de communiquer entre eux et de transmettre des informations pour mettre à jour leurs valeurs. Par exemple, le sens du mot "modèle" dans la phrase "modèle d'apprentissage automatique" est différent de son sens dans la phrase "modèle de mode". Le cluster d'attention est responsable de déterminer quels mots dans le contexte sont pertinents pour mettre à jour les significations des autres mots, et comment ces significations doivent être mises à jour. Voici une visualisation tirée de cet article

original sur l'attention **(Figure 2.9)** :

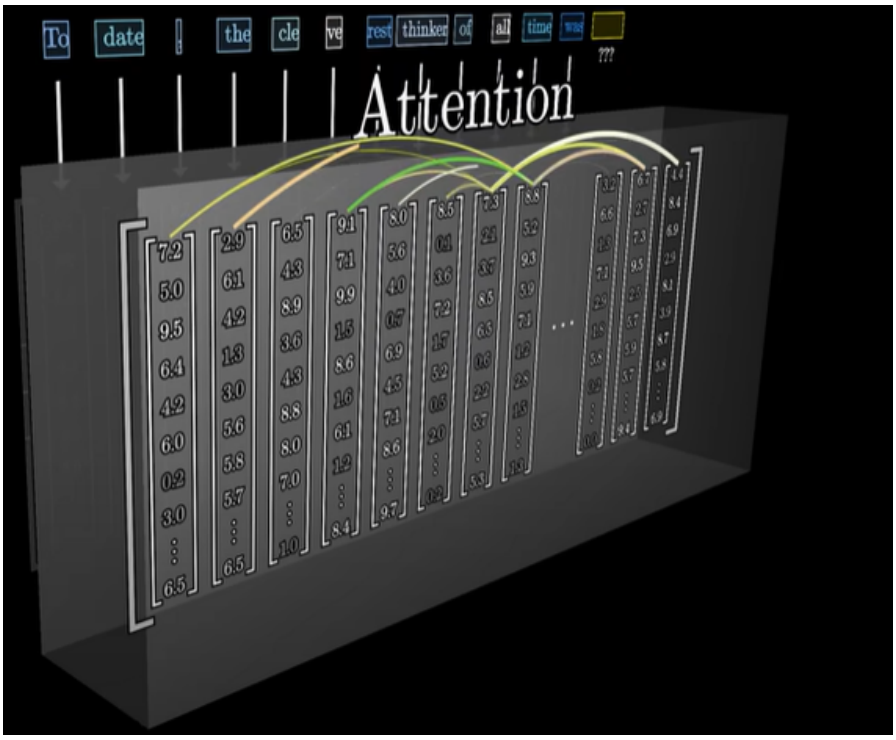


Fig. 2.8: Communication des Vecteurs via les Mécanismes d'Attention, source [29]

Les mécanismes d'attention ont révolutionné l'apprentissage profond en permettant aux modèles de se concentrer dynamiquement sur des parties pertinentes des données d'entrée, semblable à la manière dont les humains prêtent attention à certains aspects d'une scène visuelle ou d'une conversation. Dans les transformers, ces mécanismes pondèrent l'influence des différents tokens d'entrée pour produire une sortie, dépassant parfois les performances humaines.

- Auto-Attention

L'auto-attention permet à un modèle de se concentrer sur différentes positions de sa séquence d'entrée pour calculer une représentation de cette séquence. Elle permet au modèle de pondérer l'importance de chaque mot de la séquence par rapport aux autres, capturant les dépendances entre différents mots de l'entrée [24]. Le mécanisme comporte trois éléments principaux :
- Query : Il s'agit d'un vecteur représentant l'objectif ou la question actuelle que le modèle a sur un mot spécifique de la séquence. C'est comme une lampe de

poche que le modèle braque sur un mot particulier pour comprendre son sens dans le contexte.
- Key : Chaque mot possède une étiquette ou un point de référence — le vecteur clé agit comme cette étiquette. Le modèle compare le vecteur de requête (query) avec tous les vecteurs clés (keys) pour voir quels mots sont les plus pertinents pour répondre à la question sur le mot focalisé.
- Value : Ce vecteur contient les informations réelles associées à chaque mot. Une fois que le modèle a identifié les mots pertinents par le biais des comparaisons de clés, il récupère les vecteurs de valeurs correspondants pour obtenir les détails nécessaires à la compréhension.

Les scores d'attention peuvent être calculés en effectuant un produit scalaire mis à l'échelle entre les vecteurs de requête et les vecteurs clés. Finalement, ces scores sont multipliés par les vecteurs de valeurs pour produire une somme pondérée des valeurs.

2. Que peuvent faire les Transformers
L'architecture Transformer est le bloc de construction fondamental de tous **les grands modèles de langage (LLMs)**. Elle a permis à des modèles comme GPT de générer des sorties plus précises et contextuellement pertinentes. Avec la capacité d'exécuter diverses tâches de traitement du langage naturel, telles que la génération de texte, la synthèse et les réponses aux questions, **les LLMs comme GPT** ouvrent de nouvelles possibilités pour la communication et l'interaction homme-machine.

2.2.4 Large Language Models LLMs

Le langage, crucial pour la communication humaine et l'interaction homme-machine, a motivé le développement de modèles généralisés pour des tâches linguistiques complexes. Les progrès récents, notamment grâce aux Transformers, à l'augmentation de la puissance de calcul et aux vastes corpus d'entraînement, ont permis l'émergence de modèles de langage de grande taille (LLM). Ces LLMs atteignent des performances quasi-humaines sur diverses tâches, démontrant une capacité remarquable de traitement et de génération de texte cohérent, ainsi qu'une généralisation efficace à de multiples applications [1].

Le tableau 2.5 présente l'évolution des modèles de langage au fil du temps.

Pour chaque modèle, le tableau fournit une brève description de son fonctionnement, indique s'il est considéré comme "large" (seuls les Transformers le sont), et donne la période d'émergence. Les Transformers sont présentés comme l'approche

Modèle de Langage	Description	"Large" ?	Émergence
Modèle Bag-of-Words	Représente le texte comme un ensemble de mots désordonnés, sans considérer la séquence ou le contexte	Non	Années 1950–1960
Modèle N-gramme	Considère des groupes de N mots consécutifs pour capturer la séquence	Non	Années 1950–1960
Modèles de Markov Cachés (HMMs)	Représente le langage comme une séquence d'états cachés et de sorties observables	Non	Années 1980–1990
Réseaux Neuronaux Récurrents (RNNs)	Traite les données séquentielles en maintenant un état interne, capturant le contexte des entrées précédentes	Non	Années 1990–2010
Réseaux de Mémoire à Long Court Terme (LSTM)	Extension des RNNs qui capture des dépendances à plus long terme	Non	Années 2010
Transformers	Architecture de réseau neuronal qui traite des séquences de longueur variable en utilisant un mécanisme d'auto-attention	**Oui**	**2017–Présent**

Tab. 2.5: Evolution des modèles de langagage

la plus récente et la seule considérée comme un modèle "large", marquant une avancée significative dans le domaine du traitement du langage naturel.

1. Evolution des LLMs

Depuis les débuts de la traduction automatique pendant la Seconde Guerre mondiale jusqu'à l'avènement de modèles puissants comme GPT-4 et des initiatives open source comme LLaMA, nous avons assisté à une transformation profonde dans le domaine de l'intelligence artificielle et du traitement du langage naturel. La chronologie est un témoignage de l'ingéniosité humaine, de la dédication et de la collaboration.

Nous avons observé la transition des modèles basés sur des règles vers des approches statistiques et, finalement, l'introduction révolutionnaire de l'architecture Transformer, rendant possibles des modèles comme GPT-4. En chemin, des modèles comme BERT et Seq2Seq ont marqué leur empreinte, redéfinissant notre

compréhension du langage.

La figure 2.9 montre une chronologie du développement des LLMs ayant plus de 10 milliards de paramètres, mettant en évidence des avancées significatives et des sorties au cours des dernières années. Cette chronologie représente visuellement les progrès dans le domaine, montrant la croissance rapide et l'évolution des LLMs, marquée par des modèles clés et des étapes importantes qui ont repoussé les limites de ce que ces modèles peuvent accomplir.

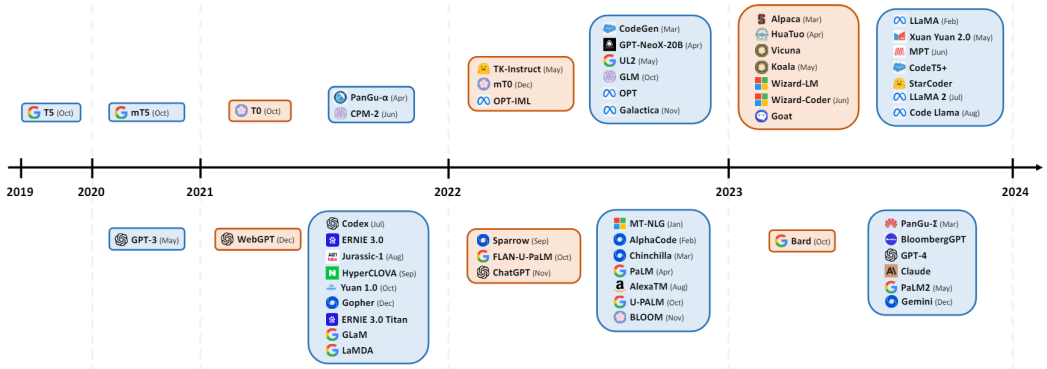


FIG. 2.9: Affichage chronologique des versions des LLMs : les cartes bleues représentent les modèles pré-entraînés, tandis que les cartes oranges correspondent aux modèles ajustés par instruction. Les modèles situés dans la moitié supérieure indiquent une disponibilité en open source, tandis que ceux dans la moitié inférieure sont en source fermée, source [1]

2. Architecture des LLMs

2.1. Décodeur

La fonction principale du décodeur est de générer des séquences de sortie (comme des phrases) à partir d'une représentation intermédiaire des données d'entrée. Cette architecture fait souvent partie d'un cadre plus large encodeur-décodeur, mais peut également être utilisée indépendamment dans certains modèles, que nous allons voir juste après. La figure 2.10 illustre l'architecture d'un décodeur.

a. Décodeur Causal

Un type d'architecture qui n'a pas d'encodeur et qui traite et génère la sortie en utilisant un décodeur, où le token prédit dépend uniquement des étapes de

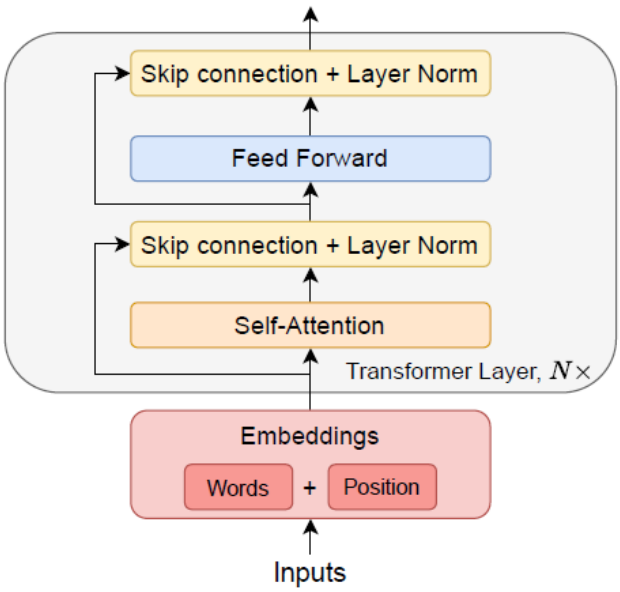


FIG. 2.10: Architecture d'un décodeur

temps précédentes. L'attention n'est pas bidirectionnelle mais unidirectionnelle, elle s'oriente soit de gauche à droite, soit de droite à gauche [1]. Il s'agit de l'architecture utilisée pour les GPTs de OpenAI. Un exemple de masque d'attention causal est montré à la Figure 2.11

$$\begin{cases} s_{ij} = q_i^T k_j \in \mathbb{R}, & 1 \leq j \leq i, \\ \alpha_i = \text{Softmax}(s_i) \in \mathbb{R}^i, \\ y_i = \sum_{j=1}^{i} \alpha_{ij} v_j \in \mathbb{R}^d. \end{cases} \tag{2.1}$$

b. Décodeur Préfixe

Aussi connu sous le nom de décodeur non causal, où le calcul de l'attention ne dépend pas strictement des informations passées et l'attention est bidirectionnelle [1]. Un exemple de masque d'attention non causal est montré à la Figure 2.12.

$$\begin{cases} s_{ij} = q_i^T k_j \in \mathbb{R}, & 1 \leq j \leq L, \\ \alpha_i = \text{Softmax}(s_i) \in \mathbb{R}^L, \\ y_i = \sum_{j=1}^{L} \alpha_{ij} v_j \in \mathbb{R}^d. \end{cases} \tag{2.2}$$

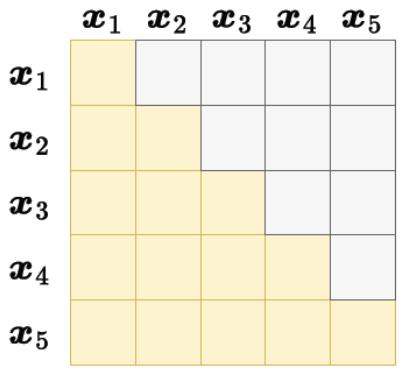


FIG. 2.11: Attention unidirectionnelle, les tokens ne peuvent se concentrer que vers l'arrière.

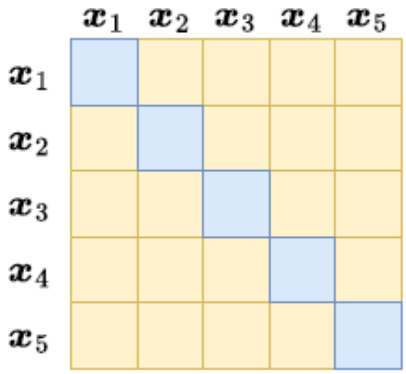


FIG. 2.12: Attention bidirectionnelle, les tokens se concentrent sur chaque token.

2.2. Encodeur-Décodeur

Cette architecture traite les entrées via l'encodeur et transmet la représentation intermédiaire au décodeur pour générer la sortie. Ici, l'encodeur voit la séquence complète en utilisant l'attention automatique, tandis que le décodeur traite la séquence une par une en mettant en œuvre l'attention croisée [1]. Il s'agit de la combinaison d'un encodeur bidirectionnel comme BERT et d'un decodeur causal. Au lieu de calculer la similarité d'un token x_i avec les autres tokens $(x_j)_{j\neq i}$, nous calculons la similarité avec la sortie de l'encodeur.La figure 2.13 illustre plus en détail cette architecture.

$$\begin{cases} q_i = Qx_i, \\ v_j = Vh_j^{\text{enc}}, \\ k_j = Kh_j^{\text{enc}}, \end{cases}$$

$$\begin{cases} s_{ij} = q_i^T k_j \in \mathbb{R}, & 1 \leq j \leq L, \\ \alpha_i = \text{Softmax}(s_i) \in \mathbb{R}^L, \\ y_i = \sum_{j=1}^{L} \alpha_{ij} v_j \in \mathbb{R}^d. \end{cases}$$

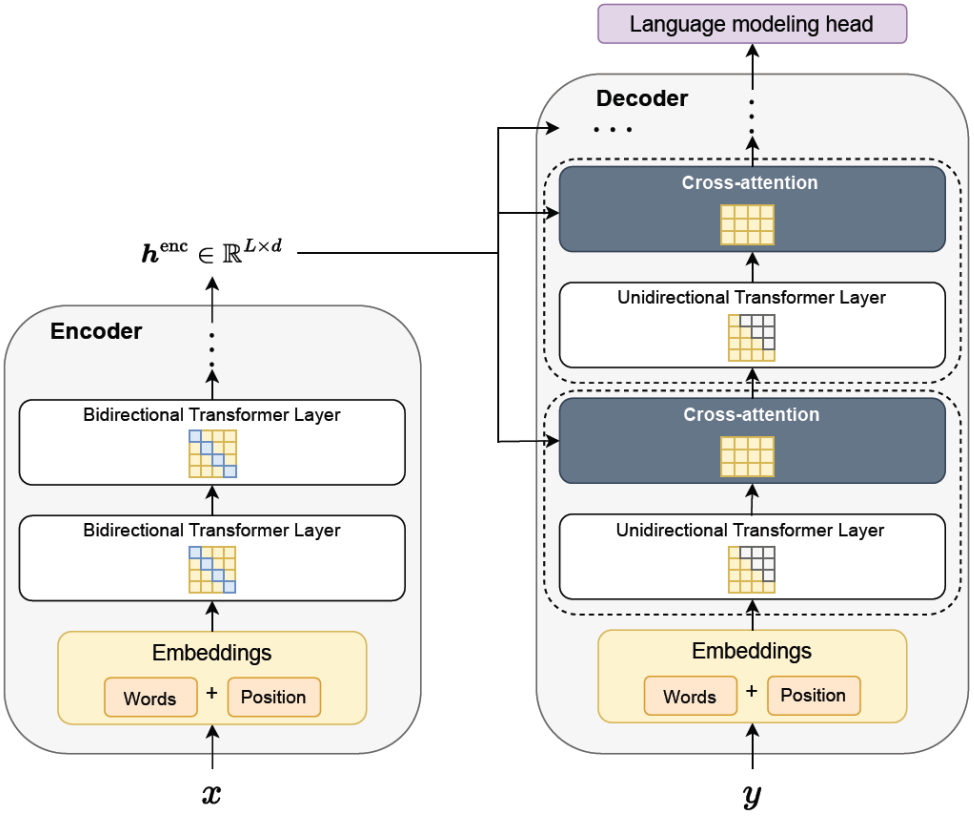


FIG. 2.13: Architecture encodeur-décodeur avec couches d'attention croisée

2.3. Mixture-of-Experts

C'est une variante de l'architecture des transformers avec des experts indépendants parallèles et un routeur pour diriger les tokens vers les experts. Ces experts sont des couches feed-forward après le bloc d'attention. Mixture-of-Experts (MoE) est une architecture sparse efficace qui offre des performances comparables aux modèles denses et permet d'augmenter la taille du modèle sans augmenter le coût computationnel en activant seulement quelques experts à la fois [1].

2.4. Paramètres clés dans un LLM

Les grands modèles de langage (LLM) ont plusieurs paramètres clés qui définissent leur architecture et leur comportement [27].

- **Taille du Modèle :** La taille du modèle fait référence au nombre de paramètres dans le LLM. Un paramètre est une variable apprise par le LLM lors de l'entraînement. La taille du modèle est généralement mesurée en milliards ou en trillions de paramètres. Une plus grande taille de modèle se traduit généralement par de meilleures performances, mais nécessite également plus de ressources informatiques pour s'entraîner et fonctionner.

  La taille du modèle peut être représentée mathématiquement comme suit :

  $$\text{Taille du Modèle} = \text{Nombre de Paramètres}$$

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

- **Taille du Vocabulaire :** La taille du vocabulaire fait référence au nombre de tokens uniques (mots, ponctuation, etc.) sur lesquels le LLM est entraîné. Un vocabulaire plus large permet au LLM de comprendre et de générer une gamme plus large de langage, mais cela nécessite également plus de ressources informatiques.

  La taille du vocabulaire peut être représentée comme suit :

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

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

- **Température :** La température est un paramètre qui contrôle l'aléa de la sortie du LLM. Une température plus élevée produira un texte plus créatif et imaginatif, tandis qu'une température plus basse produira un texte plus précis et factuel [28].

La température peut être représentée mathématiquement comme suit :

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

où 'Logit' est la sortie brute du LLM avant l'opération softmax, et 'Température' est le paramètre qui échelle les logits.

Par exemple, si vous définissez la température à 1,0, le LLM générera toujours le mot suivant le plus probable. Cependant, si vous définissez la température à 2,0, le LLM sera plus susceptible de générer des mots moins probables, ce qui pourrait produire un texte plus créatif.

- **Fenêtre de Contexte :** La fenêtre de contexte est le nombre de mots que le LLM considère lors de la génération de texte. Une fenêtre de contexte plus grande permettra au LLM de générer un texte plus contextuellement pertinent, mais rendra également le processus d'entraînement plus coûteux en termes de calcul.

  La fenêtre de contexte peut être représentée comme suit :

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

  Par exemple, si la fenêtre de contexte est fixée à 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 filtrent la sélection des tokens pendant la génération de texte. Top-k sélectionne les k tokens les plus probables, garantissant une sortie de haute qualité. Top-p, quant à lui, définit un seuil de probabilité cumulée, conservant les tokens dont la probabilité totale est supérieure à ce seuil. Top-k est utile pour éviter les réponses absurdes, tandis que Top-p peut assurer la diversité.

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

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

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

  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 Top-p à 0,9, le LLM ne générera que des mots ayant une probabilité d'au moins 0,9.

Ces paramètres, ainsi que d'autres comme les séquences d'arrêt, permettent d'ajuster le comportement du LLM pour répondre à des exigences et des cas d'utilisation spécifiques [27].

3. Domaine d'application

L'adoption généralisée des modèles de langage de grande envergure (Large Language Models, LLMs) pour diverses applications en aval constitue une tendance prédominante tant dans les milieux de recherche en intelligence artificielle que dans les secteurs industriels. Bien que chacun de ces domaines présente des défis spécifiques, la généralisation des LLMs ouvre la voie à des contributions potentiellement substantielles [1].

- **Utilisation générale :**Les LLMs sont reconnus comme des outils polyvalents grâce à leur capacité à comprendre, générer et manipuler le langage humain de manière contextuelle. Leurs applications s'étendent de tâches simples comme la traduction à des opérations complexes telles que la synthèse et la génération de texte. Ils jouent également un rôle crucial dans l'analyse de données textuelles à grande échelle. Cependant, leur efficacité dépend de la qualité des données d'entraînement [1].
- **La finance :** Les grands modèles de langage (LLMs) spécialisés en finance, comme BloombergGPT et FinGPT, montrent une efficacité supérieure grâce à leur entraînement sur des données financières spécifiques. BloombergGPT illustre l'avantage des données propriétaires, tandis que FinGPT, en tant que modèle open-source, favorise l'innovation accessible. [1].

4. Evaluation des LLMs

4.1. Evaluation traditionnelle :

- **Perplexité**

  Perplexité est une métrique couramment utilisée pour évaluer les performances des modèles de langage. Elle quantifie à quel point le modèle prédit correctement un échantillon de texte. Des valeurs de perplexité plus faibles indiquent de meilleures performances [30][31]. La perplexité est définie comme :

$$\text{Perplexité} = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 p(x_i)}$$

  où N est le nombre de jetons (tokens) dans le texte, et $p(x_i)$ est la probabilité attribuée par le modèle au i-ème jeton.

- **BLEU (Bilingual Evaluation Understudy)**

  BLEU est une métrique couramment utilisée dans les tâches de traduction automatique. Elle compare la sortie générée avec une ou plusieurs traductions de référence et mesure la similarité entre elles. Les scores BLEU vont de 0 à 1, avec des scores plus élevés indiquant de meilleures performances [30][31].

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)**

  ROUGE est un ensemble de métriques utilisées pour évaluer la qualité des résumés. Il compare le résumé généré avec un ou plusieurs résumés de référence et calcule la précision, le rappel et le score F1. Les scores ROUGE fournissent des insights sur les capacités de génération de résumés du modèle de langage[30][31].

4.2. Évaluation Humaine

L'évaluation humaine implique de faire appel à des évaluateurs humains qui évaluent la qualité des sorties du modèle de langage. Ces évaluateurs notent les réponses générées selon différents critères, tels que la pertinence, la fluidité, la cohérence et la qualité globale. Cette approche offre des retours subjectifs sur les performances du modèle [32].

4.3. Évaluation GPT (G-Eval)

G-Eval est un cadre récemment développé utilisant GPT-4 pour évaluer les sorties des LLM. Il génère une série d'étapes d'évaluation en utilisant des chaînes de pensées (CoTs), puis utilise ces étapes générées pour déterminer le score final via un paradigme de remplissage de formulaire [32] [34].

L'algorithme G-Eval fonctionne comme suit :

1. Présenter une tâche d'évaluation à GPT-4 (par exemple, évaluer cette sortie de 1 à 5 basée sur la cohérence).

2. Fournir une définition des critères d'évaluation (par exemple, "Cohérence - la qualité collective de toutes les phrases dans la sortie réelle").

3. GPT-4 génère des étapes d'évaluation en fonction de la tâche et des critères donnés.

4. Utiliser GPT-4 pour produire un score de 1 à 5 basé sur les étapes d'évaluation générées.

G-Eval est considéré comme l'une des meilleures méthodes pour créer des métriques spécifiques à la tâche pour évaluer les sorties des LLM, car elle est bien alignée avec les attentes humaines et fournit des scores fiables et précis [32].

En résumé, les méthodes traditionnelles d'évaluation comme la perplexité, BLEU et ROUGE fournissent des mesures quantitatives des performances des modèles de langage. L'évaluation humaine offre des retours subjectifs, tandis que G-Eval utilise GPT-4 pour générer des métriques spécifiques à la tâche qui sont en accord avec les attentes humaines.

