

RÉPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



Département du Génie Industriel

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

Développement d'un chatbot bancaire intelligent : Comparaison de l'application de la DPO et du RAG pour l'amélioration de l'interaction client chez KPMG

Application : Département Data/IT de KPMG

Réalisé par :

Samah BELBAKI
Sihem BOUTEBAL

Encadré par :

Mme. Samia BELDJOUDI
Mr. Islam SADAT

Présenté et soutenu publiquement le (03/07/2024)

Composition du Jury

Président	Mr. Ali BOUKABOUS	MAA	ENP
Examineur	Mr. Oussama ARKI	MCA	ENP
Promoteur	Mme Samia BELDJOUDI	MCA	ENP
Promoteur	Mr. Islam SADAT	Assistant Manager	KPMG

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

École Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



Département du Génie Industriel

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

Développement d'un chatbot bancaire intelligent : Comparaison de l'application de la DPO et du RAG pour l'amélioration de l'interaction client chez KPMG

Application : Département Data/IT de KPMG

Réalisé par :

Samah BELBAKI
Sihem BOUTEBAL

Encadré par :

Mme. Samia BELDJOUDI
Mr. Islam SADAT

Présenté et soutenu publiquement le (03/07/2024)

Composition du Jury

Président	Mr. Ali BOUKABOUS	MAA	ENP
Examineur	Mr. Oussama ARKI	MCA	ENP
Promoteur	Mme Samia BELDJOUDI	MCA	ENP
Promoteur	Mr. Islam SADAT	Assistant Manager	KPMG

Remerciements

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

Nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à tous ceux qui ont contribué à la réalisation de ce mémoire. Votre soutien inestimable, vos conseils précieux et votre encouragement constant ont été essentiels pour mener à bien ce projet.

Tout d'abord, nous souhaitons remercier notre promotrice, Mme Beldjoudi, pour sa guidance experte et ses précieux conseils tout au long de notre travail. Votre expertise et votre implication ont été d'une importance capitale pour nous guider dans notre réflexion et nous aider à mener à bien nos recherches.

Nous adressons également nos remerciements à notre promoteur, M. Sadat, pour son soutien et sa contribution significative à notre mémoire. Votre encadrement, votre suivi régulier, et vos critiques constructives à chaque étape ont été des facteurs clés qui nous ont permis de mener ce travail à terme. Votre expérience dans le domaine a enrichi notre travail et lui a donné une dimension concrète et appliquée.

Nous tenons également à remercier M. Cherfaoui Aziz, partenaire de l'équipe DATA/IT de KPMG, pour son soutien et sa confiance en nous. Les réunions hebdomadaires avec lui ont été des points de repère essentiels, et ses critiques constructives nous ont constamment guidés vers les meilleures décisions, nous incitant toujours à rechercher l'excellence dans notre travail.

Nous tenons également à remercier notre collègue, M. Zelmat, pour son soutien tout au long du stage. Ses conseils, ses encouragements, et son encadrement ont été cruciaux à la réalisation de ce travail. Sans eux, nous n'aurions jamais pu atteindre un tel degré d'excellence. Son expertise dans le domaine a été pour nous une référence constante.

Nous tenons à exprimer notre reconnaissance envers l'ensemble de l'équipe pédagogique du département du Génie Industriel, et en particulier Messieurs BOUKABOUS, ARKI, BOUBAKEUR, FOURAR et ZOUAGHI, ainsi que Madame BOUCHAFAA et . Vos enseignements, vos conseils et votre encadrement ont été essentiels dans notre parcours académique et ont joué un rôle déterminant dans la réalisation de ce mémoire.

Nous souhaitons également remercier chaleureusement l'équipe DATA/IT de KPMG, pour avoir constamment été à notre écoute, nous avoir généreusement offert leur aide et partagé leurs connaissances. Votre collaboration et votre soutien ont été d'une valeur inestimable et ont contribué à l'enrichissement de notre travail.

Merci à tous !

Dédicace Sihem

Je dédie ce mémoire à toutes les personnes qui ont été présentes à mes côtés et m'ont apporté leur soutien tout au long de ce parcours académique et professionnel, transformant chaque défi en une opportunité de grandir et d'apprendre.