5. Inconvénients et challenges

Les modèles de langage de grande envergure (LLMs) tels que GPT-4 ont considérablement fait progresser le traitement du langage naturel. Cependant, leur développement et leur utilisation s'accompagnent de nombreux défis[1].

Sur le plan technique, ces modèles font face à des problèmes de coût computationnel, de robustesse face aux attaques adversariales et d'interprétabilité. Leur extension à des tâches plus complexes ou à des environnements dynamiques soulève de nouvelles questions en termes de scalabilité, de confidentialité et de traitement en temps réel. La recherche explore également l'intégration de la multi-modalité et l'efficacité de l'apprentissage par transfert[1].

- **Coût Computationnel :** L'entraînement des LLMs requiert des ressources computationnelles considérables, engendrant des coûts de production élevés et des préoccupations environnementales liées à la consommation énergétique massive.

- **Biais et Équité :** Les LLMs risquent d'hériter et d'amplifier les biais sociétaux présents dans leurs données d'entraînement. Ces biais peuvent se manifester dans les sorties du modèle, soulevant des questions éthiques et d'équité significatives.

- **Surapprentissage :** Malgré leurs capacités d'apprentissage importantes, les LLMs sont susceptibles de surapprendre des motifs spécifiques et bruyants de leurs vastes ensembles de données, conduisant potentiellement à des réponses illogiques. Le défi réside dans l'équilibre entre mémorisation et généralisation : la mémorisation permet des réponses précises à des questions spécifiques, tandis que la généralisation est essentielle pour traiter des tâches diverses et nouvelles.

- **Inégalité Économique et de Recherche :** Les coûts élevés associés au développement et au déploiement des LLMs risquent de concentrer leur développement au sein d'organisations bien dotées financièrement, exacerbant potentiellement les inégalités économiques et de recherche dans le domaine de l'IA.

- **Hallucinations :** Les LLMs peuvent produire des "hallucinations", générant des réponses qui semblent plausibles mais sont en réalité incorrectes ou ne correspondent pas aux informations fournies. Ces hallucinations se déclinent en trois catégories : contradictions à l'entrée, au contexte, ou aux faits établis.

- **Ingénierie des Prompts :** La formulation des prompts joue un rôle crucial dans la détermination des sorties du modèle. Des variations subtiles dans les requêtes peuvent entraîner des changements significatifs dans les réponses générées. L'ingénierie des prompts vise à concevoir des requêtes efficaces pour guider les réponses des LLMs de manière optimale.

- **Connaissances Limitées :** Les informations acquises lors du pré-entraînement sont limitées et peuvent devenir obsolètes. La mise à jour des connaissances du modèle est coûteuse, nécessitant soit un réentraînement complet, soit l'utilisation de techniques comme l'augmentation de la récupération générative (RAG).

- **Oubli Catastrophique :** Le processus d'entraînement des LLMs comprend généralement une phase de pré-entraînement sur des ensembles de données volumineux, suivie d'une phase d'affinement sur des données spécifiques à un domaine particulier.

- **Interprétabilité et Explicabilité :** Les LLMs fonctionnent comme des "boîtes noires", ce qui complique la compréhension de leur processus décisionnel. Cette opacité entrave leur acceptation et leur fiabilité, particulièrement dans les domaines sensibles.

2.3 Exploitation et amélioration des LLMs

2.3.1 Les assistants IA

1. Définition
Un assistant IA est un programme logiciel qui utilise l'intelligence artificielle pour effectuer des tâches, fournir des informations et interagir avec les utilisateurs de manière conversationnelle. Les assistants IA sont conçus pour comprendre les entrées en langage naturel des utilisateurs et y répondre de manière appropriée,

souvent en utilisant l'apprentissage automatique pour devenir plus efficaces avec le temps.[25]

Les assistants IA pour les entreprises peuvent accomplir un large éventail de fonctions, notamment : Fourniture d'informations, Automatisation des tâches, Assistance personnalisée Interaction vocale, Intégration avec d'autres services.

Des exemples populaires d'assistants IA incluent Siri d'Apple, Google Assistant, Amazon Alexa, Microsoft Cortana et Samsung Bixby. Ces assistants sont intégrés à divers appareils, tels que des smartphones, des enceintes intelligentes et des appareils domotiques intelligents, offrant aux utilisateurs un accès pratique aux informations et aux services.

2. Composants architecturaux des assistants IA
L'architecture de conception d'un assistant virtuel IA vise à créer un système robuste et convivial capable de comprendre et de répondre efficacement aux entrées humaines, de réaliser une variété de tâches et de s'améliorer continuellement tout en maintenant des normes élevées de sécurité et de confidentialité.

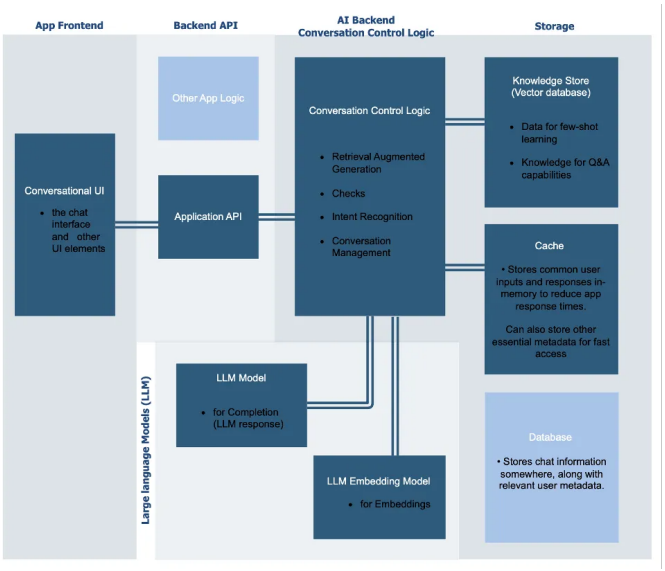


FIG. 2.14: Architecture de haut niveau pour un assistant virtuel IA typique basé sur un LLM, source : [25]

Bien que diverses architectures de conception existent pour la construction d'assistants virtuels IA, la structure illustrée dans la Figure 1 sert de modèle de référence, répondant aux exigences de conception de base d'un assistant virtuel IA standard et cette dernière se compose de :

a. Interface utilisateur conversationnelle
Les composants architecturaux des assistants IA comprennent une interface utilisateur conversationnelle, une rétroaction en temps opportun, la gestion des erreurs et la priorité à la sécurité de l'utilisateur.

b. LLMs
Les grands modèles de langue (LLM) forment la base des systèmes d'assistant IA. Ils utilisent des techniques avancées de traitement du langage naturel pour comprendre les entrées des utilisateurs et générer des réponses humaines. Le LLM agit comme un traducteur entre l'utilisateur et le système, convertissant les requêtes en actions et en réponses compréhensibles.

c. Base de connaissances
La base de connaissances constitue la base pour fournir des interactions précises, pertinentes et personnalisées entre les utilisateurs et l'assistant IA, cette dernière stocke les données utilisateur, les exemples dynamiques et les informations factuelles sous forme de vecteurs numériques. Des modèles d'incrustation sont utilisés pour créer ces vecteurs, qui codent des informations sémantiques sur les données.

d. Backend application API
L'API de l'application backend est un lien crucial entre l'assistant IA et les applications externes, permettant une communication fluide et une interaction efficace. Elle permet à l'assistant IA d'exécuter diverses tâches telles que la récupération d'informations depuis des bases de données, l'accès à des services web, le contrôle d'appareils intelligents et l'exécution de la logique métier.

2.3.2 Techniques pour adapter les LLMs à des tâches spécifiques.

Depuis la sortie des grands modèles de langage (LLMs) et des modèles de chat avancés, diverses techniques ont été utilisées pour extraire les résultats souhaités de ces systèmes d'IA. Certaines de ces méthodes impliquent de modifier le comportement du modèle pour mieux l'aligner avec nos attentes, tandis que d'autres se concentrent sur l'amélioration de notre manière de formuler les requêtes aux

LLMs afin d'extraire des informations plus précises et pertinentes.Des techniques comme le Retrieval Augmented Generation (RAG), le Prompting et le fine-tuning sont les plus largement utilisées.

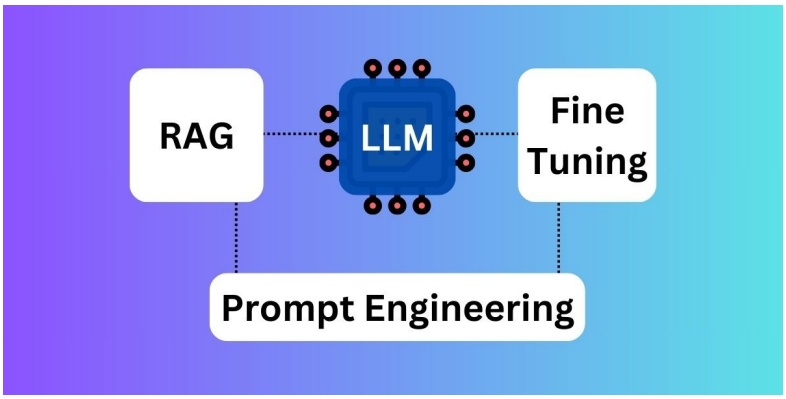


Fig. 2.15: Fine-Tuning vs. Prompting vs. RAG, source : [57]

Toutes les trois techniques (prompt engineering, fine-tuning, et RAG) sont des méthodes pour adapter les grands modèles de langage (LLMs) à des tâches spécifiques. Elles entraînent essentiellement le LLM à mieux performer dans un domaine particulier.
Prompting : C'est comme donner une instruction claire à votre LLM, facile à utiliser mais avec un contrôle limité sur le résultat. Idéal pour les tâches simples !
Fine-Tuning : Imaginez entraîner votre LLM comme un spécialiste sur un sujet particulier. Très précis mais nécessite beaucoup de données et d'efforts.
RAG (Retrieval-Augmented Generation) : Pensez à donner à votre LLM accès à des documents de référence en plus du prompt. Offre des réponses plus riches avec moins de données nécessaires que le fine-tuning.
Le tableau 2.6 présente une comparaison entre les 3 techniques :

La meilleure technique dépend des besoins du projet, mais généralement pour :
- Des tâches simples et ressources limitées ; Le prompt engineering est le meilleur choix.
- Pour de haute précision et personnalisation essentielles ; Le fine-tuning peut être nécessaire.
- Tandis que quand on a besoin d'un équilibre entre performance et efficacité ; Le RAG offre un bon compromis.

Technique	Description	Forces	Limitations
RAG (Génération Augmentée par la Récupération)	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 pré-entraîné sur une tâche ou un ensemble de données spécifique	S'adapte à des domaines/tâches spécialisés, exploite les connaissances pré-entraînées, efficace sur le plan informatique	Potentiel d'oubli catastrophique, limité par la distribution des données de pré-entraînement
Prompt Engineering	Conception minutieuse des invites pour guider le comportement du modèle linguistique	Oriente le modèle sans réentraînement, flexible, peut être combiné avec d'autres techniques	Prend du temps, nécessite des essais et des erreurs, peut ne pas bien se généraliser

Tab. 2.6: Comparaison des techniques d'amélioration des modèles linguistiques

2.3.3 Le RAG - Retrieval Augmented Generation

1. Définition

Retrieval Augmented Generation (RAG) est une technique d'apprentissage automatique qui combine la puissance des méthodes basées sur **la récupération d'information** (retrieval-based methods) avec **les modèles génératifs**.
Elle est particulièrement utilisée en traitement automatique du langage naturel (NLP) pour améliorer les capacités des grands modèles de langage (LLM). RAG fonctionne en récupérant des documents ou des extraits de données pertinents en réponse à des requêtes, qui sont ensuite utilisés pour générer des résultats plus précis et contextuellement pertinents.

2. Le processus du RAG

Comme son nom l'indique, RAG se compose de deux parties : la récupération et la génération. Mais cela ne clarifie pas grand-chose. Il est plus utile de considérer RAG comme un processus en quatre étapes.

Étape 1 : Collecte de données

Dans la première étape de la Génération Augmentée par Récupération (RAG), l'accent est mis sur l'acquisition d'informations pertinentes à partir de sources diversifiées. Cela implique l'identification de répertoires appropriés tels que des bases de données, des articles et des sites web contenant des données pertinentes liées à la tâche ou au sujet spécifique. Des requêtes sont méticuleusement élaborées pour récupérer des informations ciblées, assurant ainsi que les données récupérées sont alignées sur le contexte et les exigences du modèle d'IA.

Étape 2 : Fragmentation des données (Chunking)

Le chunking consiste à diviser un texte long ou un document en segments plus petits et plus gérables, appelés "chunks". Chaque chunk représente une portion cohérente du texte, qui peut être traitée indépendamment.

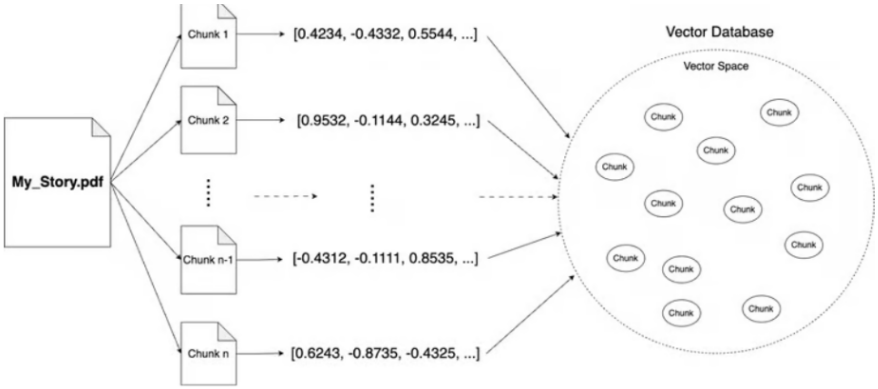


Fig. 2.16: Le Chunking, source : [56]

De cette manière, chaque fragment de données est axé sur un sujet spécifique. Lorsqu'une information est extraite du jeu de données source, elle est plus susceptible d'être directement applicable à la requête de l'utilisateur, car nous évitons

d'inclure des informations non pertinentes provenant de l'ensemble des documents.

Étape 3 : Vectorisation des documents - Embeddings

Les embeddings sont des représentations vectorielles des mots ou des phrases dans un espace de haute dimension. Chaque mot ou phrase est converti en un vecteur de nombres réels, capturant ainsi les relations sémantiques et syntaxiques entre eux comme le montre la figure 2.17 :

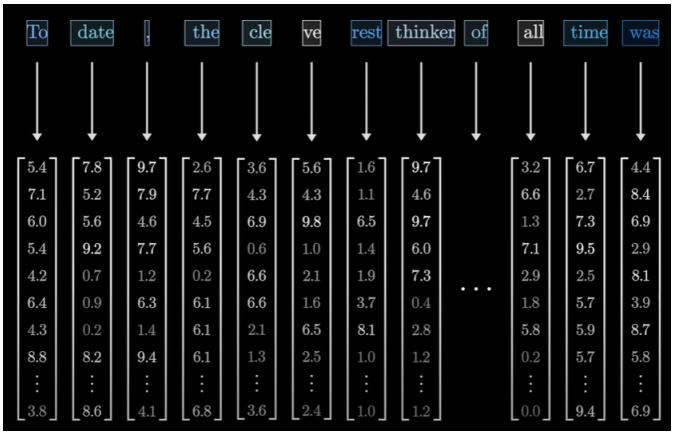


FIG. 2.17: Vectorisation des mots, source : [**?**]

Chacun de ces tokens est ensuite associé à un vecteur, c'est-à-dire une liste de nombres, qui est destiné à encoder le sens de ce token d'une certaine manière. Si l'on considère ces vecteurs comme donnant des coordonnées dans un espace de très haute dimension, alors les mots ayant des significations similaires ont tendance à atterrir sur des vecteurs proches les uns des autres dans cet espace.
Cela nous invite à penser à ces vecteurs de manière très géométrique comme des points dans un espace de haute dimension comme le montre la figure 2.18

Visualiser une liste de trois nombres comme des coordonnées pour des points dans l'espace 3D ne poserait pas de problème, mais les embeddings de mots tendent à être beaucoup plus dimensionnels. Par exemple, Dans GPT-3, ils ont 12 288 dimension.

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

Une base de données vectorielle est un type spécifique de base de données

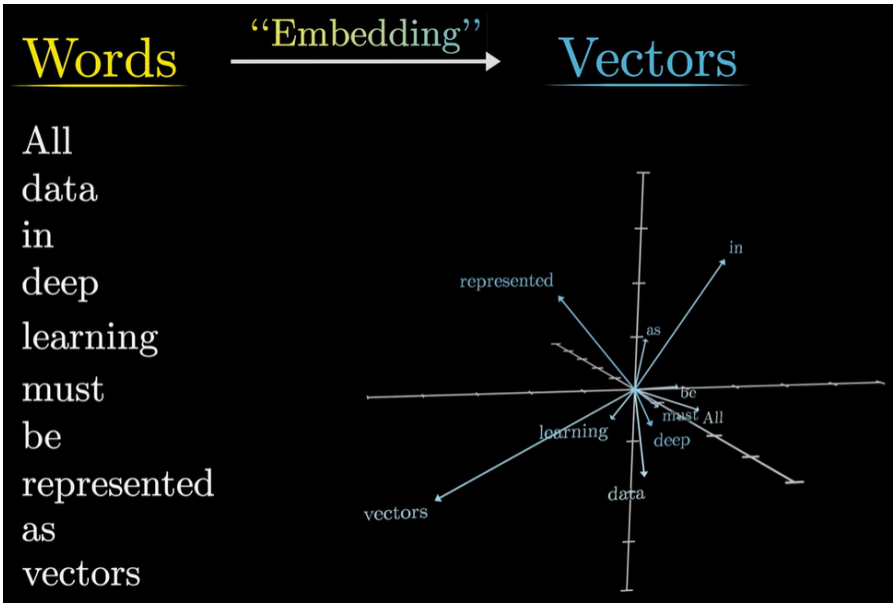


FIG. 2.18: représentation spatiales des mots dans la base de données vectorielles, source : [28]

qui enregistre des informations sous forme de vecteurs multidimensionnels représentant certaines caractéristiques ou qualités. Le nombre de dimensions de chaque vecteur peut varier considérablement, allant de quelques-unes à plusieurs milliers, en fonction de la complexité et du niveau de détail des données. [56]
Ces données, qui peuvent inclure des textes, des images, des fichiers audio et vidéo, sont transformées en vecteurs à l'aide de divers procédés tels que des modèles d'apprentissage automatique, des embeddings de mots ou des techniques d'extraction de caractéristiques.

Nous connaissons tous plus ou moins le fonctionnement des bases de données traditionnelles : elles stockent des chaînes de caractères, des nombres et d'autres types de données scalaires dans des lignes et des colonnes. En revanche, une base de données vectorielle fonctionne sur des vecteurs, ce qui modifie considérablement son optimisation et ses méthodes d'interrogation.

Étape 5 : Recherche de similarité requêtes utilisateur-base de données

La recherche de similarité vectorielle est un processus qui consiste à comparer la similarité entre des vecteurs en utilisant diverses **métriques de distance** qui représentent la "proximité" entre les vecteurs, et aussi des **algorithmes de recherche de similarité vectorielle.**

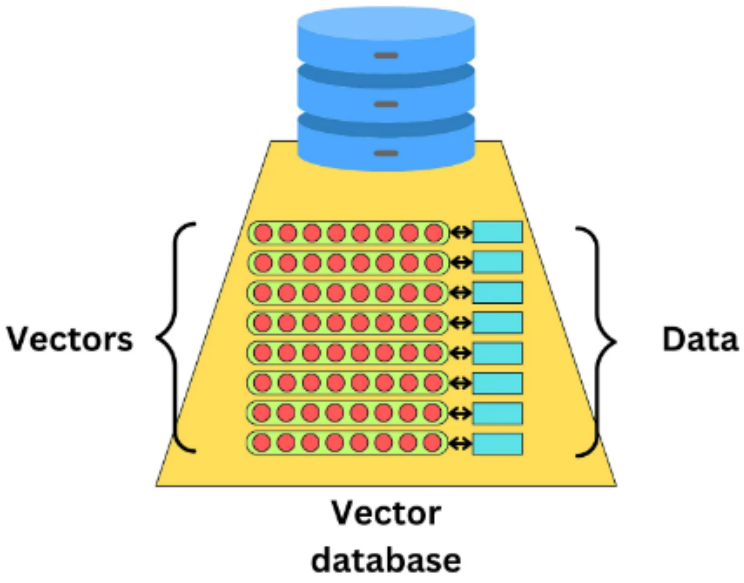


FIG. 2.19: Base de données vectorielles, source : [56]

- Métriques de similarité vectorielle

Les vecteurs peuvent être représentés sous forme de listes de nombres ou par une orientation et une magnitude. Pour mieux comprendre cela, on peut imaginer les vecteurs comme des segments de ligne pointant dans des directions spécifiques dans l'espace. [55]
La métrique L2 ou euclidienne est la métrique "hypoténuse" de deux vecteurs. Elle mesure la magnitude de la distance entre les extrémités des lignes de vos vecteurs.
La similarité cosinus est l'angle entre vos lignes là où elles se rencontrent.
Le produit intérieur est la "projection" d'un vecteur sur l'autre. Intuitivement, il mesure à la fois la distance et l'angle entre les vecteurs.

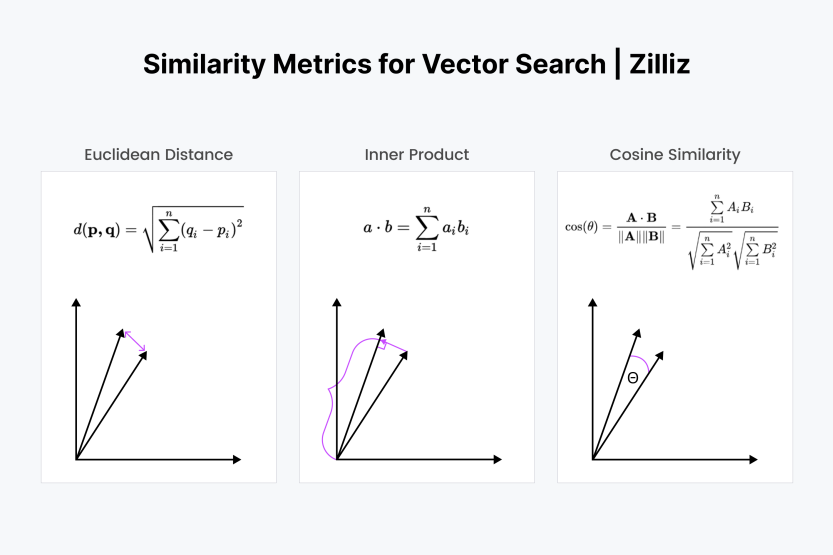


FIG. 2.20: Métriques de similarité vectorielle, source : [55]

- Algorithmes de recherche de similarité vectorielle

Les algorithmes utilisés dans la recherche vectorielle comprennent la recherche exhaustive des k plus proches voisins KNN ; K Nearest Neighbors, ou bien une recherche approximative du voisin le plus proche ou bien les ANN ; Approximate Nearest Neighbors.

a. Le KNN : le k plus proches voisins exhaustif effectue une recherche brute qui scanne l'ensemble de l'espace vectoriel. Comme il nécessite beaucoup de calculs, utilisez le KNN exhaustif pour les ensembles de données de petite à moyenne taille, ou lorsque les exigences de précision l'emportent sur les considérations de performances des requêtes.

b. Le ANN La recherche de voisins les plus proches approximative est une technique qui permet une recherche efficace de similarité sémantique dans de grands ensembles de données souvent trouvés dans des bases de données vectorielles comme, elle vise à trouver un voisin le plus proche approximatif avec une forte probabilité tout en minimisant le coût computationnel.

On peut diviser les algorithmes ANN en trois catégories distinctes : les arbres, les hachages et les graphes. Les deux algorithmes ANN les plus connus sont :

1. HSNW Hierarchical Navigable Small worlds

HNSW se classe dans la catégorie des graphes Plus précisément, il s'agit d'un graphe de proximité , dans lequel deux sommets sont liés en fonction de leur proximité (les sommets les plus proches sont liés) — souvent définie en distance euclidienne.
HNSW est une évolution naturelle de la recherche de vecteurs à l'aide de graphiques, Navigable Small World (NSW), qui s'inspire des multicouches hiérarchiques de la structure de liste de sauts de probabilité.

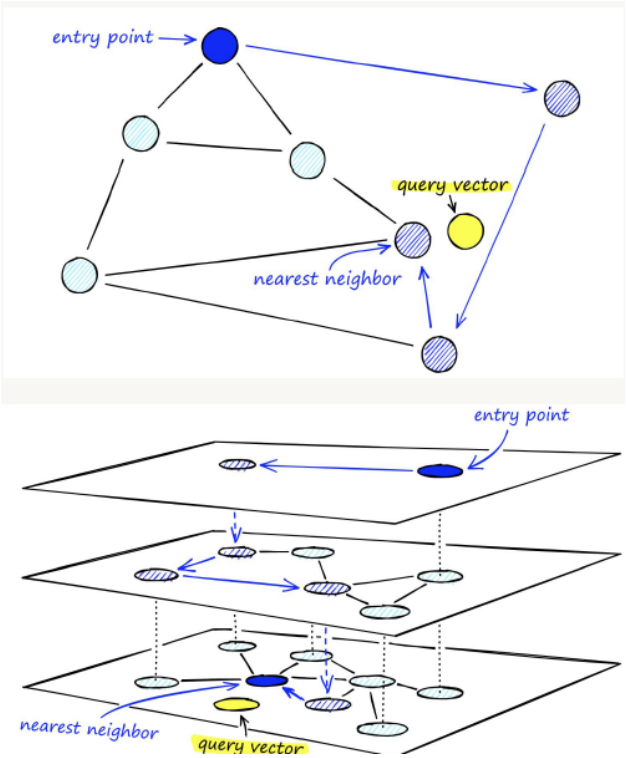


FIG. 2.21: Graphiques navigables du petit monde NSW, source : [54]

2. FAISS Facebook AI similarity Search

Faiss se classe dans la catégorie des hachages, c'est est une bibliothèque — développée par Facebook AI — qui permet une recherche de similarité efficace. Elle fonctionne en formant des clusters de ces vecteurs et en utilisant la quantification par produit (Product Quantization) pour réduire la dimensionnalité des données et accélérer les recherches.

Faiss nous permet d'ajouter plusieurs étapes qui peuvent optimiser notre recherche en utilisant de nombreuses méthodes différentes. Une approche populaire consiste

à partitionner l'index en cellules de Voronoï come le montre la figure 2.21 :

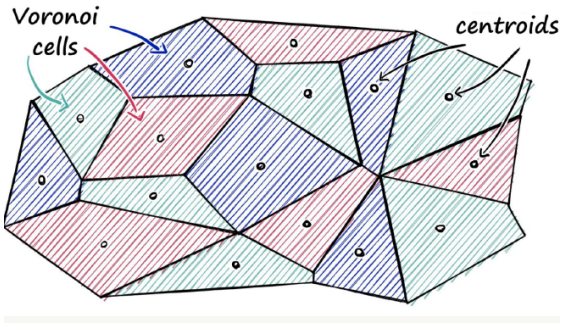


FIG. 2.22: Cellules de Voronoï, source : [54]

2.4 Conclusion

Lors de ce chapitre, nous avons défini les termes et concepts essentiels pour notre projet de fin d'études. Nous avons présenté de manière détaillée les éléments nécessaires pour appréhender notre solution, en expliquant l'évolution vers l'IA Générative ainsi que ses principales composantes, notamment les LLMs, tout en passant par les dernières approches comme le RAG permettant de complémenter et tirer parti au mieux de ces modèles.

Le prochain chapitre portera sur la conception de l'assistant intelligent basé sur l'IA Générative ainsi que sa mise en œuvre pour optimiser les performances des consultants.

Chapitre 3

Résolution de la problématique

Cette section vise à répondre à la problématique identifiée dans le chapitre précédent. Nous examinerons les solutions spécifiques conçues pour répondre aux besoins exprimés par KPMG, visant à améliorer la performance de leur due diligence par la conception d'un assistant intelligent capable d'analyser et de lire des rapports financiers et de faire ressortir des insights importants. La solution proposée sera décrite en détail en passant sur les différentes étapes clés suivies dans notre démarche, puis finalement évaluée afin de valider sa performance et fiabilité, conformément aux référentiels mentionnés dans les chapitres antérieurs.

3.1 Récapitulatif

Avant de rentrer de le vif du sujet, passons en revue les besoins et objectifs qu'on souhaite atteindre, ainsi que les techniques que nous allons utiliser tout en justifiant ces choix.

3.1.1 Besoins et Contraintes :

Le domaine financier présente des défis uniques pour les grands modèles de langage (LLMs) :

1. Premièrement, les modèles nécessitent des connaissances spécifiques sur les sujets, la terminologie et les entreprises/industries financiers, dont la présence dans les données de pré-entraînement est incertaine.

2. Deuxièmement, les modèles doivent disposer d'informations financières actualisées et comprendre les nouvelles pertinentes. Cependant, leurs données datent souvent de plusieurs mois/années avant leur sortie.

3. Troisièmement, les questions financières impliquent souvent un raisonnement numérique, limitation avérée des LLMs [4][5].

4. Quatrièmement, pour répondre aux questions financières, les modèles doivent traiter les entrées non structurées (questions qualitatives en langue naturelle) et structurées (données tabulaires), ces dernières étant plus difficiles sans entraînement supplémentaire [6] [7]

5. Cinquièmement, les modèles doivent gérer plusieurs informations (parfois de sources multiples) et analyser de longs passages de texte, tâche plus ardue que le traitement de chaînes courtes issues d'une source unique.

3.1.2 Justification du choix

Après avoir rappeler les critères à satisfaire lors de notre travail, nous allons à présent expliquer et justifier nos choix de méthodologie pour améliorer notre LLM.

Comme expliquer précédemment, nous avons opté pour l'utilisation du RAG qui va combiner le meilleur de deux mondes en IA : la recherche d'informations (retrieval, qui ne génère pas de réponse originale) et la génération de contenu (qui ne s'appuie que sur les données de son entraînement). Nous avons présenté dans l'état de l'art un tableau comparatif des différentes méthode d'adaptation d'un LLM. Le RAG a été retenu pour les raisons suivantes :

- **Accès à des données en temps réel :** Le RAG permet d'intégrer des sources de données externes pour fournir des informations actuelles et fiables, ce qui est particulièrement important dans les domaines où les faits évoluent rapidement.[35]
- **Réduction des hallucinations :** Le RAG réduit les hallucinations en injectant des connaissances externes pour améliorer la précision des réponses, ce qui est essentiel pour des applications commme celle-ci où la précision est critique.[35]

- **Flexibilité et adaptabilité :** Le RAG permet de créer des assistants virtuels pour n'importe quel domaine ou fonction métier, ce qui est particulièrement utile pour des entreprises qui ont des besoins spécifiques.[35]
- **Meilleure pertinence des résultats :** Le RAG utilise des techniques de recherche sémantique pour fournir des résultats plus pertinents, même avec des requêtes imparfaites, ce qui facilite l'accès à l'information pour les utilisateurs.[35]
- **Réduction des coûts :** Le RAG peut réduire les coûts associés à la formation de modèles et à la collecte de données, car il permet d'utiliser des sources de données existantes et de les intégrer directement dans le modèle.[35]
- **Amélioration de la fiabilité :** Le RAG garantit la fiabilité des réponses en utilisant des sources de données fiables et en permettant aux utilisateurs de vérifier les sources citées, ce qui est essentiel pour une application comme la notre.[35]

3.1.3 Plan d'actions

Après avoir justifié notre choix, nous allons maintenant présenter notre plan d'action afin d'attaquer cette problématique :

1. **Préparations :** nous allons commencer par préparer les éléments nécessaires pour la création d'un premier workflow de RAG basique.
2. **Expérimentations :** nous allons tester différentes configurations de valeurs pour les paramètres de la pipeline du RAG, à savoir : la taille des chunks, le modèle de embedding, le LLM. Nous comparerons, évaluerons et analyserons les résultats obtenus afin de choisir la confirguation optimale qui sera utilisée.
3. **Définition de l'architecture :** après avoir validé les modèle et paramètres à utiliser, nous allons concevoir l'architechture finale de notre application.
4. **Implémentation :** après avoir défini l'architechture de l'application, nous allons implémenter cela en code en utilisant Langchain.
5. **Évaluation :** nous allons évaluer la solution finale obtenue après avoir effectuer les derniers ajustements.
6. **Développement de l'interface utilisateur :** nous allons concevoir l'interface graphique qui servira d'intermédiraire d'interaction avec les utilisateurs.

3.2 Préparations

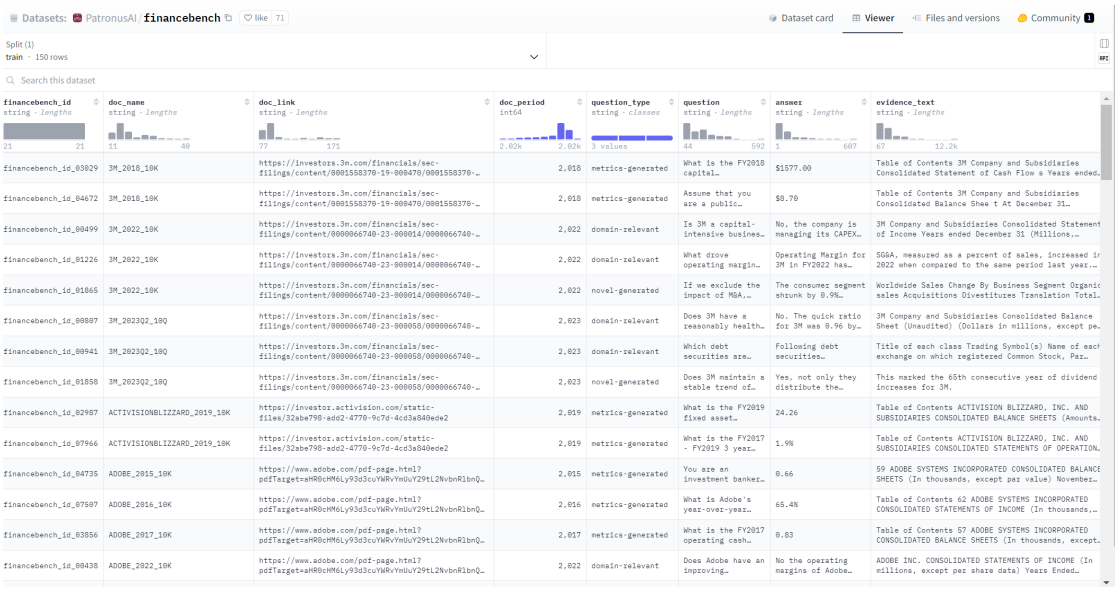
Tout d'abord, nous allons implémenter étape par étape une pipeline de RAG naïf. C'est cette première architecture qui nous servira lors de nos expérimentations mentionnées précédemment.

3.2.1 Sources de données

Notre principale sources de données sera composée de rapports annuels financiers de diverses grandes entreprises disponible via ce dataset HuggingFace : lien vers le dataset.

Présentation du dataset

Cette base de données comporte non seulement les liens vers les PDFs comportant les rapports dont nous avons besoin, mais il servira également ultérieurement de base solide pour l'évaluation des réponses générées, car elle comporte aussi un échantillon de questions financière représentant le premier benchmark référence en industrie en matière de performance LLM sur les questions financières, FinanceBench proposé par Patronus AI.



financebench_id	doc_name	doc_link	doc_period	question_type	question	answer	evidence_text
financebench_id_03029	3M_2018_10K	https://investors.3m.com/financials/sec-filings/content/0001558370-19-000470/0001558370-…	2,018	metrics-generated	What is the FY2018 capital…	$1577.00	Table of Contents 3M Company and Subsidiaries Consolidated Statement of Cash Flow s Years ended…
financebench_id_04672	3M_2018_10K	https://investors.3m.com/financials/sec-filings/content/0001558370-19-000470/0001558370-…	2,018	metrics-generated	Assume that you are a public…	$8.70	Table of Contents 3M Company and Subsidiaries Consolidated Balance Shee t At December 31…
financebench_id_00499	3M_2022_10K	https://investors.3m.com/financials/sec-filings/content/0000066740-23-000014/0000066740-…	2,022	domain-relevant	Is 3M a capital-intensive busines…	No, the company is managing its CAPEX…	3M Company and Subsidiaries Consolidated Statement of Income Years ended December 31 (Millions,…
financebench_id_01226	3M_2022_10K	https://investors.3m.com/financials/sec-filings/content/0000066740-23-000014/0000066740-…	2,022	domain-relevant	What drove operating margin…	Operating Margin for 3M in FY2022 has…	SG&A, measured as a percent of sales, increased in 2022 when compared to the same period last year…
financebench_id_01865	3M_2022_10K	https://investors.3m.com/financials/sec-filings/content/0000066740-23-000014/0000066740-…	2,022	novel-generated	If we exclude the impact of M&A,…	The consumer segment shrunk by 0.9%…	Worldwide Sales Change By Business Segment Organic sales Acquisitions Divestitures Translation Total…
financebench_id_00807	3M_2023Q2_10Q	https://investors.3m.com/financials/sec-filings/content/0000066740-23-000058/0000066740-…	2,023	domain-relevant	Does 3M have a reasonably health…	No. The quick ratio for 3M was 0.96 by…	3M Company and Subsidiaries Consolidated Balance Sheet (Unaudited) (Dollars in millions, except pe…
financebench_id_00941	3M_2023Q2_10Q	https://investors.3m.com/financials/sec-filings/content/0000066740-23-000058/0000066740-…	2,023	domain-relevant	Which debt securities are…	Following debt securities…	Title of each class Trading Symbol(s) Name of each exchange on which registered Common Stock, Par…
financebench_id_01858	3M_2023Q2_10Q	https://investors.3m.com/financials/sec-filings/content/0000066740-23-000058/0000066740-…	2,023	novel-generated	Does 3M maintain a stable trend of…	Yes, not only they distribute the…	This marked the 65th consecutive year of dividend increases for 3M.
financebench_id_02987	ACTIVISIONBLIZZARD_2019_10K	https://investor.activision.com/static-files/32abe798-add2-4770-9c7d-4cd3a840ede2	2,019	metrics-generated	What is the FY2019 fixed asset…	24.26	Table of Contents ACTIVISION BLIZZARD, INC. AND SUBSIDIARIES CONSOLIDATED BALANCE SHEETS (Amounts…
financebench_id_07966	ACTIVISIONBLIZZARD_2019_10K	https://investor.activision.com/static-files/32abe798-add2-4770-9c7d-4cd3a840ede2	2,019	metrics-generated	What is the FY2017 - FY2019 3 year…	1.9%	Table of Contents ACTIVISION BLIZZARD, INC. AND SUBSIDIARIES CONSOLIDATED STATEMENTS OF OPERATION…
financebench_id_04735	ADOBE_2015_10K	https://www.adobe.com/pdf-page.html?pdfTarget=aHR0cHM6Ly93d3cuYWRvYmUuY29tL2NvbnRlbnQ…	2,015	metrics-generated	You are an investment banker…	0.66	59 ADOBE SYSTEMS INCORPORATED CONSOLIDATED BALANCE SHEETS (In thousands, except par value) Novembe…
financebench_id_07507	ADOBE_2016_10K	https://www.adobe.com/pdf-page.html?pdfTarget=aHR0cHM6Ly93d3cuYWRvYmUuY29tL2NvbnRlbnQ…	2,016	metrics-generated	What is Adobe's year-over-year…	65.4%	Table of Contents 62 ADOBE SYSTEMS INCORPORATED CONSOLIDATED STATEMENTS OF INCOME (In thousands,…
financebench_id_03856	ADOBE_2017_10K	https://www.adobe.com/pdf-page.html?pdfTarget=aHR0cHM6Ly93d3cuYWRvYmUuY29tL2NvbnRlbnQ…	2,017	metrics-generated	What is the FY2017 operating cash…	0.83	Table of Contents 57 ADOBE SYSTEMS INCORPORATED CONSOLIDATED BALANCE SHEETS (In thousands, except…
financebench_id_00438	ADOBE_2022_10K	https://www.adobe.com/pdf-page.html?pdfTarget=aHR0cHM6Ly93d3cuYWRvYmUuY29tL2NvbnRlbnQ…	2,022	domain-relevant	Does Adobe have an improving…	No the operating margins of Adobe…	ADOBE INC. CONSOLIDATED STATEMENTS OF INCOME (In millions, except per share data) Years Ended…

FIG. 3.1: Aperçu de la base de données des rapports annuels et questions associées

Développé par les chercheurs en intelligence artificielle de Patronus AI et 15 experts du secteur financier, FinanceBench est un ensemble de 10 000 paires de

questions-réponses de haute qualité et à grande échelle, basé sur des documents financiers publics tels que les formulaires SEC 10-K, SEC 10-Q, SEC 8-K, les rapports de résultats et les transcriptions des appels de résultats. Ce banc d'essai est présenté comme une première ligne d'évaluation pour les modèles de langage sur les questions financières, avec des tests plus avancés à venir [2] [3].

3.2.2 Chargement des données

A présent, nous allons charger nos données (autrement dit nos rapports financiers) qui se trouvent sur ce dataset. Pour se faire, nous allons avoir besoin de deux librairies python qui sont :

- **Pandas :** pour télécharger le dataset HuggingFace et accédez à la colonne comportant les liens des PDFs.
- **PyPDFLoader :** pour charger les PDFs en format document et les rendre exploitanles pour l'étape à suivre.

Nous allons nous contenter de charger une dizaine de PDFs (16) et visualiser dans un premier temps le contenu. Pour rappel, PyPDFLoader transforme chaque fichier PDF en une liste d'objets de type document qui sont les pages de ce même rapport.

La figure 3.2 illustre le nombre de caractères que contient chaque de l'ensemble des PDFs téléchargés :

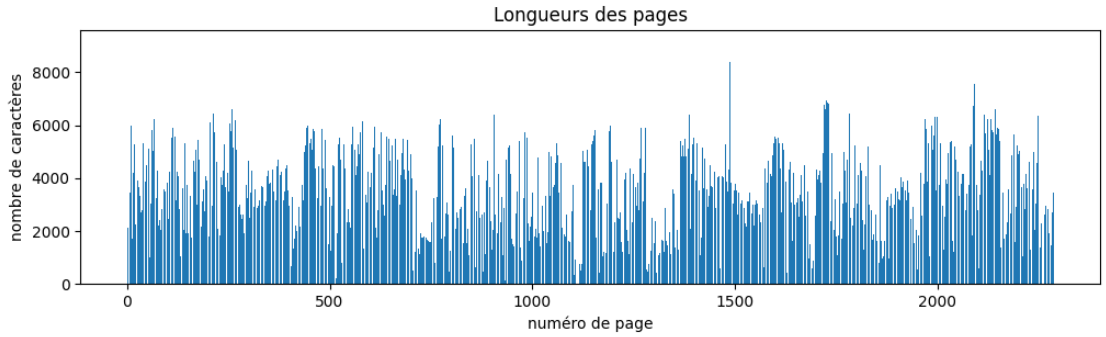


Fig. 3.2: Diagramme en bâton illustrant la longueur des pages

A travers cette visualisation, on constate que la longueur des pages n'est pas assez uniforme et varie énormément de page en page en possédant même dans certain cas des tailles conséquentes allant au delà de 9000 caractères. Ceci se confirme

davantage à travers le calcul de la moyenne et l'écart type de la taille des pages estimés à environ 3462 et 1607 caractères, respectueusement.

3.2.3 Chunking

Comme constaté précédemment, l'importante variabilité des longueurs des pages suggère la nécessite de procéder à quelques traitements textuels.

En effet, si nous utilisions ces grandes sections, nous insérerions beaucoup de contexte bruyant ou indésirable, et comme tous les modèles de langage ont une longueur de contexte maximale, nous ne pourrions pas inclure beaucoup d'autres contextes pertinents.

Par conséquent, nous allons diviser le texte de chaque section en morceaux plus petits. Intuitivement, les petits morceaux encapsuleront un ou quelques concepts et seront moins bruyants que les morceaux plus grands. Nous allons choisir des valeurs typiques pour la division du texte (par exemple, chunk_size=1000) pour créer nos morceaux pour l'instant, mais nous expérimenterons avec une gamme plus large de valeurs plus tard.

Afin d'implémenter le chunking, nous allons faire recours à la fonction *RecursiveCharacterTextSplitter* faisant partie de la librairie *langchain.text_splitter*. La fonction procède de manière récursive pour diviser le texte en segments. Elle commence par tenter de diviser le texte en utilisant des séparateurs plus larges (comme des paragraphes) et, si nécessaire, passe à des séparateurs plus fins (comme des phrases ou des mots) pour s'assurer que chaque segment respecte la taille maximale spécifiée. Elle possède les paramétres suivants :

- **chunk_size :** La taille maximale de chaque segment (en nombre de caractères).
- **chunk_overlap :** Le nombre de caractères qui se chevauchent entre deux segments consécutifs pour maintenir la continuité contextuelle.

La figure 3.3 illustre les nouvelles longueurs obtenues après avoir effectuer le chunking avec une taille de 1000 caractères.

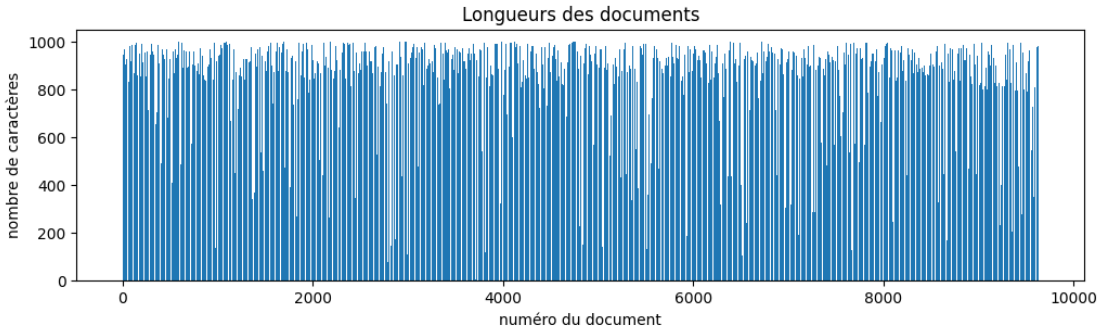


FIG. 3.3: Diagramme en bâton illustrant les tailles des documents après chunking

3.2.4 Embedding

Après avoir fractionné nos sections en petits segments, il nous faut désormais un moyen d'identifier les segments les plus pertinents pour une requête donnée.

Une méthode très efficace et rapide consiste à encoder nos données à l'aide d'un modèle pré-entraîné, puis à utiliser ce même modèle pour encoder la requête. Nous pouvons alors calculer la distance entre les encodages de tous les segments et l'encodage de la requête afin de déterminer les k segments les plus proches.

De nombreux modèles pré-entraînés différents peuvent être choisis pour encoder nos données, mais les plus populaires peuvent être découverts via le classement du Massive Text Embedding Benchmark (MTEB) de HuggingFace [36]. Ces modèles ont été pré-entraînés sur de très grands corpus textuels à l'aide de tâches telles que la prédiction du prochain/mot masqué, ce qui leur a permis d'apprendre à représenter les sous-mots en N dimensions et à capturer les relations sémantiques. Nous pouvons tirer parti de cela pour représenter nos données et identifier les contextes les plus pertinents à utiliser pour répondre à une requête donnée. Nous utilisons les enveloppes d'encodage de Langchain (HuggingFaceEmbeddings) pour charger facilement les modèles et encoder nos segments de documents.

Pour le moment nous allons utilisé le modèle *Alibaba-NLP/gte-large-en-v1.5* qui est relativement bien classé, léger et performant.

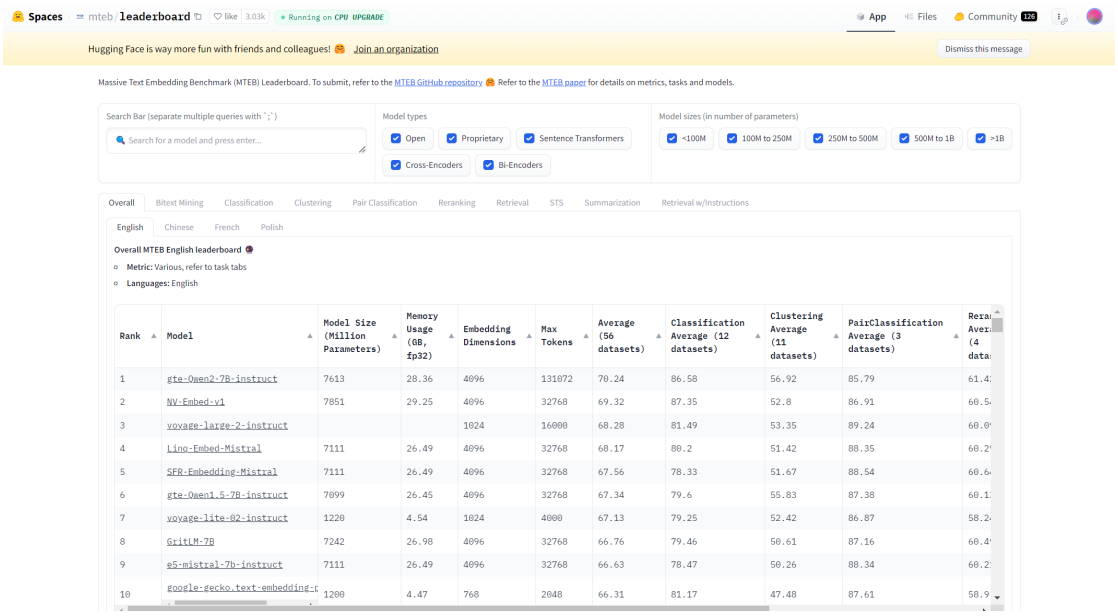


FIG. 3.4: Aperçu du Massive Text Embedding Benchmark (MTEB) Leaderboard

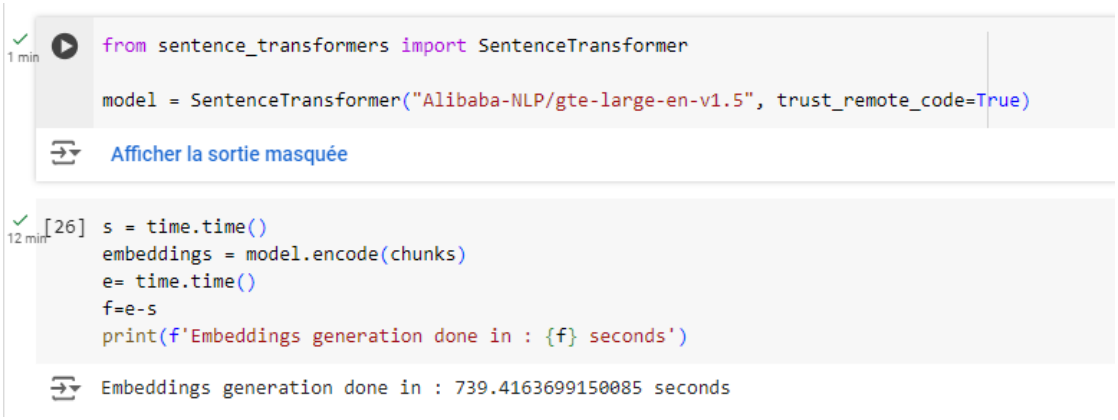


FIG. 3.5: Génération des embeddings

Remarque : Les embeddings ne sont pas le seul moyen de déterminer les segments les plus pertinents. Nous pourrions également utiliser un LLM pour décider ! Cependant, comme les LLMs sont beaucoup plus grands que ces modèles d'embeddings et ont des longueurs de contexte maximales, il est préférable d'utiliser les embeddings pour récupérer les k segments les plus pertinents. Ensuite, nous pourrions utiliser les LLMs sur ces k segments pour déterminer les <k segments à utiliser comme contexte pour répondre à notre requête. Nous pourrions également utiliser le reranking (par exemple, Cohere Rerank) pour identifier les segments

les plus pertinents à utiliser. Nous pourrions également combiner les embeddings avec des méthodes traditionnelles de récupération d'informations, telles que la correspondance de mots-clés, ce qui pourrait être utile pour faire correspondre des tokens uniques qui pourraient potentiellement être perdus lors de l'embedding des sous-tokens.