À ma mère, ta présence bienveillante et ton énergie positive m'ont aidée à sortir de la négativité qui m'envahissait. Tes encouragements constants, surtout lorsque mes pensées négatives étaient plus fortes que moi, m'ont donné la force de persévérer. Tes conseils avisés et tes innombrables sacrifices pour moi sont des trésors que je ne pourrai jamais assez te remercier. Merci pour tout ce que tu m'as donné, même dans les moments les plus difficiles. Je sais que le chemin a été épuisant, mais j'espère que tu es fière de moi et que tes efforts ont enfin porté leurs fruits.

À mon père, l'homme numéro un de ma vie, la personne qui a cru en moi plus que quiconque. À celui qui m'a appris la transparence, l'authenticité, et ce que signifie être un vrai homme. Je te remercie infiniment d'avoir toujours élevé la barre et de m'avoir constamment poussée vers l'excellence. Sans toi, je n'aurais jamais eu cette confiance en moi ni parcouru un tel chemin. Merci d'avoir été là pour moi pendant ces 20 dernières années. Que Dieu bénisse ton âme et qu'elle repose en paix. Même si tu n'es pas là pour me voir, je te serai toujours reconnaissante pour tout ce que tu m'as donné. Merci infiniment, belle âme. Merci infiniment, l'homme de ma vie.

A ma grande sœur Nachoua, tes encouragements et ta confiance en moi m'ont toujours donné un surcroît de confiance. Ton énergie positive et tes compliments ont toujours réussi à me sortir de la négativité. Merci infiniment.

À ma petite sœur Imène, ta sérénité et ta confiance dans mon intelligence ont constamment renforcé ma propre confiance. Ton aide constante, surtout quand j'étais fatiguée du travail, et les repas que tu me préparais étaient les gestes d'amour dont j'avais besoin. Merci beaucoup.

Merci à vous deux, et que Dieu vous accorde le meilleur de cette vie.

À mes amies de lycée, Imène et Dounia, Ce fut un immense plaisir de vous connaître. Merci pour votre soutien moral et votre amitié sincère. Votre présence à mes côtés a été une source inestimable de motivation et de réconfort.

À mes amies de Polytech, Khawla, Amira, Samah, Yousra, et Boutheina,

Khawla, la première personne que j'ai rencontrée à l'école, celle avec le cœur le plus pur et naïf que j'ai jamais connu. Je n'oublierai jamais ton aide dans les moments les plus difficiles de ma vie. Si tu n'avais pas été à mes côtés, je n'aurais jamais été capable de terminer ces années. Merci d'avoir été là à chaque fois que j'avais besoin de toi.

Amira, ta personnalité courageuse et intrépide m'a toujours fait croire que je pouvais accomplir tout ce que je voulais et que rien n'était hors de ma portée ou impossible. Merci d'avoir toujours été à mes côtés et de m'avoir donné les conseils nécessaires.

Samah, ta sympathie et ta gentillesse font de toi la RH du groupe, celle vers qui nous nous tournons tous quand ça ne va pas. Tu es toujours prête à écouter et à aider. Ton âme pure et ton esprit amical font de toi une personne avec qui je me sens très à l'aise et en paix. Merci pour tous tes conseils, tes encouragements constants et d'avoir été là à chaque fois que j'avais besoin de toi.

Yousra, la maman du groupe, sans toi, nous serions perdus. Tu es notre guide, la personne qui donne le plus sans jamais rien attendre en retour. Tu penses toujours à nous et n'hésites jamais à offrir ton aide.

Ton existence dans ma vie est une bénédiction. Merci énormément.

Boutheina, la personne la plus réaliste et directe du groupe, mais aussi celle qui nous ouvre toujours les yeux sur la réalité et nous guide pour être fortes et indépendantes. Tu nous donnes la confiance et nous rappelles toujours "ch7al haylat hna". Merci infiniment pour tes encouragements.