3.2.5 Stockage des données

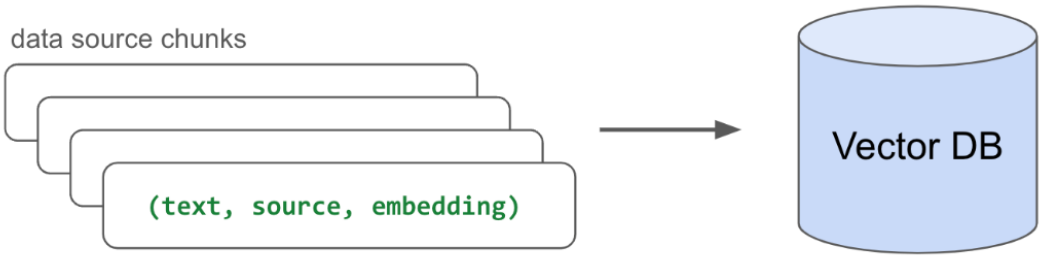


FIG. 3.6: Schéma montrant l'étape de stockage de données

Maintenant que nous avons créé les embeddings de nos chunks, nous devons les indexer (les stocker) quelque part afin de pouvoir les récupérer rapidement pour l'inférence. Bien qu'il existe de nombreuses options de bases de données vectorielles populaires, nous allons utiliser ChromaDB pour sa simplicité et ses performances. Nous allons créer une collection et lui ajouter les embeddings calculés.

```
[ ]  import chromadb

     client = chromadb.Client()
     collection = client.create_collection(name="document_embeddingssss")

     # Add documents and embeddings to the collection
     for i, (text, embedding) in enumerate(zip(chunks, embeddings)):
         collection.add(ids=[f"doc_{i}"], embeddings=[embedding.tolist()], metadatas=[{"text": text}])
```

FIG. 3.7: Code permettant la création de la collection et le stockage des embeddings

3.2.6 Recherche

Avec nos chunks vectorisés et indexés dans notre base de données vectorielle, nous sommes prêts à effectuer une recherche pour une requête donnée. Nous commencerons par utiliser le même modèle d'embedding que nous avons utilisé pour

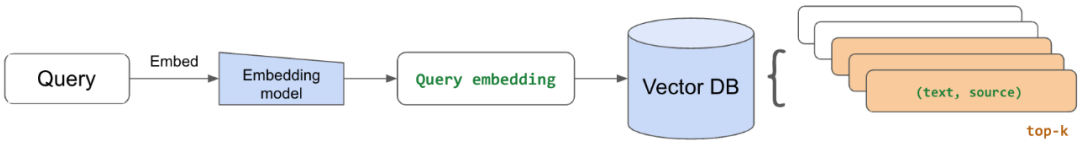


FIG. 3.8: Schéma montrant l'étape de recherche des documents pertinents

vectoriser nos chunks de texte afin de vectoriser la requête entrante. Ensuite, nous récupérerons les chunks les plus pertinents en extrayant les chunks vectorisés les plus proches de notre requête vectorisée. Nous utilisons la distance cosinus, mais il existe de nombreuses options parmi lesquelles choisir. La figure 3.8 illustre le déroulement de cette étape.

Une fois que nous avons récupéré les meilleurs chunks num_chunks, nous pouvons collecter le texte de chaque chunk et l'utiliser comme contexte pour générer une réponse.

```
[ ]  def compute_embeddings(texts):
         embeddings = model.encode(texts)
         return embeddings
```

```
[ ]  def get_relevant_documents(query, k=3):
         query_embedding = compute_embeddings([query])
         results = collection.query(query_embeddings=query_embedding, n_results=k)
         return results
```

FIG. 3.9: Code permettant la recherche de documents pertinent

3.2.7 Génération de réponses

Nous pouvons maintenant utiliser le contexte pour générer une réponse à partir de notre LLM. La figure 3.10 illustre cette étape, où le LLM va avoir en entrée la question de l'utilisateur (query) ainsi que les chunks de texte les plus pertinents (top-k).

Sans ce contexte pertinent que nous avons récupéré, le LLM n'aurait peut-être pas pu répondre avec précision à notre question. Et à mesure que nos données augmentent, nous pouvons tout aussi facilement intégrer et indexer de nouvelles données et les récupérer pour répondre à des questions.

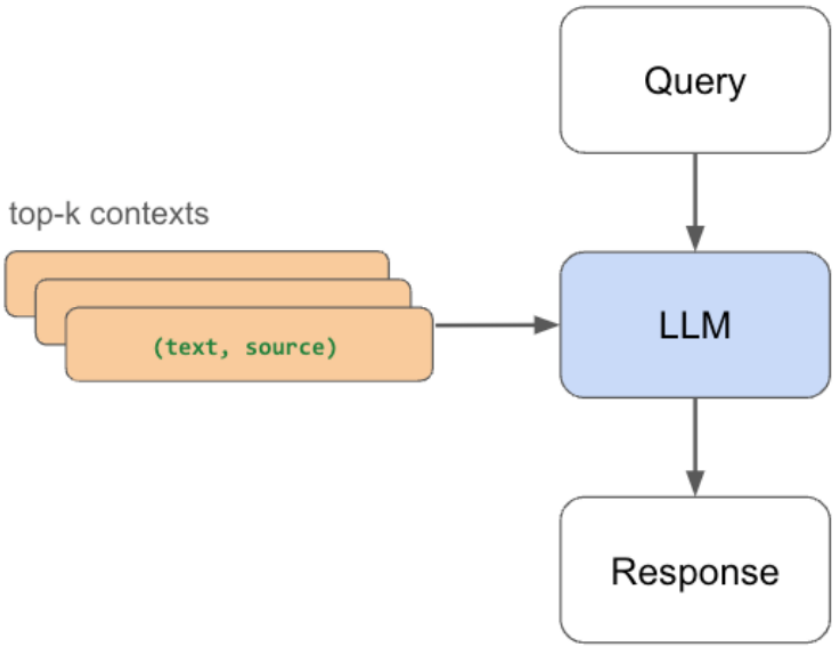


Fig. 3.10: Schéma illustrant l'étape de génération de réponse

Note : Nous utilisons une température de 0,0 pour permettre des expériences reproductibles. Pour les cas d'utilisation qui doivent toujours être factuellement corrects, il est recommandé d'utiliser des valeurs de température très faibles, tandis que les tâches plus créatives peuvent bénéficier de températures plus élevées.

3.3 Expérimentations : Trouver la configuration adéquate

Notre application comporte de nombreux éléments variables : modèles d'embedding, logique de chunking, le LLM lui-même, et bien plus encore. Il est donc important d'expérimenter différentes configurations pour optimiser la qualité des réponses.

Cependant, évaluer et comparer quantitativement ces configurations pour une tâche générative est loin d'être évident. Nous allons décomposer l'évaluation des différentes parties de notre application (recherche à partir d'une requête, génération à partir d'une source), évaluer la performance globale (génération de bout en bout) et partager nos conclusions pour une configuration optimisée.

Mais alors une question se pose : comment y procéder ?

3.3.1 Méthode d'évaluation

Au cœur de l'évaluation des modèles de langage de grande taille (LLMs) se trouve le concept intriguant de l'évaluation de l'IA par l'IA, souvent appelée évaluation assistée par l'IA. Bien que cela puisse sembler initialement comme une boucle paradoxale, cela reflète une pratique de longue date dans l'intelligence humaine, où les individus évaluent souvent leurs propres capacités, que ce soit lors d'entretiens d'embauche ou d'examens académiques. L'avènement des systèmes d'IA avancés permet désormais des capacités d'auto-évaluation similaires dans le domaine de l'intelligence artificielle.

Une tendance émergente dans l'évaluation des LLMs implique l'utilisation de modèles de pointe, tels que GPT-4, pour évaluer non seulement leur propre performance, mais aussi celle d'autres LLMs. Cette approche gagne en popularité en raison de la précision accrue et de la sophistication de ces modèles à la pointe de la technologie. Parmi les outils facilitant cette tendance figurent DeepEval et Prometheus, qui exploitent les capacités des LLMs de premier ordre à des fins d'évaluation. Un cadre notable dans ce domaine est G-Eval, introduit dans un papier intitulé "NLG Evaluation using GPT-4 with Better Human Alignment[32]. La figure 3.11 montre le déroulement de cette démarche.

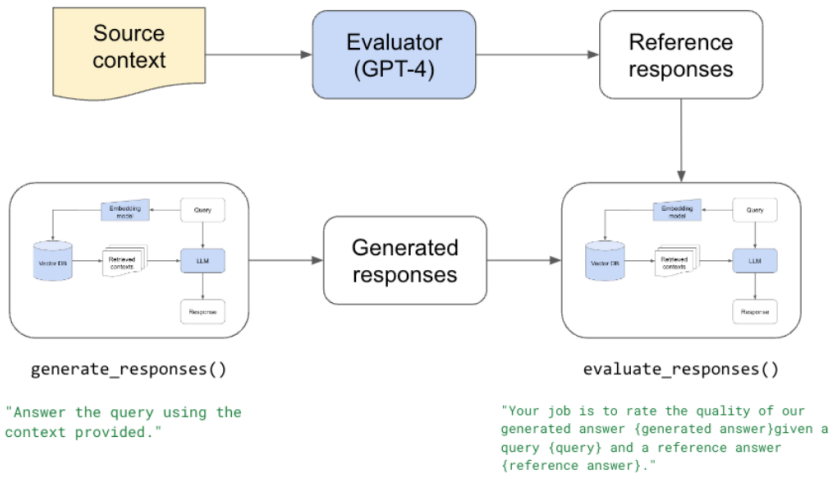


FIG. 3.11: Schéma illustrant le processus d'évaluation des réponses

Avec Giskard, nous pouvons également mettre en œuvre cette méthode d'évaluation assistée par l'IA. Giskard permet de créer des tests d'IA personnalisés pour évaluer les modèles de manière approfondie. En utilisant Giskard, nous pouvons configurer des évaluations détaillées où des modèles avancés comme GPT-4 peuvent être utilisés pour évaluer la performance d'autres LLMs, en s'assurant que nous obtenons une évaluation précise et fiable de nos modèles d'IA [34].

Fig. 3.12: Aperçu sur l'outil Giskard

Après avoir trouvé la configuration optimale, nous aurons ainsi déterminer les pièces maîtresses qui sont le mieux adaptées à nos données, et par la suite, nous essayerons d'implémenter une version de RAG plus avancée avec cette même configuration.

Le schéma 3.13 représente les configurations que nous allons testé :

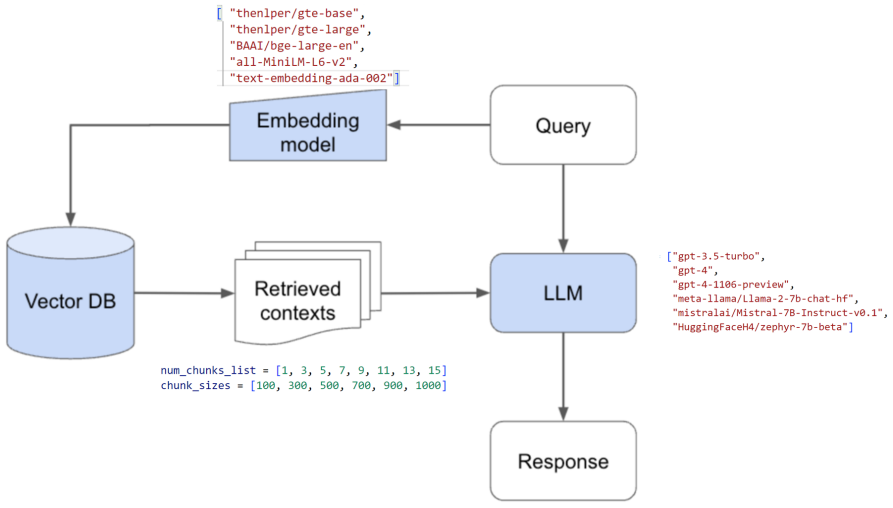


Fig. 3.13: Schéma montrant les différentes configurations à tester

3.3.2 Présentation de résultats :

1. Contexte

Nous allons d'abord tester si le contexte additionnel que nous fournissons est utile. Cela vise à valider que le système RAG en vaut effectivement la peine. Nous pouvons le faire en fixant num_chunks=0 (pas de contexte) et en comparant cela à num_chunks=5.

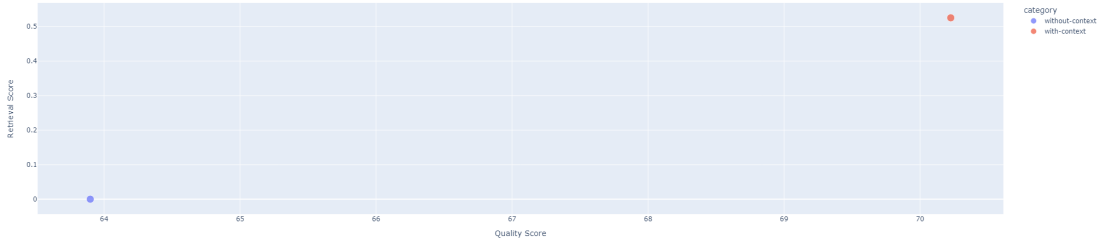


FIG. 3.14: Score des réponses avec et sans contexte

Sanity check : le score de récupération sans contexte est nul puisque nous n'utilisons aucun contexte.

Comme nous pouvons le voir dans la figure 3.14, l'utilisation du contexte (RAG) aide effectivement à améliorer la qualité de nos réponses (et de manière significative).

2. Taille des chunks

Selon la figure 3.15, il semble que des tailles de chunk plus grandes aident mais atteignent un plateau (trop de contexte pourrait être trop bruyant). Les tailles de chunks plus grandes ne sont pas toujours meilleures.

La taille de chunk = 1000 caractères semble être la plus adéquate, nous allons donc fixer cette valeur de la sorte.

Note : Si nous devions utiliser des tailles de chunks plus grandes (basées sur des caractères), il faut garder à l'esprit que la plupart des modèles d'embedding open source ont une longueur maximale de séquence de 512 jetons sous-mots. Cela signifie que si notre chunk contient plus de 512 jetons sous-mots (4 caractères équivaut approximativement à 1 token), l'embedding ne le prendrait pas en compte

de toute façon.

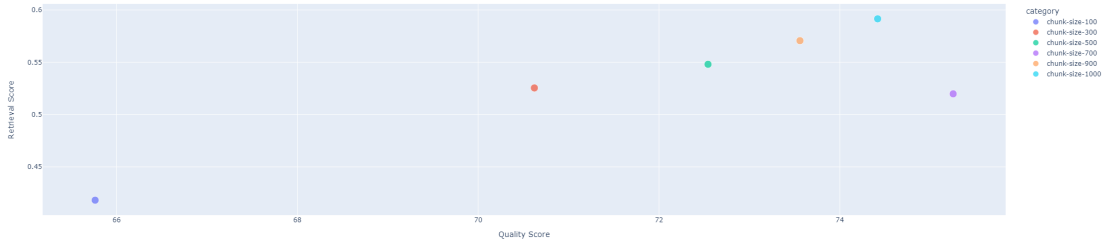


FIG. 3.15: Score obtenus des réponses pour chaque taille de chunk

3. Nombre de chunks

Ensuite, nous allons expérimenter avec le nombre de morceaux à utiliser. Plus de morceaux nous permettront d'ajouter plus de contexte, mais trop pourraient potentiellement introduire beaucoup de bruit.

Note : La taille des chunks que nous avons choisie, multipliée par le nombre de chunks ci-dessous, tient dans la longueur de contexte du LLM. Nous expérimentons la taille des chunks et le nombre de chunks comme s'ils étaient des variables indépendantes, mais ils sont fortement liés. Surtout que tous nos LLM ont une longueur de contexte maximale finie.

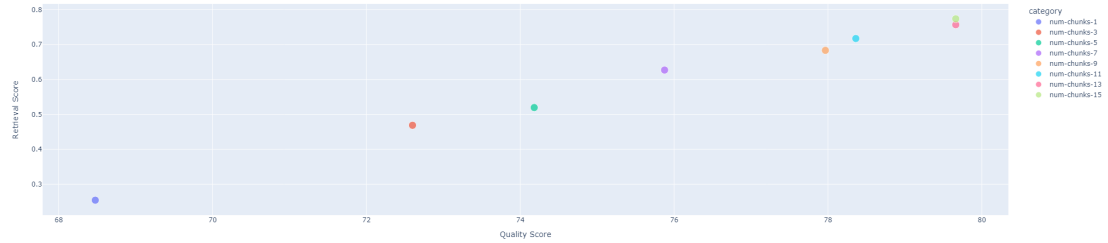


FIG. 3.16: Score obtenus des réponses pour différents nombre de chunks

D'après la figure 3.16, augmenter notre nombre de morceaux améliore nos scores de récupération et de qualité jusqu'à un certain point. Cependant, pour certains modèles (par exemple, llama-2), la longueur du contexte est beaucoup plus courte, nous ne pourrons donc pas utiliser autant de morceaux

4. Modéles d'embedding

Jusqu'à présent, nous avons utilisé thenlper/gte-base comme modèle d'embedding parce que c'est une option relativement petite (0,22 GB) et performante. Mais maintenant, explorons d'autres options populaires telles que le leader actuel du classement MTEB, BAAI/bge-large-en (1,34 GB), thenlper/gte-large (une version plus grande de gte-base),all-MiniLM-L6-v2, et le modèle d'OpenAI, text-embedding-ada-002.

FIG. 3.17: Score obtenus des réponses pour différents modéles d'embedding

La figure 3.17 montre un résultat plutôt intéressant car le modèle numéro 1 (BAAI/bge-large-en) du classement actuel n'est pas nécessairement le meilleur pour notre tâche spécifique. L'utilisation du modèle plus petit thenlper/gte-large a produit les meilleurs scores de récupération et de qualité dans nos expériences.

5. LLM

Nous allons maintenant utiliser les meilleures configurations ci-dessus pour évaluer différents choix pour le modèle de langage principal (LLM).

Note :

- Jusqu'à présent, nous avons utilisé un LLM spécifique pour décider de la configuration, donc les performances de ce LLM seront un peu biaisées ici.
- Cette liste n'est pas exhaustive et même pour les LLMs que nous utilisons, il existe des versions avec des fenêtres de contexte plus longues disponibles.

Sanity check : les scores de récupération sont tous les mêmes car le LLM que nous choisissons n'a pas d'impact sur cette partie de notre application

FIG. 3.18: Score obtenus des réponses pour différents LLMs

La figure 3.18 montre la performance des modèles GPTs de OpenAI face au reste. Nous allons donc opter pour un modéle GPT, mais pour trancher définitivement sur lequel choisir, une analyse supplémentaire sera requise.

Note : Certains de nos modèles LLM ont des longueurs de contexte beaucoup plus grandes, par exemple, GPT-4 a 8192 tokens et GPT-3.5-turbo-16k a 16 384 tokens. Nous pourrions augmenter le nombre de chunks que nous utilisons pour ceux-ci puisque nous avons constaté qu'augmenter le nombre de chunks continuait à améliorer les scores de récupération et de qualité. Cependant, nous allons garder cette valeur fixe pour le moment, car les performances ont de toute façon commencé à diminuer et ainsi nous pouvons comparer ces performances dans les mêmes configurations exactes.

6. Analyse coût

En plus de la performance, nous souhaitons également évaluer le coût de nos configurations, en particulier en raison des prix élevés des grands modèles LLM. Nous allons décomposer cela en deux parties : le coût du prompt et le coût de l'échantillonnage. La taille du prompt correspond au nombre de caractères dans le contenu de notre système, assistant et utilisateur (y compris les contextes récupérés). La taille de l'échantillonnage correspond au nombre de caractères générés par le LLM dans sa réponse. Pour plus de détails sur la tarification, voir [38].

La figure 3.19 illustre l'écart des coûts engendrés par l'utilisation des différents LLMs choisis. On constate que le modéle GPT-3.5-turbo, malgré sa qualité de réponse relativement moins bonne, est considérablement plus économique. Il semble donc présenter un rapport coût/qualité de réponse plutôt intéressant.

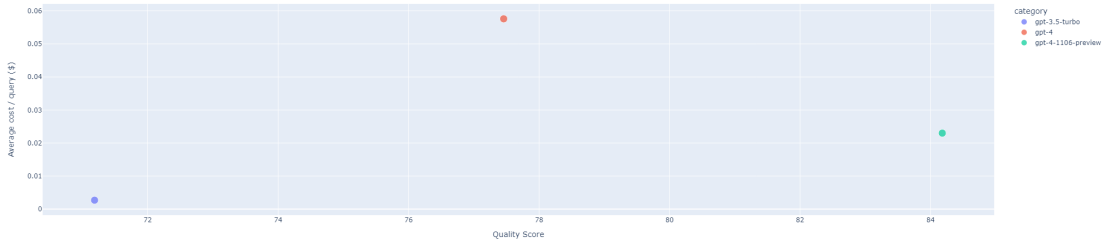


FIG. 3.19: Analyse des coûts des réponses pour différents LLMs

3.3.3 Configuration finale

A l'issue de ces expérimentations, nous pouvons désormais fixer les valeurs optimales pour la configuration de notre solution. La configuration finale est donc :

- Taille des chunks : 1000 caractéres.
- Nombre de chunks (top k) : 15
- Modéle d'embedding : thenlper/gte-large
- LLM : GPT-3.5-turbo

3.4 Architecture de la solution

Tout d'abord nous allons créer un schéma qui représente une architecture commune des applications Assistant IA - RAG .
En détail, cela montre comment les différentes composantes de l'application interagissent pour répondre aux requêtes des utilisateurs comme le montre la figure 3.20.
Pour expliquer le schéma en détail, nous allons suivre chaque étape du processus et clarifier les rôles de chaque composant impliqué dans cette application RAG (Retrieval-Augmented Generation).

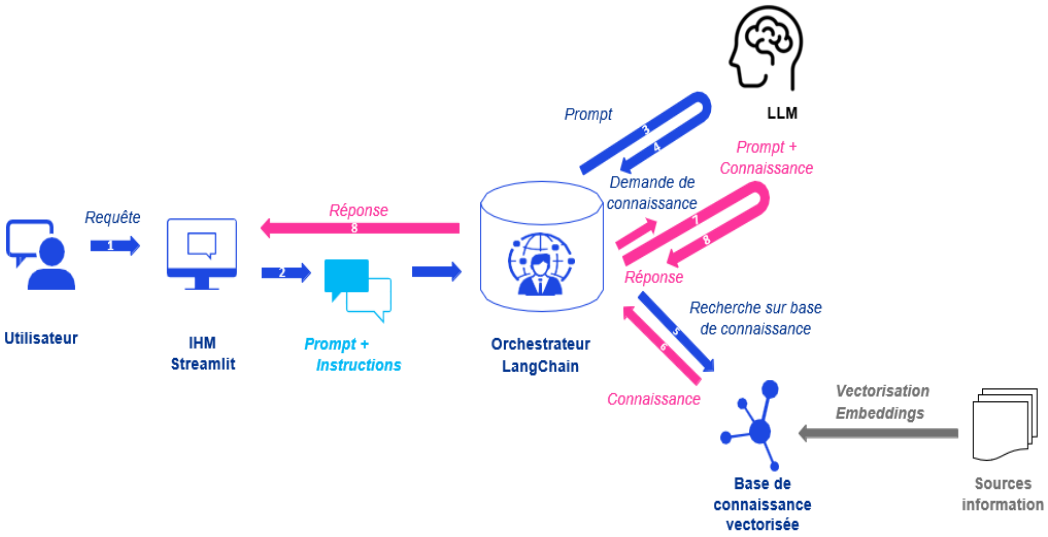


FIG. 3.20: Principe de l'assistant IA

1. Utilisateur

- Requête : L'utilisateur commence en envoyant une requête via une interface comme le montre la figure 3.21. Cette requête peut être une question ou une demande d'information spécifique.

2. IHM Streamlit

- Interface Homme-Machine (IHM) : L'interface utilisateur est construite avec Streamlit, un outil populaire pour créer des applications web interactives. Streamlit est une plate-forme open source qui permet de créer et de déployer facilement des applications Web depuis la machine locale vers le cloud.

FIG. 3.21: intéragir avec l'assistant, 'Capture à partir de l'interface'

FIG. 3.22: Streamlit, source [51]

Streamlit permet de créer des applications Web partageables en quelques minutes en utilisant Python pur sans expérience en développement front-end. elle offre une expérience magique pour créer rapidement des applications interactives et est particulièrement utile pour les applications centrées sur les données en raison de ses nombreuses capacités de visualisation et pour les applications d'IA générative en raison de sa prise en charge intégrée et de ses intégrations pour les interfaces basées sur le chat. [51]

En plus de fonctionner localement, Streamlit peut être déployé à l'aide de diverses

méthodes, notamment le déploiement autogéré sur les fournisseurs de cloud, Community Cloud , Streamlit in Snowflake (SiS) et Snowflake Native Apps .

- Transfert de la requête : La requête de l'utilisateur est envoyée à l'étape suivante dans la chaîne de traitement, en l'occurrence à l'orchestrateur LangChain. À ce moment, la requête peut être accompagnée de prompts et d'instructions supplémentaires pour orienter la réponse.

3. Orchestrateur LangChain

L'orchestrateur LangChain est le cœur du système, orchestrant la communication entre la requête de l'utilisateur, le modèle de langage (LLM) et la base de connaissance vectorisée.

LangChain est une bibliothèque open source de premier plan qui a acquis une grande popularité pour la création d'interfaces de discussion simples et avancées permettant d'interagir avec les modèles LLM et d'autres outils. avancées dans vos applications.

LangChain utilise un certain nombre d'abstractions qui permettent une grande flexibilité lors de la création d'applications basées sur GenAI. L'abstraction homonyme de LangChain est bien sûr **la chaîne .** [52]
LangChain propose deux façons de créer des chaînes. L'un d'entre eux est le nouveau **LangChain Chain Expression Language (LCEL)** qui simplifie le processus de codage en facilitant la composition de chaînes complexes à l'aide d'une syntaxe simple similaire à celle de pipe sous Linux. [52]

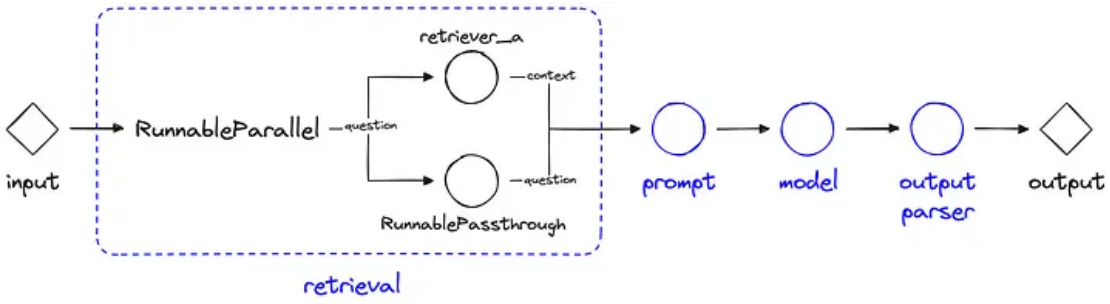


FIG. 3.23: principe LCEL de Langchain, source : [52]

LCEL utilise deux nouveaux objets : **RunnableParallel** et **RunnablePassthrough**.

L' objet **RunnableParallel** nous permet de définir plusieurs valeurs et opérations, et de les exécuter toutes en parallèle. Ici, nous appelons retriever_a en utilisant l'entrée de notre chaîne , puis nous transmettons les résultats de retriever_a au composant suivant de la chaîne via le paramètre « context ».
L' objet **RunnablePassthrough** est utilisé comme un « passthrough » qui prend n'importe quelle entrée dans le composant actuel (récupération) et nous permet de la fournir dans la sortie du composant via la clé « question ».

4. LLM (Large Language Model)

- Génération de texte : Le LLM génère des réponses basées sur les prompts fournis par l'orchestrateur LangChain. Il utilise ses capacités de compréhension et de génération de langage naturel pour traiter les demandes.

Nous avons utilisé le LLM **GPT-3.5-turbo-0125** de OpenAI, Ce tableau fournit une vue d'ensemble complète sur le modèle GPT-3.5-turbo-0125 :

Catégorie	Détail
Nom du modèle	GPT-3.5-turbo-0125
Organisation	OpenAI
Date de sortie	Janvier 2023
Architecture	GPT-3.5 (basé sur GPT-3)
Nombre de paramètres	Environ 175 milliards
Taille des couches	96 couches
Taille des têtes	128 têtes par couche
Dimensions des têtes	96 dimensions par tête
Dimensions des embeddings	12 288 dimensions
Taille du contexte	Jusqu'à 4096 tokens
Optimisation	Utilisation de l'optimisation Adam pour l'entraînement
Entraînement	Entraîné sur des données massives provenant de diverses sources textuelles
GPU utilisés	Plusieurs GPU haute performance (par exemple, NVIDIA A100)
Durée d'entraînement	Plusieurs semaines à plusieurs mois sur des superordinateurs
Données d'entraînement	Environ 570 To de texte brut
Précision	Précision en virgule flottante mixte (FP16)
Évaluation	Utilisation de benchmarks comme GLUE, SuperGLUE, et autres

FIG. 3.24: Tableau présentant des informations générales et techniques sur le modèle GPT-3.5-turbo-0125, source : [53]

5. Base de connaissance vectorielle

- Vectorisation des données : Les informations provenant de diverses sources sont prétraitées et converties en vecteurs (embeddings).
Pour cela nous avons choisi le modèle de Embedding :
- Stockage des informations : Cette base contient des informations structurées et vectorisées, c'est-à-dire converties en vecteurs numériques.
Pour cela nous avons choisis la base de données ChromaDB

Catégorie	Description
Nom	ChromaDB
Type de Base de Données	Base de données NoSQL, orientée documents
Langage de Requêtes	SQL-like, API REST
Langages Supportés	Python, Java, JavaScript, C#, Go, Ruby, PHP
Concurrence	Optimistic Concurrency Control (OCC)
Sécurité	Authentification, Autorisation, Chiffrement des données en transit et au repos
Modèle de Données	Modèle orienté documents, stockage JSON/BSON
Scalabilité	Scalabilité horizontale via le partitionnement et le sharding
Haute Disponibilité	Réplication multi-région, failover automatique
Consistance	Éventuellement consistant, avec options pour la consistance stricte
Performances	Optimisé pour les lectures et écritures rapides, faible latence
Indexation	Indexes secondaires, indexation full-text, géo-indexation
Requêtes	Recherche full-text, recherche géospatiale, agrégations
Transactions	Support des transactions ACID
Sauvegarde	Sauvegardes incrémentales, snapshots
Communauté et Support	Support commercial, documentation complète, communauté active
Intégrations	Intégrations avec Hadoop, Spark, Kafka, et autres systèmes de Big Data
Utilisation Typique	Applications web, applications mobiles, analyse de données, IoT

FIG. 3.25: Tableau récapitulatif des informations générales et techniques sur la base de données ChromaDB, source : [53]

6. Recherche d'information

Lorsque l'orchestrateur interroge cette base, les vecteurs correspondants aux requêtes sont recherchés et les informations pertinentes sont extraites.

Résumé du Flux de Travail

- L'utilisateur envoie une requête via l'interface Streamlit.
- Cette requête est traitée par l'orchestrateur LangChain, qui envoie un prompt au LLM.
- Le LLM génère une réponse initiale.
- Si nécessaire, l'orchestrateur effectue une recherche dans la base de connaissance vectorisée pour compléter ou enrichir la réponse.
- Les informations supplémentaires sont intégrées à la réponse initiale pour former une réponse finale.
- La réponse finale est renvoyée à l'interface utilisateur et présentée à l'utilisateur.

Le schéma 3.26 montre comment une application RAG combine la génération de texte par un modèle de langage avec une recherche de connaissances structurées pour fournir des réponses riches et contextuelles aux utilisateurs.

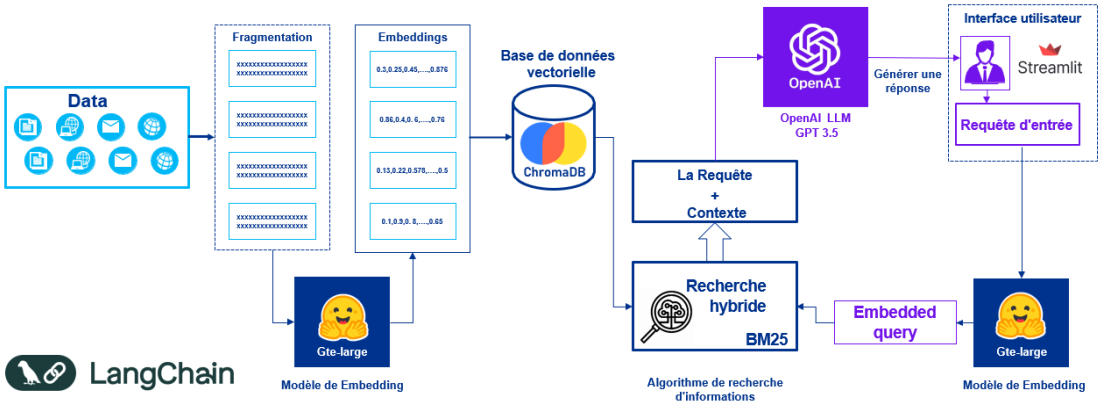


FIG. 3.26: Architecture de la solution

3.5 Implémentation de la solution

1. **Chargement du document**

   Nous allons tout d'abord charger des documents à inclure dans la base de connaissances qui alimente notre chatbot. LangChain propose des abstractions pour les documents et les chargeurs de documents. Les chargeurs de documents peuvent charger certains formats de fichiers locaux ou récupérer des documents depuis internet. Nous allons utiliser quelques chargeurs de documents qui nous aideront à couvrir un ensemble de sources assez large.

   (a) Pages Web HTML en utilisant WebBaseLoader qui extraira le texte du HTML.

   (b) Fichiers téléchargés : soit des fichiers texte brut, soit des fichiers PDF

   (c) Fichiers PDF ou texte récupérés à partir d'URL.

   (d) Fichiers CSV

   (e) Pages Wikipédia utilisant WikipediaLoader pour récupérer les meilleures correspondances pour une requête Wikipédia.

   Chaque chargeur renvoie une liste python d'objets Document, pour s'y faire nous allons créer deux scripts : **local_loader.py** et **remote_loader.py**.

   (a) **local_loader.py** : ici nous Combinons nos fonctions de chargement local.

   (b) **remote_loader.py** : ici nous Combinons nos fonctions de chargement distants.

2. **Diviser les documents en chunks - chunking**
   Nous utilisons RecursiveCharacterTextSplitter de LangChain pour convertir chaque document en une série de morceaux d'une longueur maximale, et nous mettons le code dans un fichier : **splitter.py**.

3. **Vectorisation (Embedding) et stockage dans la base de données vectorielles**
   Nous allons utiliser des transformateurs de phrases pour calculer localement des intégrations de texte à l'aide de modèles pré-entraînés.
   Nous utilisons le modèle **thenlper/gte-large** qui est un modèle d'intégration de texte général (GTE). Vers des intégrations de texte général avec un apprentissage contrastif à plusieurs étapes.
   Les modèles GTE sont formés par Alibaba DAMO Academy. Ils sont principalement basés sur le framework BERT et proposent actuellement trois

tailles de modèles différentes, notamment GTE-large , GTE-base et GTE-small . Les modèles GTE sont formés sur un corpus à grande échelle de paires de textes pertinents, couvrant un large éventail de domaines et de scénarios. Cela permet aux modèles GTE d'être appliqués à diverses tâches en aval des intégrations de texte, notamment la recherche d'informations , la similarité textuelle sémantique , le reclassement de texte , etc..
Puis nous allons stocker nos vecteurs dans la base de données **ChromaDB.** Nous mettons tous le code du vectorisation et crétation de la base de données vectorielles ainsi que le chargement des ces vecteurs dasn un fichier qu'on va appeller **vector_store.py**.

4. **Recherche hybride : ajout d'une récupération basée sur des mots clés**

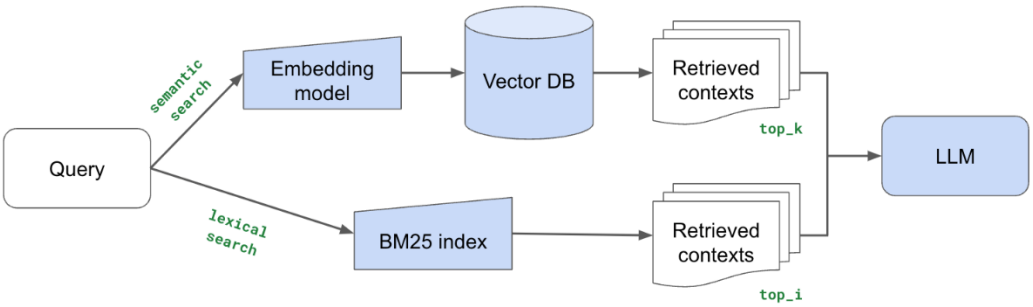


FIG. 3.27: Schéma montrant le processus de recherche hybride

Nous allons utiliser BM25 pour notre recherche basée sur des mots clés. BM25 est un modèle de type sac de mots que nous utiliserons pour classer les résultats de récupération en fonction des mots-clés de la requête présents dans le document.

Nous allons maintenant créer un récupérateur pour BM25, un autre pour notre magasin de vecteurs, et les transmettre tous les deux au récupérateur d'ensemble qui reclassera les résultats.

Et nous utiliserons une chaîne RetrievalQA avec un modèle LLM comme le montre la figure 3.27.

Nous mettons le code pour créer notre modèle dans un fichier appelé **Ensemble.py**.

5. **Ajout de mémoire conversationnelle** Lorsque nous créons un chatbot, nous voulons donc qu'il se souvienne réellement des premières parties de

la conversation. Cela se fait avec les LLM en utilisant l'une des différentes méthodes d'insertion de l'historique des discussions dans le contexte des requêtes adressées au modèle via un modèle d'invite qu'on va implémenter dans le fichier **memory.py**.

6. **Création de notre chaîne RAG complète**
   Pour pouvoir créer une chaine de rag complete nous devons ajouter des quelques fonctions qui facilitent l'utilisation de notre code pour cela : on va créer deux scripts :
   Basic_chain.py :
   pour configurer et retourner un LLM (soit ChatOpenAI soit ChatHuggingFace) basé sur le repo_id fourni et un token API optionnel et ainsi configurer une chaîne de traitement impliquant un prompt et un modèle, et retourne la chaîne pour une utilisation ultérieure.
   Full_chain.py : Créer une chaîne complète pour interagir avec un modèle de langage, en utilisant un retriever pour obtenir du contexte pertinent et une mémoire de chat pour conserver l'historique des messages
   Rag_chain.py : Pour créer une chaîne qui utilise les documents récupérés comme contexte dans une chaîne RAG de base utilisant LCEL.

7. **Création de l'application Streamlit**

   Nous allons maintenant créer une application Streamlit intégrant notre chaîne RAG et notre mémoire conversationnelle.

   Ici, nous utilisons le composant st.chat which ainsi que l' intégration LangChain avec l'historique des messages de discussion Streamlit pour créer une interface utilisateur Web pour notre code.

   Pour Streamlit, au lieu d'un .env fichier, nous allons utiliser st.secrets en créant un fichier dans secrets.tomlun .streamlitsous-répertoire du répertoire de notre application. Notre application est également configurée pour permettre à l'utilisateur de saisir son. propres clés API si aucun secret n'est trouvé.

   Ce code tire parti de l'intégration entre LangChain et Streamlit en utilisant StreamlitChatMessageHistory qui aide LLMChain à mémoriser les messages dans une conversation et à les stocker dans l'état de session de Streamlit afin qu'ils soient automatiquement conservés lors des réexécutions.

   Nous mettons ensuite le code de l'application sur le fichier **streamlit_app.py.**

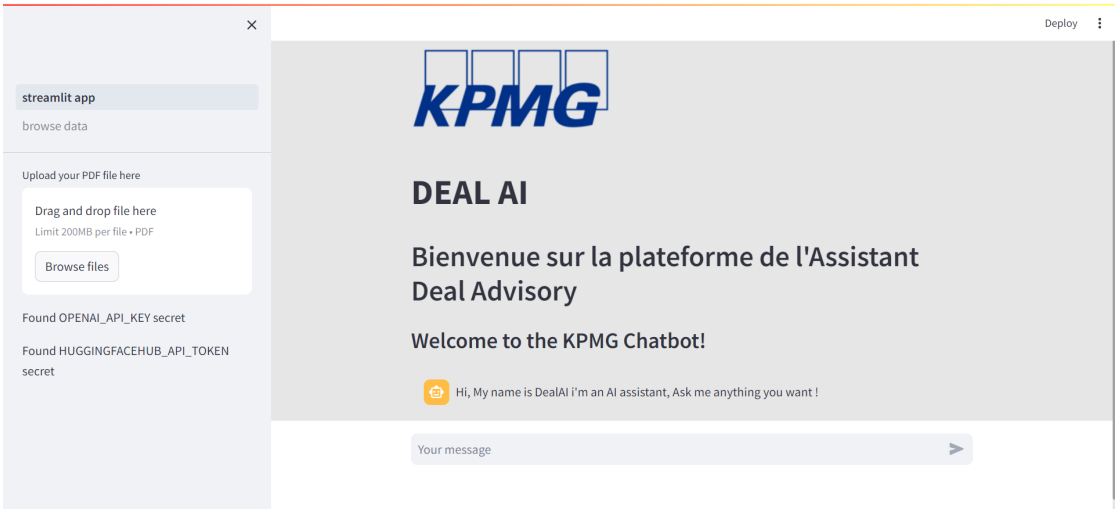


FIG. 3.28: Interface de l'assistant

3.6 Evaluation de la solution

Après avoir conçu et implémenter l'architecture finale de notre assistant, nous allons à présent passer à son évaluation.

Pour se faire, nous allons utiliser encore une fois l'outil RAGET (RAG Evaluation Toolkit) de Giskard. Tout comme dans tout système d'IA, les performances des composants individuels du pipeline LLM et RAG ont un impact significatif sur l'expérience globale. RAGET nous permettra de générer un rapport sous forme de dashboard illustrant la performance de l'assistant à travers les différentes composantes de l'application, ce de manière intuitive et ergonomique [34].

RAGET évalue chaque composant de l'agent RAG en attribuant des scores. Ces scores sont obtenus en agrégeant la justesse des réponses de l'agent sur différents types de questions. Chaque composant est noté sur une échelle de 0 à 100, 100 étant le score parfait. Des scores bas peuvent indiquer les points faibles de votre agent RAG et les composants nécessitant des améliorations [37].

Voici les composants évalués par RAGET :

- **Générateur :** le LLM utilisé par le RAG pour produire les réponses
- **Récupérateur :** récupérer les documents pertinents de la base de connaissances en fonction de la requête utilisateur

- **Réécriveur (optionnel) :** réécrire la requête de l'utilisateur pour la rendre plus pertinente par rapport à la base de connaissances ou tenir compte de l'historique des conversations
- **Routeur (optionnel) :** filtrer la requête de l'utilisateur en fonction de ses intentions (détection des intentions)
- **Base de connaissances :** ensemble des documents fournis au RAG pour générer les réponses

La figure 3.29 illustre les scores obtenus par l'assistant. Analysons-les de plus près :

- **Generator (86.19%) :** Le module de génération a une précision de 86.19%. Cela signifie que le texte généré par le modèle est correct dans la majorité des cas, indiquant une performance élevée.
- **Retriever (82.57%) :** Le module de récupération atteint une précision de 82.57%. Ce résultat montre que le système est efficace pour récupérer les informations pertinentes nécessaires pour répondre aux requêtes.
- **Rewriter (76.35%) :** Le module de réécriture a une précision de 76.35%. Ce score suggère que le système est relativement bon à reformuler correctement les informations récupérées, bien que des améliorations soient encore possibles.
- **Routing (100.0%) :** Le module de routage a une précision parfaite de 100%. Cela indique que le système distribue correctement les requêtes aux modules appropriés sans erreur.
- **Knowledge Base (0.0%) :** Un score de base de connaissances de 0% signifie que les performances du système RAG sont uniformes sur tous les sujets de la base de connaissances. Cela indique une absence de variation entre les scores de correction maximaux et minimaux, reflétant une précision constante dans les réponses générées pour chaque sujet.
- **Overall Correctness Score (91%) :** Le score global de correction est de 91%. Ce score reflète la performance globale du système, suggérant que l'application est très performante dans la plupart des aspects évalués.

Ces résultats montrent que l'application RAG est performante dans la génération, la récupération, la réécriture et le routage des informations, avec une constance

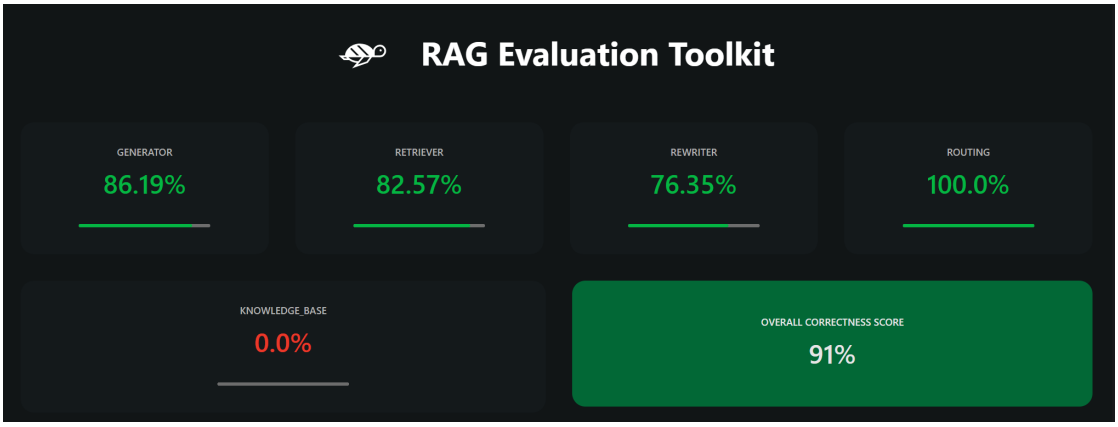


FIG. 3.29: Aperçu des scores de l'évaluation avec l'outil RAGET de Giskard

remarquable dans la précision des réponses sur tous les sujets de la base de connaissances.

La figure 3.30 illustre les niveaux d'exactitude des réponses par rapport à chaque topic (thème) lié à la base de connaissance. Cela nous donne un meilleur aperçu sur les sujets où lesquels l'assistant arrive à mieux répondre, mais aussi où il éprouve quelques difficultés.

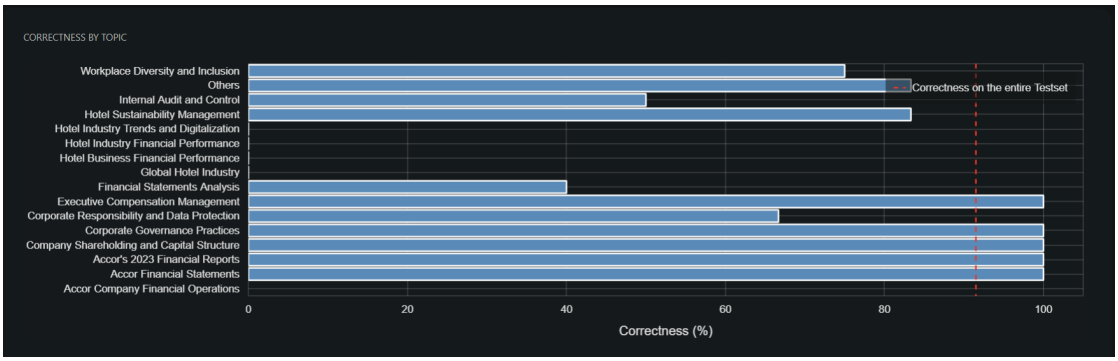


FIG. 3.30: Exactitude des réponses par topic de la knowledge base

La figure 3.31 affiche un nuage de point permettant de visualiser l'ensemble de la base de connaissance en fonction des sujets. Tandis que les figures 3.32 et 3.33 nous montre un zoom sur les réponses (correcte ou fausse) aux questions en respectant les topic de la base de connaissance.