Au Meilleur comité de tous les temps

Avoir eu la chance de vous connaître et de partager cette expérience avec vous a été un immense plaisir. votre amitié sincère et nos sorties inoubliables resteront à jamais gravés dans mon cœur. Merci pour chaque main tendue, chaque éclat de rire et chaque instant de bonheur partagé. Je vous souhaite, du fond du cœur, une réussite éclatante et la force de réaliser tous vos rêves.

À toute la communauté du génie industriel et à tous les membres du club IEC, Un immense merci pour toutes les expériences inoubliables que vous m'avez offertes. Je suis profondément reconnaissante pour chaque instant partagé avec vous. Merci de tout cœur !

À mes professeurs et mentors, pour leur guidance, leurs enseignements et leur patience. Vous avez su éveiller en moi la passion pour l'apprentissage et l'innovation. À tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire, je vous exprime ma profonde gratitude.

Dédicace Samah

C'est avec honneur et fierté que je dédie ce travail aux êtres les plus chers à mes yeux : mes parents.

À toi la femme qui a semé mon chemin d'amour et de sérénité, qui m'a élevé dans un empire de tendresse, qui a toujours visé l'excellence pour moi et qui a tout sacrifié pour que je ne manque en rien, maman je porte chaque sacrifice dans mon dos un poids qui est allégé par ta satisfaction.

Toi père, le pilier de ma vie, le soutien présent incessamment, le repère qui m'oriente dans mon chemin. Ma vie est guidée par tes principes, tes valeurs et tes enseignements qui sont le socle sur lequel je construis mes jours.

Vous rendre fiers est l'essence qui me pousse à gravir les échelons de cette vie jusqu'au sommet. Des simples mots ne pourraient combler la reconnaissance que j'ai pour vous.

À toi ma grande sœur, le phare au milieu de chaque obscurité et défis, reflétant une lumière douce et réconfortante, une lumière qui guide mes pas sur le chemin de l'admiration et de l'inspiration.

À toi mon frère, ma bouée de sauvetage qui me rattrape et me garde dans le droit chemin, l'homme qui m'a appris les secrets et les valeurs de la vie.

À toi ma petite boule de bonheur, ta présence est un cadeau précieux que je chéris soigneusement, ta lumière brille si fort qu'elle me rend fier à chaque instant.

Je remercie Dieu de m'avoir bénie par votre fraternité.

Je ne peux oublier mes amis qui ont embelli chaque pas de ma vie.

À vous : Wissem, ma partenaire de route, je t'ai toujours admiré pour ta force et ton ambition qui m'inspirent. Avec toi, j'ai vécu les meilleurs moments de ma vie, ainsi que les plus dures, et t'avoir à mes côtés rend chaque instant plus magnifique.

Meriem, ma sœur de cœur, dans tes yeux, j'ai toujours perçu la pureté et la tendresse indispensables pour traverser chaque moment de ma vie apportant lumière et chaleur quand j'en ai le plus besoin.

Yousra, captivé par ta tendresse et ton instinct maternel, je t'ai tout de suite considérée comme ma jumelle, ma confidente, celle à qui je peux tout confier et dont je sais qu'elle m'accueillera à bras ouverts.

Sihem, dès le premier moment où je t'ai parlée, j'ai immédiatement ressenti ta bonté et ton authenticité. Un sentiment de sérénité et de sécurité m'envahit chaque fois que je suis à tes côtés. Ta force de caractère et ton esprit résolu me pousse à viser toujours plus haut et à me surpasser.

Amira, la fille la plus courageuse que je connaisse, celle qui ne recule devant rien. À tes côtés, je sais que je peux conquérir toutes les merveilles du monde. Ta détermination sans faille et ton audace me motivent à repousser mes limites et à explorer de nouveaux horizons.

Boutheina, la plus forte et la plus rationnelle, celle qui remet mes idées en place quand j'en ai besoin. Ta clarté de pensée et ton jugement sûr m'ont permis de surmonter les moments difficiles que je redoutais.

Je vous remercie, mes amies, pour tout l'amour et l'amitié pure partagés avec vous.

À IEC, l'école de la vie, celle qui m'a fait grandir, devenir mature et responsable. Celle qui m'a fait découvrir des personnes merveilleuses avec qui j'ai partagé les plus beaux moments.