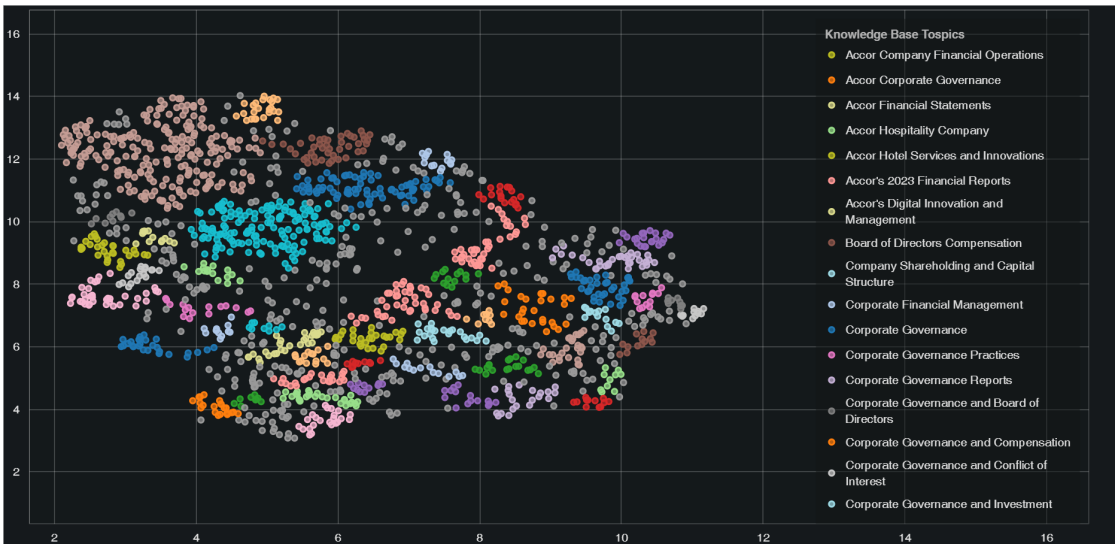


FIG. 3.31: Overview sur les sujets de la base de connaissances

FIG. 3.32: Aperçu sur l'exactitude des réponses aux questions liées à la base de connaissances

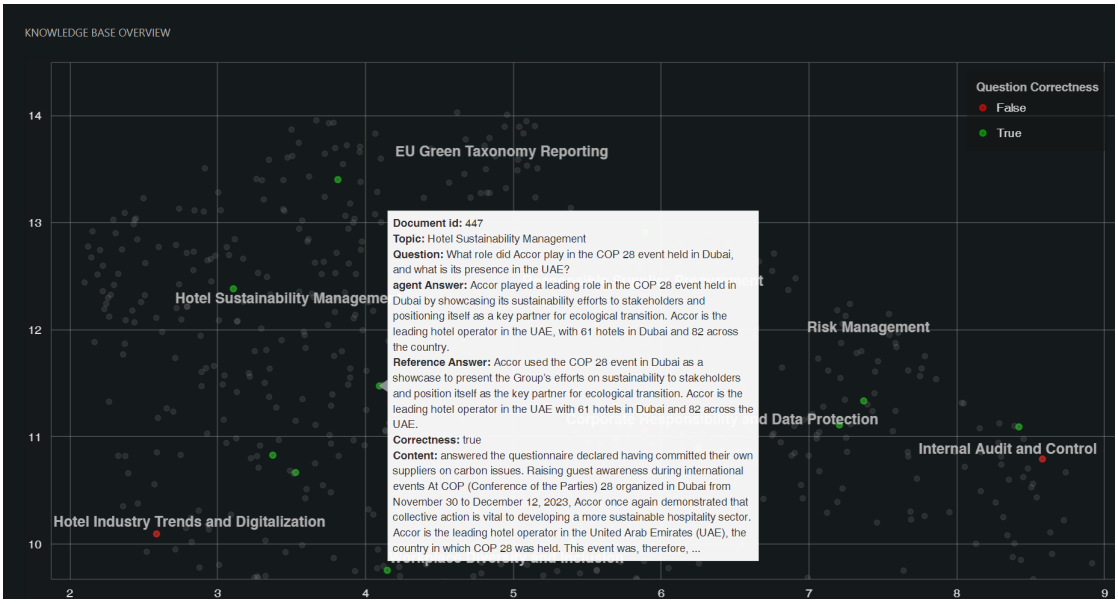


FIG. 3.33: Affichage des détails d'un point donné dans la base de connaissances

3.7 Conclusion

Ce chapitre a présenté le développement complet d'un assistant intelligent répondant aux critères et attentes définis. Nous avons décrit en détail chaque étape du processus de conception de la solution. Plusieurs expériences ont été menées pour déterminer la configuration optimale des paramètres de base de l'application RAG. La solution finale a ensuite été améliorée grâce à une approche RAG avancée, complétée par une recherche hybride utilisant l'algorithme BM25. L'évaluation de notre solution a démontré des résultats concluants, répondant efficacement à la problématique posée.

Perspectives

Les perspectives futures pour notre assistant intelligent, basé sur la génération augmentée par la recherche (RAG) et une recherche hybride, sont vastes et prometteuses. Quatre pistes principales pour des contributions et améliorations potentielles sont explorées :

1. **Les graphes de connaissances** Ils apportent des avantages significatifs en capturant les relations complexes entre entités, améliorant la pertinence et le contexte des données. Leur structure facilite le traitement de requêtes complexes et unifie données structurées et non structurées, augmentant ainsi la précision et la fiabilité des systèmes RAG [40][39].

2. **Combiner le Fine-tuning et le RAG : RAFT** Le Fine-Tuning Augmenté par Récupération (RAFT) fusionne les atouts du RAG et de l'affinage classique. Cette approche intègre des documents spécifiques au domaine durant l'affinage, permettant au modèle de mieux saisir les subtilités propres au domaine ciblé et d'améliorer sa compréhension du contexte externe[41].

3. **Modèles multimodaux** Ces modèles traitent diverses sources d'information (texte, images, audio, vidéo), offrant des réponses plus naturelles et contextuelles. Ils sont particulièrement utiles pour exploiter des sources visuelles complexes comme les présentations PowerPoint, permettant une expérience plus complète et fluide.

4. **Approche Agent** Le RAG multi-agents répartit les tâches entre plusieurs agents spécialisés, offrant une meilleure pertinence, une latence réduite, et une flexibilité accrue. Cette approche s'avère particulièrement efficace pour les cas d'utilisation complexes nécessitant un raisonnement sur diverses sources d'information[42].

En optimisant ces approches, nous pouvons continuer à repousser les frontières des assistants intelligents, créant des outils plus performants, polyvalents et sécurisés, capables de répondre aux besoins croissants des utilisateurs.

Conclusion Générale

L'essor remarquable de l'intelligence artificielle générative, porté par les avancées technologiques telles que les Transformers et les LLMs, ouvre de nouvelles perspectives prometteuses pour améliorer l'efficacité opérationnelle et stimuler l'innovation au sein des entreprises. Dans le secteur stratégique de l'audit et du conseil financier, dominé par les grands cabinets comme KPMG, l'accès rapide à des informations fiables et pertinentes revêt une importance cruciale pour optimiser la productivité et la prise de décision éclairée.

C'est dans cette optique que s'inscrivait notre projet de fin d'études, visant à « **concevoir un assistant intelligent exploitant l'IA générative afin d'améliorer les performances des consultants en Deal Advisory au sein de KPMG Algérie** ». Face aux défis liés au respect des politiques de confidentialité des données et aux limites des solutions commerciales existantes, notre approche visait à développer une solution sur mesure, adaptée aux besoins et pratiques spécifiques du cabinet.

Pour atteindre cet objectif ambitieux, nous avons adopté une méthodologie novatrice combinant l'IA générative, les modèles de fondation de pointe tels que les Transformers, ainsi que des techniques avancées comme le RAG (Retrieval Augmented Generation). Cette démarche holistique nous a permis de concevoir un outil performant et adapté, capable de soutenir efficacement les consultants dans leurs tâches d'analyse et d'interprétation des données financières complexes.

L'implémentation réussie de notre assistant intelligent a démontré sa capacité à faciliter l'accès rapide aux informations pertinentes, optimisant ainsi les flux de travail et la prise de décision éclairée au sein du département Deal Advisory. Grâce à cet outil novateur, les consultants peuvent désormais se concentrer sur des tâches à plus forte valeur ajoutée, tandis que les processus analytiques chronophages sont pris en charge de manière autonome et fiable par l'assistant.

Les résultats probants obtenus soulignent le potentiel considérable de l'IA généra-

tive pour améliorer la productivité et l'efficacité opérationnelle au sein de KPMG Algérie. Ils ouvrent de nouvelles perspectives pour une adoption plus large de ces technologies de pointe au sein du cabinet, renforçant ainsi sa position de leader dans le domaine du conseil et de l'expertise comptable. L'intégration réussie de notre solution démontre la capacité de KPMG à s'adapter aux évolutions technologiques et à tirer parti des innovations pour offrir des services toujours plus performants à ses clients.

Au-delà des gains opérationnels, ce projet met en lumière l'importance stratégique du développement de solutions sur mesure, adaptées aux exigences spécifiques des entreprises en matière de confidentialité et de sécurité des données. En contournant les limites des solutions commerciales existantes, notre approche garantit un contrôle total sur les modèles utilisés et les données d'entraînement, assurant ainsi une conformité optimale avec les politiques internes de KPMG.

En définitive, cette étude souligne le rôle clé que peut jouer l'IA générative dans la transformation numérique des entreprises de conseil financier. En exploitant pleinement le potentiel de ces technologies émergentes, tout en respectant les contraintes réglementaires et en développant des solutions sur mesure, les cabinets comme KPMG peuvent se doter d'un avantage concurrentiel durable et renforcer leur leadership sur un marché en constante évolution.

Bibliographie

[1] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, et A. Mian, « A comprehensive overview of large language models », arXiv preprint arXiv :2307.06435, 2023. [En ligne]. Disponible sur : https://arxiv.org/pdf/2307.06435

[2] P. Islam, A. Kannappan, D. Kiela, R. Qian, N. Scherrer, et B. Vidgen, « Financebench : A new benchmark for financial question answering », arXiv preprint arXiv :2311.11944, 2023. [En ligne]. Disponible sur : https://arxiv.org/pdf/2311.11944

[3] Patronus AI, « Patronus AI launches FinanceBench, the industry's first benchmark for LLM performance on financial questions », 2023. [En ligne]. Disponible sur : https://www.patronus.ai/announcements/patronus-ai-launches-financebench-the-industrys-first-benchmark-for-llm-perform

[4] P. Lu, L. Qiu, W. Yu, S. Welleck, and K. W. Chang, « A survey of deep learning for mathematical reasoning », arXiv preprint arXiv :2212.10535, 2022. [En ligne]. Disponible sur : https://arxiv.org/pdf/2212.10535

[5] S. Imani, L. Du, and H. Shrivastava, « Mathprompter : Mathematical reasoning using large language models », arXiv preprint arXiv :2303.05398, 2023. [En ligne]. Disponible sur : https://arxiv.org/pdf/2303.05398

[6] F. Zhu, W. Lei, Y. Huang, C. Wang, S. Zhang, J. Lv, et T. S. Chua, « TAT-QA : A question answering benchmark on a hybrid of tabular and textual content in finance », arXiv preprint arXiv :2105.07624, 2021. [En ligne]. Disponible sur : https://arxiv.org/pdf/2105.07624

[7] L. Zha, J. Zhou, L. Li, R. Wang, Q. Huang, S. Yang, et J. Zhao, « TableGPT : Towards unifying tables, nature language and commands into one GPT », arXiv preprint arXiv :2307.08674, 2023. [En ligne]. Disponible sur : https://arxiv.org/pdf/2307.08674

[8] HuggingFace, « FinanceBench Dataset », 2023. [En ligne]. Disponible sur : https://huggingface.co/datasets/PatronusAI/financebench

[9] Q. Xie, W. Han, X. Zhang, Y. Lai, M. Peng, A. Lopez-Lira, et J. Huang, « Pixiu : A large language model, instruction data and evaluation benchmark for finance », arXiv preprint arXiv :2306.05443, 2023. [En ligne]. Disponible sur : https://arxiv.org/pdf/2306.05443

[10] P. Garg, « The future of consulting in the age of Generative AI », EY, 2023. [En ligne]. Disponible sur : https://www.ey.com/en_in/consulting/the-future-of-consulting-in-the-age-of-generative-ai

[11] S. J. Russell, P. Norvig, and E. Davis, Artificial intelligence : A Modern Approach. Prentice Hall, 2010.

[12] H. Gil de Zúñiga, M. Goyanes, and T. Durotoye, "A scholarly definition of artificial intelligence (AI) : advancing AI as a conceptual framework in communication research," Political Communication, vol. 41, no. 2, pp. 317-334, 2024.[En ligne]. Disponible sur : https://www.researchgate.net/publication/374919300_A_Scholarly_Definition_of_Artificial_Intelligence_AI_Advancing_AI_as_a_Conceptual_Framework_in_Communication_Research

[13] J. Haugeland, Ed., « Artificial Intelligence : The Very Idea », MIT Press, 1985.

[14] R. E. Bellman, « An Introduction to Artificial Intelligence : Can Computers Think ? », Boyd Fraser Publishing Company, 1978.

[15] E. Charniak and D. McDermott, « Introduction to Artificial Intelligence », Addison-Wesley, 1985.

[16] P. H. Winston, « Artificial Intelligence », 3rd ed., Addison-Wesley, 1992.

[17] R. Kurzweil, « The Age of Intelligent Machines », MIT Press, 1990.

[18] E. Rich and K. Knight, « Artificial Intelligence », 2nd ed., McGraw-Hill, 1991.

[19] D. Poole, A. K. Mackworth, and R. Goebel, « Computational Intelligence : A Logical Approach », Oxford University Press, 1998.

[20] N. J. Nilsson, « Artificial Intelligence : A New Synthesis », Morgan Kaufmann, 1998.

[21] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing : an introduction," Journal of the American Medical Informatics Association, vol. 18, no. 5, pp. 544-551, 2011.

[22] K. Martineau, « What is generative AI ? », IBM Research, 20 avril 2023. [En ligne]. Disponible sur : https://research.ibm.com/blog/what-is-generative-AI

[23] N. Buhl, « The Full Guide to Foundation Models », Encord, 21 avril 2023. [En ligne]. Disponible sur : https://encord.com/blog/foundation-models/

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et I. Polosukhin, « Attention is all you need », Advances in Neural Information Processing Systems, vol. 30, 2017. [En ligne]. Disponible sur : https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[25] S. Isci, "Décryptage de l'architecture de conception des assistants virtuels IA : un examen approfondi des composants de conception," Medium, Feb. 15, 2024. [En ligne]. Disponible sur : https://medium.com/@senol.isci/decoding-the-ai-virtual-assistant-design-architecture-an-in-depth-look-into-des

[26] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, et P. Liang, « On the opportunities and risks of foundation models », arXiv preprint arXiv :2108.07258, 2021. [En ligne]. Disponible sur : https://arxiv.org/pdf/2108.07258

[27] A. Saleem, « How to tune LLM Parameters for optimal performance », Data Science Dojo, 11 septembre 2023. [En ligne]. Disponible sur : https://datasciencedojo.com/blog/tuning-optimizing-llm-parameters/

[28] 3Blue1Brown, « But what is a GPT ? Visual intro to transformers | Chapter 5, Deep Learning », YouTube. [En ligne]. Disponible sur : https://youtu.be/wjZofJX0v4M?si=6wvlCAiEahbFGGaf

[29] 3Blue1Brown, « Attention in transformers, visually explained | Chapter 6, Deep Learning », YouTube. [En ligne]. Disponible sur : https://youtu.be/eMlx5fFNoYc?si=L3UOuA5t-MA6DAvQ

[30] B. Ceylan, « Large Language Model Evaluation in 2024: 5 Methods », AIMultiple, 2 juillet 2024. [En ligne]. Disponible sur : https://research.aimultiple.com/large-language-model-evaluation/

[31] A. Quinn, « LLM Validation and Evaluation », Iguazio Acquired by McKinsey Company, 21 mai 2024. [En ligne]. Disponible sur : https://www.iguazio.com/blog/llm-validation-and-evaluation/

[32] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, « G-eval : NLG Evaluation using GPT-4 with better human alignment », arXiv preprint arXiv :2303.16634, 2023. [En ligne]. Disponible sur : https://arxiv.org/pdf/2303.16634

[33] E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Deep Neural Network Language Models," in WLM@NAACL-HLT, 2012.

[34] B. Rivera Campos, « Guide to LLM evaluation and its critical impact for businesses », Giskard, 10 avril 2024. [En ligne]. Disponible sur : https://www.giskard.ai/knowledge/guide-to-understand-llm-evaluation

[35] G. Ferreira, « Deploy Your LLM Chatbot with Retrieval Augmented Generation (RAG) », Intellias, 28 mars 2024. [En ligne]. Disponible sur : https://intellias.com/deploy-llm-chatbot-with-rag/

[36] Massive Text Embedding Benchmark (MTEB) Leaderboard. [En ligne]. Disponible sur : https://huggingface.co/spaces/mteb/leaderboard

[37] Giskard Documentation, « RAGET Evaluation ». [En ligne]. Disponible sur : https://docs.giskard.ai/en/latest/open_source/testset_generation/rag_evaluation/index.html

[38] Rubrique Pricing de la plateforme OpenAI. [En ligne]. Disponible sur : https://openai.com/api/pricing/

[39] T. Bratanič, « Enhancing the Accuracy of RAG Applications With Knowledge Graphs », Graph ML and GenAI Research, Neo4j, 30 mars 2024. [En ligne]. Disponible sur : https://neo4j.com/developer-blog/enhance-rag-knowledge-graph/

[40] T. Bratanič, « Using a Knowledge Graph to Implement a RAG Application », Graph ML and GenAI Research, Neo4j, 12 mars 2024. [En ligne]. Disponible sur : https://neo4j.com/developer-blog/knowledge-graph-rag-application/

[41] R. Ong, « What is RAFT ? Combining RAG and Fine-Tuning To Adapt LLMs To Specialized Domains », Datacamp, mai 2024. [En ligne]. Disponible sur : https://www.datacamp.com/blog/what-is-raft-combining-rag-and-fine-tuning

[42] A. Alcaraz et R. Turner, « Enhancing RAG with a Multi-Agent System », 6 avril 2024. [En ligne]. Disponible sur : https://superlinked.com/vectorhub/articles/enhancing-rag-multi-agent-system

[43] J. Foster, "Language Models," Medium, Mar. 22, 2022. [En ligne]. Disponible sur : . [Accessed : Jul. 3, 2024]. https://medium.com/ingeniouslysimple/language-models-15e45dce0805

[44] M. Chui, E. Hazan, R. Roberts, A. Singla, K. Smaje, A. Sukharevsky, et R. Zemmel, « The economic potential of generative AI : The next productivity frontier » , McKinsey, 2023. [En ligne]. Disponible sur : https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier

[45] M. Chui, R. Roberts, T. Rodchenko, L. Yee, A. Singla, A. Sukharevsky, et D. Zurkiya, « What every CEO should know about generative AI » , McKinsey Digital, 2023. [En ligne]. Disponible sur : https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/what-every-ceo-should-know-about-generative-ai?cid=njt-soc-lkn-mbm-mbm--2403--njt-bam-web&linkId=369127971

[46] J. Fernandez, K. Lueth, et P. Wegner, « Generative AI Market Report 2023–2030 », IOT Analytics, 2023. [En ligne]. Disponible sur : https://iot-analytics.com/product/generative-ai-market-report-2023-2030/

[47] Statista, « IA générative : un marché en plein essor », 2023. [En ligne]. Disponible sur : https://fr.statista.com/infographie/30941/evolution-taille-du-marche-ia-generative-et-parts-de-marche-principaux-acteurs/

[48] Statista, « IA générative : un marché encombré », 2022. [En ligne]. Disponible sur : https://fr.statista.com/infographie/31228/utilisateurs-mondiaux-ia-generation-de-texte/

[49] Statista, « Combien les entreprises investissent-elles dans l'intelligence artificielle ? », 2022. [En ligne]. Disponible sur : https://fr.statista.com/infographie/31325/investissements-des-entreprises-dans-intelligence-artificielle/

[50] QuantumBlack AI by McKinsey, «[Exclusive] QuantumBlack Round-table // Gen AI Buy vs Build, Commercial vs Open Source », YouTube, 15 février 2023. [En ligne]. Disponible sur : https://www.youtube.com/watch?v=IpXZGXeuHt4.

[51] Streamlit Documentation, Streamlit. [En ligne]. Disponible : https://streamlit.io/

[52] LangChain Documentation, LangChain.[En ligne]. Disponible : https://www.langchain.com/

[53] OpenAI Documentation, OpenAI. [En ligne]. Disponible : https://www.openai.com/.

[54] Pinecone Documentation, Pinecone. [En ligne]. Disponible : https://www.pinecone.io/

[55] Similarity Metrics for Vector Search, Zilliz Blog, Dec. 15, 2023. [En ligne]. Disponible : https://zilliz.com/blog/similarity-metrics-for-vector-search.

[56] Deep Dive : How Do Vector Databases Work ?, The AI Edge Newsletter, Jun. 20, 2024. [En ligne]. Disponible : https://newsletter.theaiedge.io/p/deep-dive-how-do-vector-databases.

[57] R. P. Padhy, "Fine-Tuning vs. Prompting vs. RAG : Which to Pick for Your LLM ?," LinkedIn, Jun. 30, 2024. [En ligne]. Disponible : https://www.linkedin.com/pulse/fine-tuning-vs-prompting-rag-which-pick-your-llm-dr-rabi-prasad-722cc/

Annexe A

Le marché de la Gen AI

Impacts économiques

L'IA générative, notamment les modèles de langage de grande envergure (LLMs) tels que GPT-4 de OpenAI et Bard de Google, révolutionne de nombreux secteurs économiques. Selon une étude du cabinet de conseil McKinsey & Company réalisée en 2023, l'impact économique de l'IA Générative en industrie serait estimé à environ 2.6 jusqu'à 4.4 billions annuellement [44].

Impact économique global

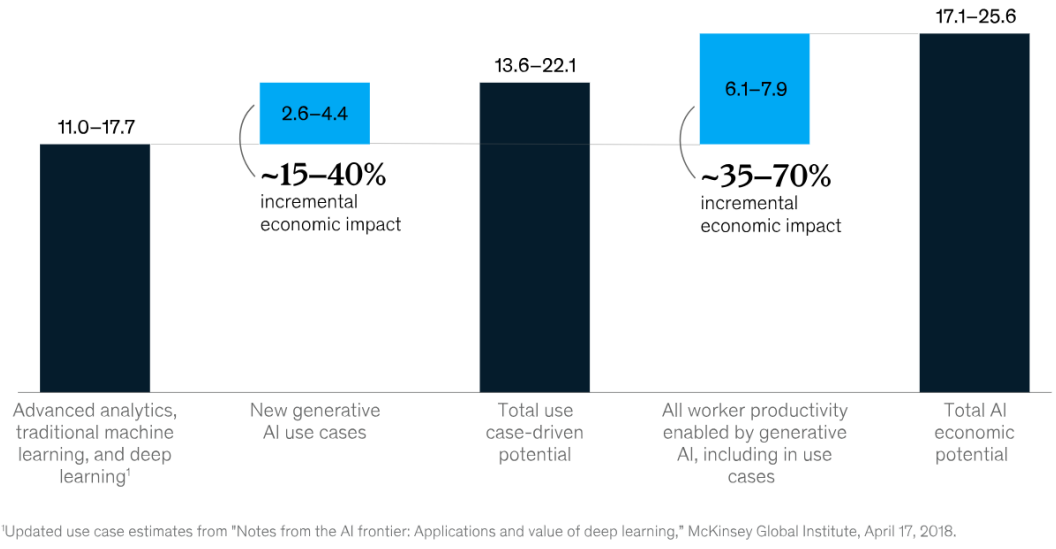


FIG. A.1: Potentiel impact économique global de l'IA, source : [44]

Le premier point de vue examine comment les organisations peuvent adopter l'IA générative, définie comme l'application ciblée de cette technologie à un défi spécifique de l'entreprise pour obtenir des résultats mesurables. Par exemple, en marketing, l'IA générative peut créer des emails personnalisés, réduisant les coûts et augmentant les revenus. Nous avons identifié 63 cas d'utilisation dans 16 fonctions d'entreprise, pouvant générer entre 2,6 et 4,4 trillions de dollars de bénéfices économiques annuels [44].

Cela ajouterait entre 15 et 40 % aux 11 à 17,7 trillions de dollars que l'IA non générative et l'analytique pourraient générer, comparé à notre estimation de 2017 de 9,5 à 15,4 trillions de dollars [44].