Les plus inoubliables sont ceux passés avec vous, le meilleur comité. Ces rencontres sont parmi les meilleures de ma vie, et j'en suis profondément reconnaissante. Merci pour chaque sourire, chaque soutien, chaque instant partagé. Vous avez marqué mon parcours d'une empreinte indélébile.

À toute ma famille,

À Tita et Jedou, dont l'âme resteront gravées dans mon cœur,

À toutes personnes proches,

À tous ceux qui ont influencé mon chemin,

Un profond merci venu du cœur.

ملخص

تستكشف هذه المذكرة تطوير روبوت محادثة قائم على الذكاء الاصطناعي التوليدي لتحسين التفاعل مع عملاء KPMG في القطاع المصرفي. من خلال دمج تقنيات متقدمة مثل التعلم المعزز مع ردود الفعل البشرية (RLHF) والتوليد المدعوم بالاسترجاع (RAG)، يهدف هذا الروبوت إلى تقديم إجابات سريعة ودقيقة ومخصصة للاستفسارات المحددة للعملاء. يبرز تقييم الأداء قدرته على تقديم إجابات ذات صلة وشخصية، مما يوضح فعالية هذه التقنيات في تعزيز تفاعل العملاء. يعرض هذا العمل استراتيجيات رئيسية للدمج الناجح للذكاء الاصطناعي المتقدم في تفاعلات العملاء في القطاع المصرفي.

الكلمات المفتاحية : الذكاء الاصطناعي التوليدي، روبوت المحادثة المصرفي، التعلم المعزز مع ردود الفعل البشرية (RLHF)، التوليد المدعوم بالاسترجاع (RAG)، تفاعل العملاء، القطاع المصرفي.

Abstract

This thesis explores the development of a generative AI-based chatbot designed to enhance interactions with KPMG's banking clients. By incorporating advanced techniques such as Reinforcement Learning with Human Feedback (RLHF) and Retrieval-Augmented Generation (RAG), this chatbot aims to provide fast, accurate, and tailored responses to specific client inquiries. Performance evaluation highlights its ability to deliver relevant and personalized answers, underscoring the effectiveness of these technologies in enhancing client engagement. This work demonstrates key strategies for the successful integration of advanced AI into client interactions in the banking sector.

Keywords : generative artificial intelligence, banking chatbot, Reinforcement Learning with Human Feedback (RLHF), Retrieval-Augmented Generation (RAG), client interaction, banking sector.

Résumé

Ce mémoire explore le développement d'un chatbot basé sur l'intelligence artificielle générative pour améliorer l'interaction avec les clients bancaires de KPMG. En intégrant des techniques avancées telles que l'Apprentissage par Renforcement à partir de Feedback Humain (RLHF) et la Génération Augmentée par Récupération (RAG), ce chatbot vise à fournir des réponses rapides, précises et adaptées aux requêtes spécifiques des clients. L'évaluation de sa performance met en évidence sa capacité à répondre de manière pertinente et personnalisée, soulignant l'efficacité de ces technologies dans l'amélioration de l'engagement client. Ce travail illustre les stratégies clés pour une intégration réussie de l'IA avancée dans les interactions clients dans le secteur bancaire.

Mots-Clés : intelligence artificielle générative, chatbot, Apprentissage par Renforcement à partir de Feedback Humain (RLHF), génération augmentée par récupération (RAG), interaction client, secteur bancaire.

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

Introduction générale	16
I État des lieux	18
1 Présentation de l'organisme d'accueil	19
1.1 Introduction	19
1.2 Présentation de KPMG	19
1.2.1 KPMG International	19
1.2.2 KPMG Algérie	19
1.3 Organigramme de KPMG Algérie	20
1.4 Principaux services	20
1.4.1 Fiscal et juridique	20
1.4.2 Audit financier	21
1.4.3 Conseil (Advisory)	21
1.4.4 Conseil en informatique (IT Advisory)	22
1.5 Structure organisationnelle	22
1.6 Structure départementale	23
1.7 Conclusion	24
2 Étude de l'existant	25
2.1 Introduction	25
2.2 Clients de KPMG dans le domaine bancaire	25
2.2.1 Secteur de la Fintech	25
2.2.2 Secteur de Customer & Operations Banque	25
2.2.3 Secteur de la Blockchain & Crypto-actifs	26
2.2.4 Secteur du Advisory Bank	26
2.3 Les appels d'offres des clients de KPMG	26
2.4 Défis principaux de ce système	27
2.5 Revue de la littérature	27
2.5.1 Les chatbots	27
2.5.2 Reinforcement learning with human feedback	28