Le second point de vue analyse l'impact potentiel de l'IA générative sur 850 professions en modélisant plus de 2 100 activités de travail. Cela permet d'estimer comment l'IA générative pourrait améliorer la productivité mondiale. En éliminant les chevauchements avec les réductions de coûts, les bénéfices économiques totaux de l'IA générative s'élèvent à 6,1 à 7,9 trillions de dollars par an [44].

Impact économique sur les fonctions

Bien que l'IA générative soit une technologie en rapide évolution, les autres applications de l'IA continuent de représenter la majorité de la valeur potentielle de l'IA. Les algorithmes d'analyse avancée et d'apprentissage automatique sont efficaces pour des tâches comme la modélisation prédictive et trouvent constamment de nouvelles applications. Cependant, l'IA générative, en se développant, peut ouvrir de nouveaux horizons en matière de créativité et d'innovation [44].

Nous mettons en lumière le potentiel de valeur de l'IA générative dans les fonctions de l'entreprise.

Selon cette visualisation du rapport de Mckinsey, l'IA générative pourrait impacter la plupart des fonctions de l'entreprise, mais quatre se distinguent par leur part de coûts fonctionnels : les opérations client, le marketing et les ventes, l'ingénierie logicielle, et la recherche et développement. Ces fonctions pourraient représenter environ 75 % de la valeur annuelle totale des cas d'utilisation de l'IA générative.

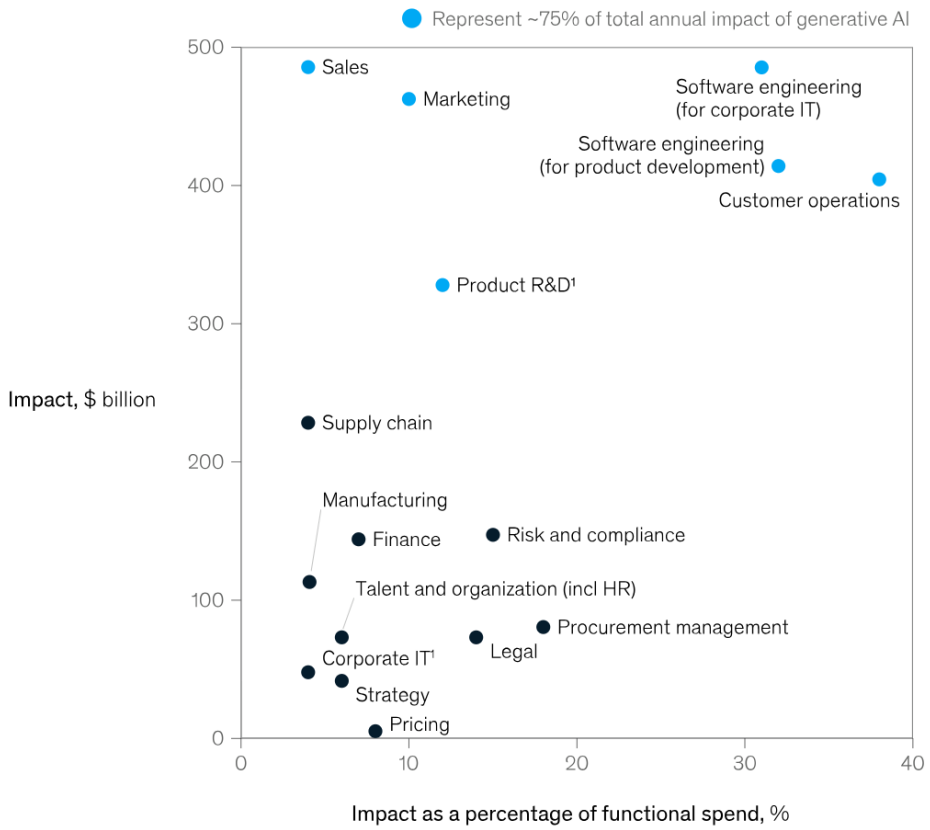


FIG. A.2: La potentielle valeur ajoutée de l'IA dans les fonctions métiers, source [44]

Principaux acteurs du marché

Le rapport sur le marché de l'IA générative 2023–2030, réalisé par IOT Analytics, analyse les GPUs pour Data Centers, les modèles de fondations et plateformes d'IA Générative, ainsi que les services d'IA générative de 2023 à 2030. Il fournit les tailles actuelles du marché, les prévisions et les parts de marché des principaux fournisseurs [46].

Le marché des GPU pour centres de données est dominé par Nvidia, qui détient une part de marché impressionnante de 92 %. AMD est le deuxième plus grand fournisseur avec 3 %. Les dépenses totales ont atteint 49 milliards de dollars en 2023, contre seulement 17 milliards en 2022, et devraient atteindre près de 160 milliards de dollars d'ici 2030.

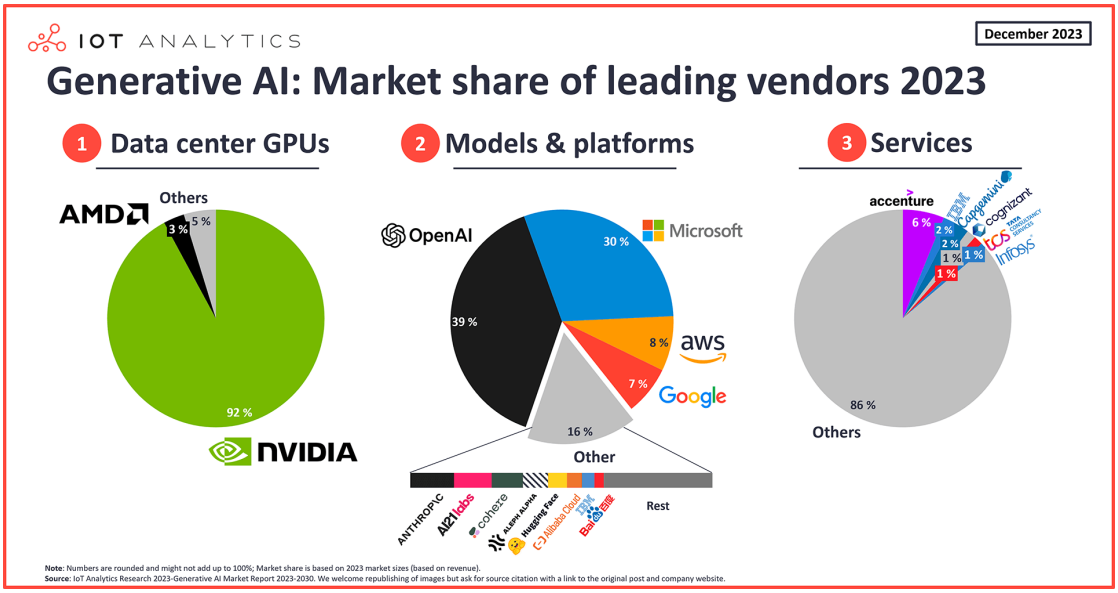


FIG. A.3: Part de marché des principaux fournisseurs de l'IA générative, source [46]

Le marché des modèles fondamentaux et des plateformes d'IA générative est beaucoup plus fragmenté. OpenAI est le leader actuel avec une part de marché de 39 %, suivi de près par son allié et investisseur Microsoft avec 30 %. Les dépenses, déjà de 3 milliards de dollars en 2023, devraient atteindre 92 milliards de dollars en 2030.

En 2023, de nombreuses entreprises ont tenté de définir leur stratégie en matière d'IA générative, et les fournisseurs de services d'IA générative ont réalisé un chiffre d'affaires de 3,2 milliards de dollars cette année seulement. Accenture mène la course avec une part de marché de 6 % et prévoit d'investir 3 milliards de dollars pour se positionner comme le principal fournisseur de services d'IA générative.

Tendances du marché

En pleine expansion, le marché de l'IA générative devrait atteindre 42 milliards d'euros en 2023, presque le double des 22 milliards de l'année précédente, selon Statista Market Insights. De 2023 à 2030, le marché devrait croître en moyenne de 24 % par an, atteignant plus de 200 milliards d'euros en 2030. Le nombre d'utilisateurs d'outils d'IA générative devrait passer de 250 millions cette année à plus de 700 millions d'ici la fin de la décennie [47].

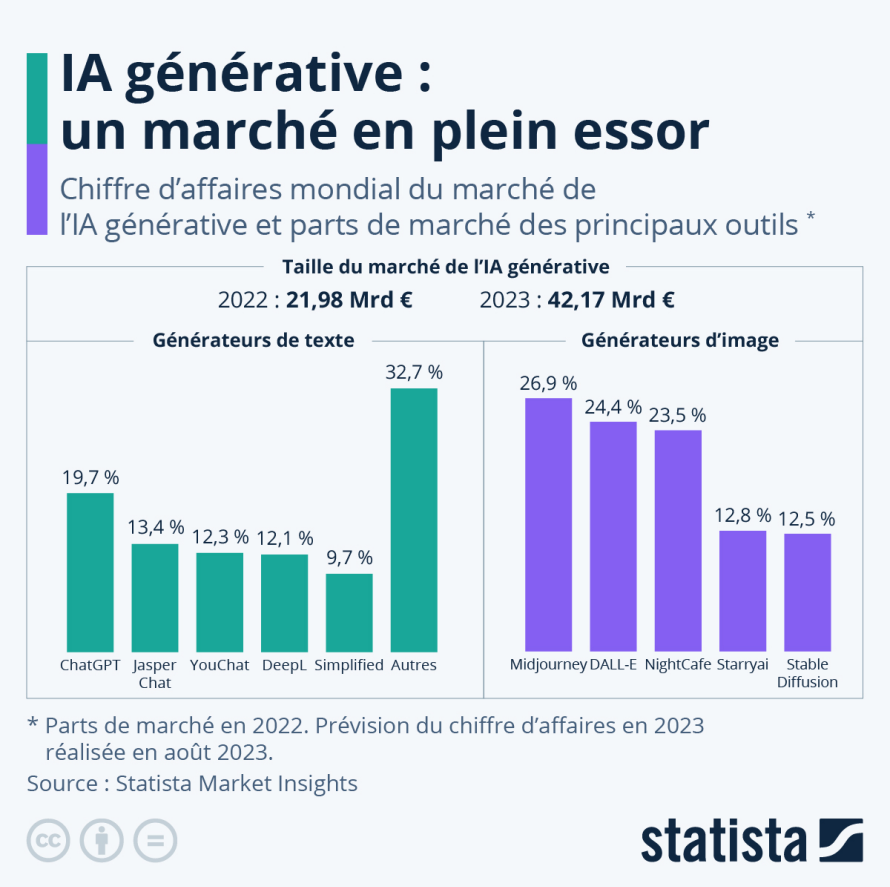


FIG. A.4: Chiffre d'affaires mondial du marché de la Gen AI et parts de marché des principaux outils, source [47]

La concurrence est particulièrement intense dans le segment des générateurs de texte. ChatGPT était en tête l'année dernière avec près de 20 % de parts de marché, suivi de Jasper Chat (13 %), YouChat (12 %), DeepL (12 %) et Simplified (près de 10 %). Le reste du marché est partagé par plusieurs autres acteurs. En ce qui concerne les générateurs d'images, trois outils dominent : Midjourney, DALL-E et NightCafe, chacun ayant environ 25 % de parts de marché l'an dernier [48].

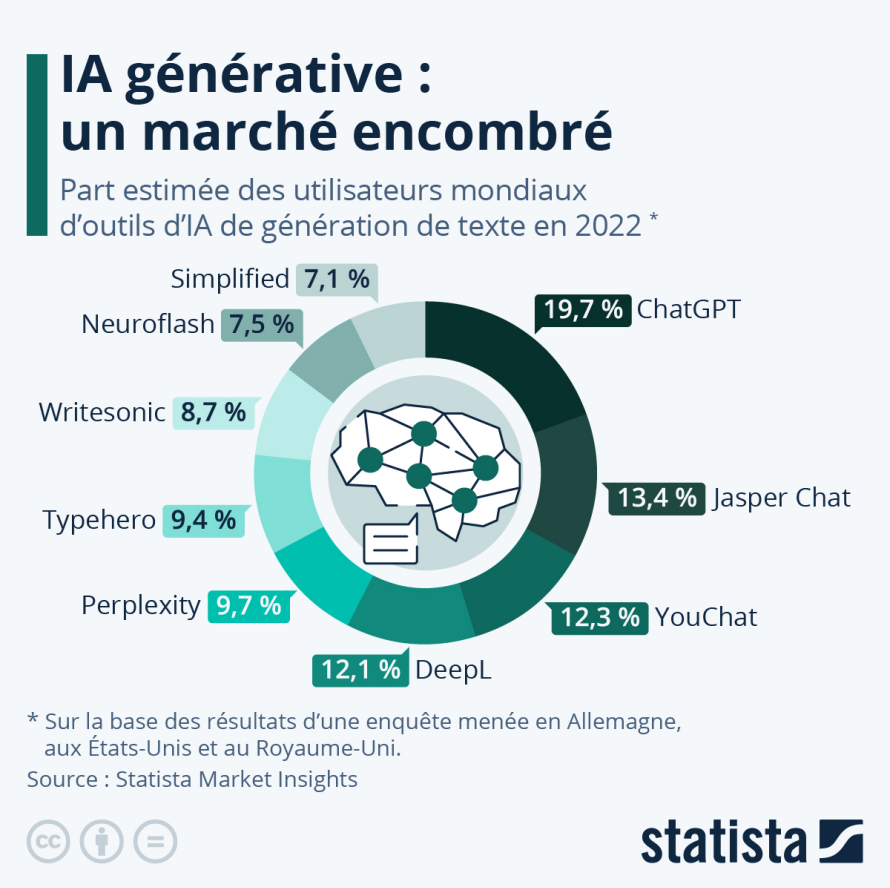


FIG. A.5: Parts estimées des utilisateurs mondiaux d'outils d'IA Générative, source [48]

Investissements en IA

Les investissements des entreprises dans l'intelligence artificielle ont considérablement augmenté au cours de la dernière décennie. Selon une analyse de l'université de Stanford, le total des investissements, y compris les fusions et acquisitions, participations minoritaires, investissements privés et offres publiques, a atteint 934,2 milliards de dollars entre 2013 et 2022 [49].

En 2021, les investissements ont culminé à plus de 276 milliards de dollars. Bien qu'une baisse ait été observée en 2022, la sortie de ChatGPT d'OpenAI en novembre 2022 a renforcé l'idée que l'IA est la prochaine grande technologie disruptive. Désormais, presque tous les grands acteurs technologiques se concentrent sur ce domaine [49].

Ces chiffres proviennent du suivi des investissements de plus de 8 millions d'entreprises publiques et privées dans le monde.

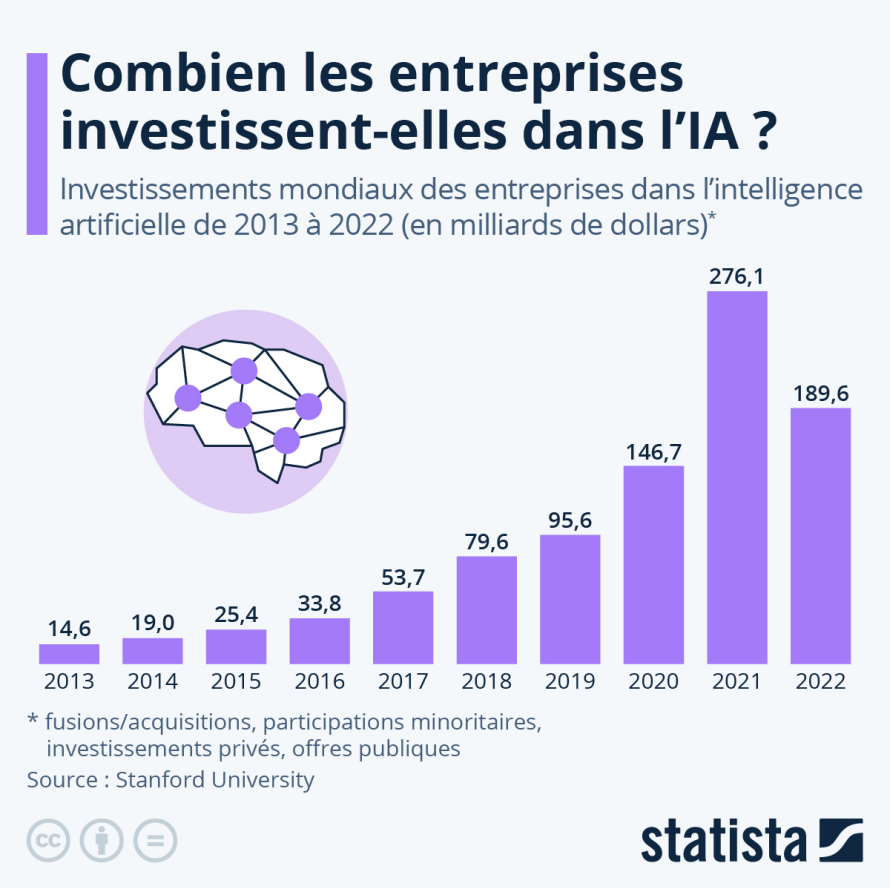


FIG. A.6: Investissements mondiaux des entreprises dans l'IA, source [49]

Annexe B

Stratégies d'implémentation d'une solution Gen AI

L'engouement autour de l'intelligence artificielle générative est palpable, et les cadres dirigeants souhaitent légitimement avancer avec une rapidité réfléchie et intentionnelle. Implémenter une solution GenAI demande une approche rigoureuse basée sur le AI Risk management. C'est pour cela que le cabinet de conseil Mckinsey & Company a défini une démarche comportant 7 étapes stratégiques à suivre afin d'aider les PDGs d'entreprises à tirer profit de l'IA Générative de manière efficace, comme indiqué dans la figure B.1 ci-contre [45]. Nous allons de ce fait expliquer puis appliquer cette méthodologie afin de pouvoir proposer notre solution.

Fig. B.1: Stratégies d'implémentaion de solution Gen AI, source [45]

Organisation pour l'intelligence artificielle générative

L'IA générative nécessite une approche coordonnée en raison de ses risques uniques et de son potentiel polyvalent. Selon le cabinet, il serait recommandé de former un groupe interfonctionnel de leaders de l'entreprise, incluant par exemple des experts en science des données, ingénierie, juridique, cybersécurité, marketing, design et d'autres fonctions commerciales. Ce groupe peut non seulement aider à identifier et à prioriser les cas d'utilisation à plus forte valeur ajoutée, mais aussi permettre une mise en œuvre coordonnée et sécurisée à travers l'organisation.

Dans notre cas, à travers différentes réunions avec les Partners et directeurs et autres responsables, le cas d'utilisation qui a été retenu et priorisé est celui de la création d'un assistant permettant la lecture et l'analyse de rapports financiers

faisant partie du domaine Corporate Real Estate, et plus particulièrement celui incluant les grands groupes hôteliers.

Réimaginer les domaines de bout en bout versus se concentrer sur les cas d'utilisation

L'IA générative est un outil puissant capable de transformer le fonctionnement des organisations, notamment dans certains domaines spécifiques de la chaîne de valeur (par exemple, le marketing pour un détaillant ou les opérations pour un fabricant). Selon le cabinet, il serait crucial d'adopter une perspective sur les familles de cas d'utilisation par domaine ayant le potentiel le plus transformateur à travers les fonctions commerciales. Les organisations réimaginent l'état cible rendu possible par l'IA générative, en synergie avec les applications d'IA traditionnelles et de nouvelles méthodes de travail auparavant impossibles.

Mise en place d'une infrastructure technologique complète

Une infrastructure technologique moderne est essentielle pour l'IA générative. Les PDG doivent consulter leurs directeurs techniques pour évaluer les capacités techniques de l'entreprise : ressources informatiques, systèmes de données, outils et accès aux modèles.

L'accès fluide aux données spécifiques est crucial. Les entreprises doivent harmoniser et faciliter l'accès à leurs données pour optimiser l'IA générative. Une architecture de données évolutive avec gouvernance et sécurité est également nécessaire. Selon les besoins, l'infrastructure informatique existante pourrait nécessiter des mises à niveau. Une stratégie claire, centrée sur la valeur commerciale de l'IA générative, est indispensable.

Dans notre cas, nous disposons bel et bien des ressources nécessaires pour le développement de cette solution.

Construire un projet « phare »

Les PDG doivent éviter de rester bloqués à la planification, car les technologies d'IA générative évoluent rapidement. Par exemple, GPT-4 a été lancé peu après GPT-3.5 et GPT-3. Il est crucial de progresser rapidement pour profiter de ces

innovations.

Pour montrer l'impact de l'IA générative, une approche de « phare » peut être utilisée. Par exemple, créer un « expert virtuel » pour aider les employés à accéder à des connaissances spécifiques et offrir un contenu pertinent aux clients, augmentant ainsi la productivité et testant l'IA en interne avant de l'étendre.

Les preuves de concept (POC : Proof of Concept) restent essentielles pour tester et affiner les cas d'utilisation avant de les généraliser. En se concentrant sur des succès initiaux, nous pouvons favoriser l'adoption de l'IA et cultiver une culture d'innovation.

Dans notre situation, nous sommes pour le moment en train de proposer un POC qui permettra par la suite d'affiner la use case mentionnée précédemment.

Équilibrer risques et création de valeur

Les dirigeants doivent équilibrer les opportunités de valeur avec les risques de l'IA générative. Il est essentiel d'établir des principes éthiques et de comprendre les risques spécifiques à chaque cas d'utilisation. Les entreprises doivent rester informées des réglementations en évolution.

Approche écosystémique des partenariats

Les entreprises doivent développer un écosystème de partenaires adapté à l'IA générative pour accélérer les progrès. Cela implique de collaborer avec des fournisseurs d'IA générative et d'infrastructure pour des solutions évolutives.

Concentration sur les talents et les compétences nécessaires

Les entreprises doivent développer leurs capacités internes et former leurs employés actuels pour tirer parti de l'IA générative. Cela nécessite une variété de compétences techniques et une compréhension approfondie des cas d'utilisation.

Annexe C

Discussion

Nous allons dans cette partie essayer de répondre à une question toute aussi importante et stratégique avant toute implémentation [50].

Faut-il Acheter ou plutôt Développer une solution GenAI ?

En interne, les data scientists peuvent contribuer à créer un propre modèle d'IA, mais passer en production et implémenter le modèle nécessite une expertise spécifique. En même temps, respecter un calendrier projeté pour mettre en œuvre un projet d'IA générative est tout aussi important. De ce fait, la question suivante se pose : sommes-nous prêts à attendre pour construire une IA à partir de zéro ?

Acheter des Solutions d'IA Générative

Avantages :

- Rapidité de Mise sur le Marché : L'achat de solutions d'IA générative permet aux entreprises d'intégrer rapidement des capacités avancées sans le long processus de développement. Cela est particulièrement avantageux dans les marchés en évolution rapide où le délai de mise sur le marché est crucial.
- Coûts Initiaux Moins Élevés : L'investissement initial pour acheter une solution d'IA est généralement inférieur à celui de la construire de toutes pièces. Les fournisseurs répartissent les coûts de développement sur plusieurs clients, ce qui entraîne des économies d'échelle.
- Risque Réduit : Les fournisseurs établis offrent souvent des solutions éprouvées et fiables. Cela réduit les risques techniques et opérationnels associés au

déploiement de nouvelles technologies.

- Support et Mises à Jour Continus : L'achat d'une solution d'IA inclut généralement l'accès au support du fournisseur et aux mises à jour régulières, garantissant que la technologie reste actuelle et efficace.

Inconvénients :

- Personnalisation Limitée : Les solutions achetées peuvent ne pas correspondre parfaitement aux besoins spécifiques d'une entreprise. Les options de personnalisation sont souvent limitées, ce qui peut entraîner des compromis sur la fonctionnalité ou l'intégration avec les systèmes existants.
- Dépendance envers le Fournisseur : S'appuyer sur un fournisseur externe peut entraîner une dépendance vis-à-vis du fournisseur pour les mises à jour, le support et les innovations futures.
- Sécurité et Conformité des Données : L'utilisation de solutions d'IA tierces soulève des préoccupations concernant la sécurité des données et la conformité, en particulier si des données sensibles ou propriétaires sont impliquées. Il est crucial de s'assurer que le fournisseur respecte les réglementations pertinentes.

Développer des Solutions d'IA Générative

Avantages :

- Personnalisation : Construire une solution d'IA générative en interne permet une personnalisation complète pour répondre aux besoins spécifiques de l'entreprise et s'intégrer parfaitement avec les systèmes existants. Cela garantit que la solution correspond précisément aux processus et objectifs de l'entreprise.
- Contrôle : Développer une solution d'IA en interne offre un plus grand contrôle sur ses fonctionnalités, mises à jour et modifications. Cette autonomie peut être cruciale pour maintenir des avantages concurrentiels et s'adapter rapidement aux conditions changeantes du marché.
- Sécurité des Données : Avec une solution interne, les entreprises conservent un contrôle total sur leurs données, réduisant le risque de violations de données et garantissant la conformité aux réglementations et normes spécifiques à l'industrie.

- Transparence et Portabilité : Les modèles ouverts, tels que Mistral, Falcon ou Llama 2, offrent une plus grande transparence par rapport aux modèles fermés comme ceux d'OpenAI ou Gemini. Cette transparence permet une meilleure compréhension et une meilleure gestion des biais et des comportements cachés des modèles.

- Adaptabilité : Les modèles ouverts peuvent être ajustés et affinés pour correspondre aux besoins spécifiques de l'entreprise, permettant une personnalisation poussée pour des domaines ou contextes locaux.

Inconvénients :

- Coûts Initiaux Élevés : Développer une solution d'IA générative à partir de zéro nécessite un investissement significatif en technologie, talents et infrastructure. Ces coûts initiaux peuvent être prohibitifs, en particulier pour les petites organisations.

- Temps de Développement Plus Long : Construire une solution d'IA en interne est un processus chronophage, ce qui peut retarder le déploiement de capacités critiques. C'est un inconvénient majeur dans les industries à rythme rapide.

- Expertise Technique : Développer des technologies d'IA sophistiquées nécessite des compétences et des connaissances spécialisées. Les entreprises doivent évaluer si elles possèdent ou peuvent acquérir l'expertise nécessaire. Un manque de capacités internes peut conduire à des solutions sous-optimales ou à des échecs de projet.

- Maintenance et Mises à Jour : Une fois développée, la solution nécessite une maintenance, des mises à jour et un support continus, ce qui peut peser sur les ressources internes. La responsabilité de maintenir l'IA à jour et efficace incombe uniquement à l'entreprise.

Modèles Fermés vs. Modèles Ouverts

Les modèles fermés, comme ceux développés par OpenAI ou Gemini, sont souvent perçus comme des "boîtes noires" en raison de leur manque de transparence. En revanche, les modèles ouverts, tels que Mistral, Falcon et Llama 2, offrent une visibilité accrue sur les mécanismes internes, permettant une meilleure compréhension et gestion des biais potentiels et des comportements cachés.

L'analyse des risques est essentielle lors de la mise en œuvre de solutions d'IA. Les biais, les comportements cachés, les données de formation utilisées et la transparence sont des facteurs cruciaux à considérer pour protéger les données de l'entreprise et assurer une utilisation éthique et conforme de l'IA.

Le tableau récapitulatif C.1 comparent les deux approches :

Aspect	**Acheter**	**Développer**
Rapidité de Mise sur le Marché	Rapide	Lent
Coûts Initiaux	Moins élevés	Élevés
Risque Technique et Opérationnel	Réduit grâce aux solutions éprouvées	Plus élevé, nécessite une gestion interne
Personnalisation	Limitée	Complète
Contrôle	Moins de contrôle, dépendance vis-à-vis du fournisseur	Contrôle total
Support et Mises à Jour	Fournis par le vendeur	Responsabilité interne
Sécurité et Conformité des Données	Risques potentiels de sécurité et conformité	Contrôle total sur la sécurité et la conformité
Transparence	Moins transparent (Modèles fermés comme OpenAI,	Plus transparent (Modèles ouverts comme Mistral,

Tab. C.1: Comparaison entre l'achat et le développement de solutions Gen AI

En résumé,

- Si l'objectif est de progresser et de consolider un avantage concurrentiel avec une équipe solide en science des données, **le développement** est conseillée.
- Si l'objectif est de rechercher une solution spécialisée avec des capacités et des délais limités, **l'achat** est recommandé.

Annexe D

Code source : Scripts Python

1. Local_loader.py :

```
import os
from pathlib import Path

from pypdf import PdfReader
from langchain.docstore.document import Document
from langchain_community.document_loaders import TextLoader,
↪  UnstructuredPDFLoader, OnlinePDFLoader, PyPDFLoader, TextLoader
from langchain_community.document_loaders.csv_loader import
↪  CSVLoader


def list_txt_files(data_dir="./data"):
    paths = Path(data_dir).glob('**/*.txt')
    for path in paths:
        yield str(path)


def load_txt_files(data_dir="./data"):
    docs = []
    paths = list_txt_files(data_dir)
    for path in paths:
        print(f"Loading {path}")
        loader = TextLoader(path)
        docs.extend(loader.load())
    return docs
```

```
def load_csv_files(data_dir="./data"):
    docs = []
    paths = Path(data_dir).glob('**/*.csv')
    for path in paths:
        loader = CSVLoader(file_path=str(path))
        docs.extend(loader.load())
    return docs



def get_document_text(uploaded_file, title=None):
    docs = []
    fname = uploaded_file.name
    if not title:
        title = os.path.basename(fname)
    if fname.lower().endswith('pdf'):
        pdf_reader = PdfReader(uploaded_file)
        for num, page in enumerate(pdf_reader.pages):
            page = page.extract_text()
            doc = Document(page_content=page, metadata={'title':
             ↪  title, 'page': (num + 1)})
            docs.append(doc)

    else:

        doc_text = uploaded_file.read().decode()
        docs.append(doc_text)

    return docs

def load_pdf(file_path):
    loader = PyPDFLoader(file_path=file_path)
    data = loader.load()


    # PyPDFLoader splits by page

    print (f'You have {len(data)} document(s) in your data')
```

```
    print (f'There are {len(data[0].page_content)} characters in
    ↪  the first document')
    print(f'Here is a sample: {data[0].page_content[:200]}')

    return data
```

2. Remote_Loader.py

```
import requests
import os

from langchain_community.document_loaders import WebBaseLoader,
↪  WikipediaLoader
from local_loader import get_document_text
from langchain_community.document_loaders import OnlinePDFLoader


CONTENT_DIR = os.path.dirname(__file__)




def load_web_page(page_url):
    loader = WebBaseLoader(page_url)
    data = loader.load()
    return data


def load_online_pdf(pdf_url):
    loader = OnlinePDFLoader(pdf_url)
    data = loader.load()
    return data


def filename_from_url(url):
    filename = url.split("/")[-1]
    return filename
```

```
def download_file(url, filename=None):
    response = requests.get(url)
    if not filename:
        filename = filename_from_url(url)

    full_path = os.path.join(CONTENT_DIR, filename)

    with open(full_path, mode="wb") as f:
        f.write(response.content)
        download_path = os.path.realpath(f.name)
    print(f"Downloaded file {filename} to {download_path}")
    return download_path


def get_wiki_docs(query, load_max_docs=2):
    wiki_loader = WikipediaLoader(query=query,
    ↪  load_max_docs=load_max_docs)
    docs = wiki_loader.load()
    for d in docs:
        print(d.metadata["title"])
    return docs
```

3. splitter.py

```
# Split documents into chunks
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.docstore.document import Document
from local_loader import load_pdf
import pandas as pd

# This function return a docment(page_content=....) there are no
↪  metadatas.
def split_documents(docs):
    text_splitter = RecursiveCharacterTextSplitter(
        chunk_size=1000, # Hyper parameter
        chunk_overlap=0, # Hyper parameter
        length_function=len,
        is_separator_regex=False)
```

```
    contents = docs
    if docs and isinstance(docs[0], Document):
        contents = [doc.page_content for doc in docs]

    texts = text_splitter.create_documents(contents)
    n_chunks = len(texts)
    print(f"Split into {n_chunks} chunks")
    return texts

def split_documentss(data):
    text_splitter = RecursiveCharacterTextSplitter(
        chunk_size=1000, # Hyper parameter
        chunk_overlap=0, # Hyper parameter
        length_function=len,
        is_separator_regex=False)

    texts = text_splitter.split_documents(data)

    print(f'Now we have {len(texts)} documents')

    texts_list = []
    for i in range(0, len(texts)):
        texts_list.append(texts[i].page_content)
    print('text list length', len(texts_list))

    page_list = []
    for i in range(0, len(texts)):
        page_list.append(texts[i].metadata['page'])
    print('len list des pages', len(page_list))

    dict_text = {'text':texts_list,
             'page':page_list}
    text_pd = pd.DataFrame(dict_text)
    text_pd['id'] = text_pd.index

    return texts,text_pd
```

4. Vector_store.py

```
import logging
import os
from typing import List
import pandas as pd
from langchain_openai import OpenAIEmbeddings
from langchain_community.vectorstores import Chroma
from local_loader import get_document_text, load_pdf
from remote_loader import download_file
from splitter import split_documents, split_documentss
from dotenv import load_dotenv
from time import sleep


EMBED_DELAY = 0.02  # 20 milliseconds



# This is to get the Streamlit app to use less CPU while embedding
↪  documents into Chromadb.
class EmbeddingProxy:
    def __init__(self, embedding):
        self.embedding = embedding

    def embed_documents(self, texts: List[str]) ->
    ↪  List[List[float]]:
        sleep(EMBED_DELAY)
        return self.embedding.embed_documents(texts)

    def embed_query(self, text: str) -> List[float]:
        sleep(EMBED_DELAY)
        return self.embedding.embed_query(text)



# This happens all at once, not ideal for large datasets.
def create_vector_db(texts, embeddings=None,
↪  collection_name="chroma"):
    if not texts:
```

```
        logging.warning("Empty texts passed in to create vector
        ↪  database")
    # Select embeddings
    if not embeddings:
        # To use HuggingFace embeddings instead:


        from langchain_huggingface import HuggingFaceEmbeddings
        embeddings =
        ↪  HuggingFaceEmbeddings(model_name="thenlper/gte-large")


        #openai_api_key = os.environ["OPENAI_API_KEY"]
        #embeddings =
        ↪  OpenAIEmbeddings(openai_api_key=openai_api_key,
        ↪  model="text-embedding-3-small")

    proxy_embeddings = EmbeddingProxy(embeddings)
    # Create a vectorstore from documents
    # this will be a chroma collection with a default name.
    db = Chroma(collection_name=collection_name,
                embedding_function=proxy_embeddings,
                persist_directory=os.path.join("store/",
                ↪  collection_name))
    db.add_documents(texts)

    return db
def find_similar(vs, query):
    docs = vs.similarity_search(query)
    return docs
```

5. Basic_chain.py

```
import os

from langchain_core.output_parsers import StrOutputParser
from langchain_core.prompts import ChatPromptTemplate
from langchain_openai import ChatOpenAI
from langchain_community.llms import HuggingFaceHub
```

```
from langchain_community.chat_models.huggingface import
↪  ChatHuggingFace



from dotenv import load_dotenv


MISTRAL_ID = "mistralai/Mistral-7B-Instruct-v0.1"
ZEPHYR_ID = "HuggingFaceH4/zephyr-7b-beta"
GPT_ID = "gpt-3.5-turbo-0125"

def get_model(repo_id= GPT_ID , **kwargs):
    if repo_id == "gpt-3.5-turbo-0125":
        try:
            chat_model = ChatOpenAI(model="gpt-3.5-turbo-0125",
            ↪  temperature=0, **kwargs)
        except Exception as e:
            print(f"Failed to load OpenAI model: {e}")
            repo_id = ZEPHYR_ID  # Fallback to the Hugging Face
            ↪  model
    if repo_id != "gpt-3.5-turbo-0125":
        huggingfacehub_api_token =
        ↪  kwargs.get("HUGGINGFACEHUB_API_TOKEN", None)
        if not huggingfacehub_api_token:
            huggingfacehub_api_token =
            ↪  os.environ.get("HUGGINGFACEHUB_API_TOKEN", None)
        os.environ["HF_TOKEN"] = huggingfacehub_api_token

        llm = HuggingFaceHub(
            repo_id=repo_id,
            task="text-generation",
            model_kwargs={
                "max_new_tokens": 512,
                "top_k": 30,
                "temperature": 0.1,
                "repetition_penalty": 1.03,
                "huggingfacehub_api_token":
                ↪  huggingfacehub_api_token,
            })
```

```
        chat_model = ChatHuggingFace(llm=llm)
    return chat_model


def basic_chain(model=None, prompt=None):
    if not model:
        model = get_model()
    if not prompt:
        prompt = ChatPromptTemplate.from_template("Tell me the most
        ↪  noteworthy books by the author {author}")

    chain = prompt | model
    return chain
```

6. ensemble.py

```
import os

from langchain_community.retrievers import BM25Retriever
from langchain.retrievers import EnsembleRetriever
from langchain_core.output_parsers import StrOutputParser
from basic_chain import get_model
from rag_chain import make_rag_chain
from remote_loader import load_web_page
from splitter import split_documents
from local_loader import get_document_text
from vector_store import create_vector_db
from dotenv import load_dotenv


def ensemble_retriever_from_docs(docs, embeddings=None):
    texts = split_documents(docs)
    from langchain_community.embeddings import
    ↪  HuggingFaceEmbeddings
    embeddings =
    ↪  HuggingFaceEmbeddings(model_name="thenlper/gte-large")
    vs = create_vector_db(texts, embeddings)
    vs_retriever = vs.as_retriever()
```

```
    bm25_retriever = BM25Retriever.from_texts([t.page_content for t
    ↪  in texts])

    ensemble_retriever = EnsembleRetriever(
        retrievers=[bm25_retriever, vs_retriever],
        weights=[0.5, 0.5])

    return ensemble_retriever
```

7. filter.py.

```
from langchain.retrievers.document_compressors import
↪  DocumentCompressorPipeline
from langchain_community.document_transformers import
↪  EmbeddingsRedundantFilter, LongContextReorder
from langchain_community.embeddings import
↪  HuggingFaceBgeEmbeddings, HuggingFaceEmbeddings
from langchain.retrievers import EnsembleRetriever,
↪  ContextualCompressionRetriever, MergerRetriever
from langchain.chains import RetrievalQA

from basic_chain import get_model
from ensemble import ensemble_retriever_from_docs
from remote_loader import load_web_page
from vector_store import create_vector_db

from dotenv import load_dotenv


def create_retriever(texts):
    dense_embeddings =
    ↪  HuggingFaceEmbeddings(model_name="thenlper/gte-large")
    sparse_embeddings =
    ↪  HuggingFaceBgeEmbeddings(model_name="thenlper/gte-large",
    ↪  encode_kwargs={'normalize_embeddings': False})
    dense_vs = create_vector_db(texts, collection_name="dense",
    ↪  embeddings=dense_embeddings)
    sparse_vs = create_vector_db(texts, collection_name="sparse",
    ↪  embeddings=sparse_embeddings)
```

```
    vector_stores = [dense_vs, sparse_vs]

    emb_filter =
    ↪  EmbeddingsRedundantFilter(embeddings=sparse_embeddings)
    reordering = LongContextReorder()
    pipeline = DocumentCompressorPipeline(transformers=[emb_filter,
    ↪  reordering])

    base_retrievers = [vs.as_retriever() for vs in vector_stores]
    lotr = MergerRetriever(retrievers=base_retrievers)

    compression_retriever_reordered =
    ↪  ContextualCompressionRetriever(base_compressor=pipeline,
    ↪  base_retriever=lotr, search_kwargs={"k": 5,
    ↪  "include_metadata": True}
    )
    return compression_retriever_reordered
```

8. rag__chain.py

```
import os

from dotenv import load_dotenv
from langchain import hub
from langchain_core.output_parsers import StrOutputParser
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.runnables import RunnablePassthrough,
↪  RunnableLambda
from langchain_core.messages.base import BaseMessage

from basic_chain import basic_chain, get_model
from remote_loader import get_wiki_docs
from splitter import split_documents
from vector_store import create_vector_db


def find_similar(vs, query):
    docs = vs.similarity_search(query)
    return docs
```

```python
def format_docs(docs):
    return "\n\n".join(doc.page_content for doc in docs)


def get_question(input):
    if not input:
        return None
    elif isinstance(input,str):
        return input
    elif isinstance(input,dict) and 'question' in input:
        return input['question']
    elif isinstance(input,BaseMessage):
        return input.content
    else:
        raise Exception("string or dict with 'question' key
        ↪  expected as RAG chain input.")


def make_rag_chain(model, retriever, rag_prompt = None):

    if not rag_prompt:
        rag_prompt = hub.pull("rlm/rag-prompt")


    rag_chain = (
            {
                "context": RunnableLambda(get_question) | retriever
                ↪  | format_docs,
                "question": RunnablePassthrough()
            }
            | rag_prompt
            | model
    )

    return rag_chain
```

9. memory.py

```
import os
from typing import List, Iterable, Any

from dotenv import load_dotenv
from langchain.memory import ChatMessageHistory
from langchain_core.callbacks import CallbackManagerForRetrieverRun
from langchain_core.chat_history import BaseChatMessageHistory
from langchain_core.documents import Document
from langchain_core.output_parsers import StrOutputParser
from langchain_core.prompts import ChatPromptTemplate,
↪  MessagesPlaceholder
from langchain_core.retrievers import BaseRetriever
from langchain_core.runnables.history import
↪  RunnableWithMessageHistory

from basic_chain import get_model
from rag_chain import make_rag_chain


def create_memory_chain(llm, base_chain, chat_memory):
    contextualize_q_system_prompt = """Given a chat history and the
    ↪  latest user question \
        which might reference context in the chat history,
        ↪  formulate a standalone question \
        which can be understood without the chat history. Do NOT
        ↪  answer the question, \
        just reformulate it if needed and otherwise return it as
        ↪  is."""

    contextualize_q_prompt = ChatPromptTemplate.from_messages(
        [
            ("system", contextualize_q_system_prompt),
            MessagesPlaceholder(variable_name="chat_history"),
            ("human", "{question}"),
        ]
    )

    runnable = contextualize_q_prompt | llm | base_chain
```

```
    def get_session_history(session_id: str) ->
    ↪  BaseChatMessageHistory:
        return chat_memory

    with_message_history = RunnableWithMessageHistory(
        runnable,
        get_session_history,
        input_messages_key="question",
        history_messages_key="chat_history",
    )
    return with_message_history


class SimpleTextRetriever(BaseRetriever):
    docs: List[Document]
    """Documents."""

    @classmethod
    def from_texts(
            cls,
            texts: Iterable[str],
            **kwargs: Any,
    ):
        docs = [Document(page_content=t) for t in texts]
        return cls(docs=docs, **kwargs)

    def _get_relevant_documents(
            self, query: str, *, run_manager:
            ↪  CallbackManagerForRetrieverRun
    ) -> List[Document]:
        return self.docs
```

10. full_chain.py

```
import os

from dotenv import load_dotenv
from langchain.memory import ChatMessageHistory
from langchain_core.prompts import ChatPromptTemplate
```

```
from basic_chain import get_model
from filter import ensemble_retriever_from_docs
from local_loader import load_txt_files
from memory import create_memory_chain
from rag_chain import make_rag_chain

GPT_ID = "gpt-3.5-turbo-0125"
def create_full_chain(retriever, openai_api_key=None,
↪  chat_memory=ChatMessageHistory()):
    model = get_model(repo_id= GPT_ID,
    ↪  openai_api_key=openai_api_key)
    system_prompt = """You are a helpful AI assistant for people
    ↪  busy to read an article, and they need information from
    ↪  these articles.
    Use the following context and the user's chat history to help
    ↪  the user:
    If you don't know the answer, just say that you don't know.
    Respond with the user's language (either French or English).

    Context: {context}

    Question: """

    prompt = ChatPromptTemplate.from_messages(
        [
            ("system", system_prompt),
            ("human", "{question}"),
        ]
    )

    rag_chain = make_rag_chain(model, retriever, rag_prompt=prompt)
    chain = create_memory_chain(model, rag_chain, chat_memory)
    return chain


def ask_question(chain, query):
    response = chain.invoke(
        {"question": query},
        config={"configurable": {"session_id": "foo"}}
```

```
    )
    return response
```

11. streamlit_app.py

```
import streamlit as st
from langchain_community.chat_message_histories import
↪  StreamlitChatMessageHistory
from langchain_community.embeddings import OpenAIEmbeddings

from ensemble import ensemble_retriever_from_docs
from full_chain import create_full_chain, ask_question
from local_loader import load_txt_files, get_document_text
from langchain_community.embeddings import HuggingFaceEmbeddings
import os



st.image("KPMG.png", width=250)


st.markdown(
    """
    <style>
    /* Customize the main background color */
    .main {
        background-color: #E6E6E6; /* light gray background */
    }

    /* Customize the sidebar background color */
    .css-1d391kg {
        background-color: #00338D; /* primary blue sidebar
        ↪  background */
    }

    /* Customize the font color in the sidebar */
    .css-1d391kg, .css-1d391kg * {
        color: white; /* white font color */
```

```
        }

        /* Customize the header */
        .css-10trblm {
            color: #005EB8; /* secondary blue header */
        }

        /* Customize button */
        .stButton button {
            background-color: #0072C6; /* accent blue background */
            color: white;
            border-radius: 10px;
            border: 2px solid #0072C6;
            padding: 10px 24px;
        }

        /* Customize text input */
        .stTextInput input {
            border: 2px solid #0072C6; /* accent blue border */
            padding: 10px;
            border-radius: 5px;
        }
        </style>
        """,
        unsafe_allow_html=True
)
st.title("DealAI")
st.header("Bienvenue sur la plateforme de l'Assistant Deal
↪  Advisory")


def show_ui(qa, prompt_to_user="Comment puis-je vous aider?"):

    if "messages" not in st.session_state.keys():
        st.session_state.messages = [{"role": "assistant",
        ↪  "content": prompt_to_user}]

    # Display chat messages
    for message in st.session_state.messages:
        with st.chat_message(message["role"]):
```

```
            st.write(message["content"])

    # User-provided prompt
    if prompt := st.chat_input():
        st.session_state.messages.append({"role": "user",
        ↪  "content": prompt})
        with st.chat_message("user"):
            st.write(prompt)

    # Generate a new response if last message is not from
    ↪  assistant
    if st.session_state.messages[-1]["role"] != "assistant":
        with st.chat_message("assistant"):
            with st.spinner("Patientez..."):
                response = ask_question(qa, prompt)
                st.markdown(response.content)
        message = {"role": "assistant", "content":
        ↪  response.content}
        st.session_state.messages.append(message)



@st.cache_resource

def get_retriever(openai_api_key=None, pdf_paths=None):
    #docs = load_txt_files()
    example_pdf_path = pdf_paths
    docs = get_document_text(open(example_pdf_path, "rb"))
    embeddings =
    ↪  HuggingFaceEmbeddings(model_name="all-MiniLM-L6-v2")
    return ensemble_retriever_from_docs(docs,
    ↪  embeddings=embeddings)




def get_chain(openai_api_key=None, huggingfacehub_api_token=None,
↪  pdf_pathg=None):
    pdf_pathg = pdf_pathg
```

```
    ensemble_retriever =
    ↪  get_retriever(openai_api_key=openai_api_key,
    ↪  pdf_paths=pdf_pathg)
    chain = create_full_chain(ensemble_retriever,
    ↪  openai_api_key=openai_api_key,
    ↪  chat_memory=StreamlitChatMessageHistory(key="langchain_messages"))
    return chain


def get_secret_or_input(secret_key, secret_name, info_link=None):
    if secret_key in st.secrets:
        st.write("Found %s secret" % secret_key)
        secret_value = st.secrets[secret_key]
    else:
        st.write(f"Please provide your {secret_name}")
        secret_value = st.text_input(secret_name,
        ↪  key=f"input_{secret_key}", type="password")
        if secret_value:
            st.session_state[secret_key] = secret_value
        if info_link:
            st.markdown(f"[Get an {secret_name}]({info_link})")
    return secret_value

def select_pdf():
    pdf = st.file_uploader(label='Téléchargez votre rapport/article
    ↪  PDF ici', type='pdf')

    if pdf is not None:
        pdf_path = f"examples/{pdf.name}"

        if os.path.exists(pdf_path):
            st.warning(f'The file "{pdf.name}" already exists in
            ↪  the examples folder.')
        else:
            # Save the uploaded file to the examples directory
            with open(pdf_path, 'wb') as f:
                f.write(pdf.read())
            st.success(f'File successfully saved to {pdf_path}')

        st.write(f'The file is saved in the path: {pdf_path}')
```

```
        return pdf_path


def run():


    ready = True

    openai_api_key = st.session_state.get("OPENAI_API_KEY")
    huggingfacehub_api_token =
     ↪  st.session_state.get("HUGGINGFACEHUB_API_TOKEN")

    with st.sidebar:
        pdf_path = select_pdf()
        if not openai_api_key:
            openai_api_key = get_secret_or_input('OPENAI_API_KEY',
             ↪  "OpenAI API
             ↪  key",info_link="https://platform.openai.com/account/api-keys")
        if not huggingfacehub_api_token:
            huggingfacehub_api_token =
             ↪  get_secret_or_input('HUGGINGFACEHUB_API_TOKEN',
             ↪  "HuggingFace Hub API Token",
             ↪  info_link="https://huggingface.co/docs/huggingface_hub"
            "/main/en/quick-start#authentication")

    if not openai_api_key:
        st.warning("Missing OPENAI_API_KEY")
        ready = False
    if not huggingfacehub_api_token:
        st.warning("Missing HUGGINGFACEHUB_API_TOKEN")
        ready = False


    if ready:

        chain = get_chain(openai_api_key=openai_api_key,
         ↪  huggingfacehub_api_token=huggingfacehub_api_token,
         ↪  pdf_pathg=pdf_path)
        #st.subheader("Welcome to the KPMG Chatbot!")
```

```
        show_ui(chain, "Bonjour! Je suis DealAI l'assistant IA pour
        ↪  le Deal Advisory  je vous offre toutes les capacités
        ↪  d'un assistant IA comme ChatGPT et je vous aide à lire
        ↪  et analyser les rapports financiers et annuels des
        ↪  entreprises.")
    else:
        st.stop()


run()
```