2.5.3	Direct Preference Optimization	28
2.5.4	Génération augmentée par récupération (RAG)	28
2.5.5	Chatbots dans le domaine bancaire	28
2.6	Énoncé de la problématique	29
2.7	Conclusion	29

II État de l'art 30

3 Fondements de l'intelligence artificielle 31

3.1	Introduction	31
3.2	Définition de l'IA	31
3.3	L'apprentissage profond (Deep Learning)	31
3.3.1	Fonctionnement du Deep Learning	32
3.3.2	Applications du Deep Learning	32
3.4	Le traitement automatique du langage naturel (NLP)	33
3.4.1	Techniques et Méthodes Clés en NLP	33
3.4.2	Application du NLP dans le Développement de Chatbots	34
3.5	Chatbots	34
3.5.1	Types de Chatbots	34
3.5.2	Intelligence Artificielle Générative et Chatbots	35
3.5.3	Avantages et Défis de l'IA Générative dans le développement des chatbot	35
3.6	Conclusion	36

4 Large Language Models (LLMs) 37

4.1	Introduction	37
4.2	Terminologies sur les LLM	37
4.3	Les applications des LLM	38
4.4	Les défis des LLM	38
4.5	Architecture des LLMs :	39
4.5.1	Structure Générale des LLMs	39
4.5.2	Composants de Base des LLMs	40
4.6	Amélioration des LLM (Large Language Models)	41
4.6.1	L'ingénierie de prompts (Prompt Engineering)	41
4.6.2	Ajustement fin (fine-tuning)	44
4.6.3	Optimisation de Préférence Direct (DPO)	46
4.6.4	Retrieval-Augmented Generation (RAG)	47
4.7	Évaluation des LLMs	50
4.7.1	Types d'Évaluations	50
4.7.2	Métriques d'évaluation automatique	51
4.8	Conclusion	52

III Conception de la solution 53

5 Présentation de la solution 54

5.1	Introduction	54
5.2	Structure de la solution	54
5.3	Conception de la solution	55
5.3.1	Collecte de données	55
5.3.2	Prompt Engineering	56

5.3.3	Direct Preference Optimization (DPO)	67
5.3.4	Retreival Augmented Generation RAG	74
5.3.5	Comparaison entre les deux méthodes utilisées :	85
5.4	Déploiement de la solution	85
5.4.1	Interface du chatbot	85
5.4.2	Tests de l'interface	86
5.5	Conclusion	88
	Conclusion générale	89
	Perspectives	90
	Références	92
	Annexes	97
	A Bibliothèques utilisées	97
	B Notions avancés sur l'Apprentissage par Renforcement à partir de Feedback Humain	103

Liste des tableaux

4.1	Tableau comparant les caractéristiques entre les modèles RAG et les modèles ajustés finement [32]	50
5.1	Résultats obtenus après chaque epoch d’entraînement	70
5.2	Résultats des scores ROUGE pour chaque question testée	73
5.3	Résultats du ROUGE Score pour le modèle Mixtral 8x7b sur huit questions de la BDD	77
5.4	Résultats du ROUGE Score pour le modèle Llama3 8b sur huit questions de la BDD.	79
5.5	Résultats du ROUGE Score pour le modèle Llama3 70b sur huit questions de la BDD.	80
5.6	Résultats du ROUGE Score pour le modèle Mistral 7b Instruct sur huit questions de la BDD.	82
5.7	Scores d’hallucination pour les différents modèles	84

Table des figures

Deuxième partie	19
1.1 Organigramme de KPMG Algérie	20
1.2 L'hierarchie des grades chez KPMG Algérie	23
1.3 Structuration hiérarchique de KPMG Algérie	23
1.4 Structure départementale de KPMG Algérie	24
Première partie	31
3.1 Réseau de neurones artificiels [6]	32
4.1 L'architecture du modèle transformateur [41]	39
4.2 Les modèles encodeur décodeur [61]	40
4.3 Les couches de l'architecture transformer [62]	41
4.4 Illustration d'un exemple du Few-shot [51]	43
4.5 Illustration d'un exemple du One-Shot [51]	43
4.6 Illustration d'un exemple du Zero-shot [51]	44
4.7 Ajustement fin du modèle de langage principal en utilisant la DPO [58]	46
4.8 Processus d'implémentation des RAG [59]	48
4.9 Comparaison des RAG avec d'autres méthodes d'optimisation [32]	49
Troisième partie	54
5.1 Schématisation des étapes de la solution	55
5.2 Graphiques du modèle Llama3 8B	60
5.3 Graphiques du modèle Llama3 70B	61
5.4 Graphiques du modèle Mixtral-8x7b	62
5.5 Graphiques du modèle Gemma 7b	63
5.6 Graphiques du modèle Mistral-7B-Instruct	64
5.7 Graphiques du modèle Falcon-7b-Instruct	65
5.8 Graphique qui compare les résultats des modèles par rapport au nombre de shots	66
5.9 Graphique qui compare les résultats des modèles par rapport au questions	66
5.10 Schématisation de la solution avec DPO	68
5.11 Les caractéristiques de la carte graphique de la machine	69
5.12 Graphique montrant la perte d'entraînement et de validation par epoch.	70
5.13 Graphique montrant la perte d'entraînement et de validation normalisées par epoch.	71
5.14 La réponse du modèle à une question extérieure de la base de données	72
5.15 La réponse du modèle à une question de la base de données	73
5.16 Distribution des scores ROUGE-1_f pour chaque question	73

5.17	Schématisation de la solution avec RAG	75
5.18	Les résultats du <i>rouge</i> $- 1_f$ du modèle Mixtral 8x7b sur huit questions de la BDD	77
5.19	Les résultats du <i>rouge</i> $- 1_f$ du modèle Mixtral 8x7b avec variation de la température sur trois questions de la BDD	78
5.20	Les résultats du <i>rouge</i> $- 1_f$ du modèle Llama3 8b sur huit questions de la BDD	79
5.21	Les résultats du <i>rouge</i> $- 1_f$ du modèle Llama3 8b avec variation de la température sur trois questions de la BDD	80
5.22	Les résultats du <i>rouge</i> $- 1_f$ du modèle Llama3 70b sur huit questions de la BDD	81
5.23	Les résultats du <i>rouge</i> $- 1_f$ du modèle Llama3 70b avec variation de la température sur trois questions de la BDD	81
5.24	Les résultats du <i>rouge</i> $- 1_f$ du modèle Mistral 7b Instruct sur huit questions de la BDD	82
5.25	Les résultats du <i>rouge</i> $- 1_f$ du modèle Mistral 7b Instruct avec variation de la température sur trois questions de la BDD	83
5.26	Les résultats du ROUGE Score moyen sur huit questions des modèles testés	84
5.27	Aperçu d'ensemble sur l'interface utilisateur	86
5.28	Choix du modèle de génération	87
5.29	Ajustement de la température	87
5.30	Exemple de question d'un client de KPMG	87
5.31	Exemple de réponse générée par le chatbot	87
5.32	Possibilité d'effacer l'historique du chat	88
A.1	Récupération d'informations avec la bibliothèque Langchain	99
A.2	Comparaison du débit moyen des jetons par seconde du modèle Llama2 70B sur 150 requêtes	100
B.1	Notre modèle de langue [55]	104
B.2	Pipeline de collecte de données [55]	105
B.3	Modèle de récompense [55]	105
B.4	Ajustement fin du modèle de langage principal en utilisant le modèle de récompense et le calcul de perte PPO. [55]	106
B.5	Fonction de perte de PPO [56]	107
B.6	Zoom sur le calcul de Policy loss [55]	108

Liste des abréviations

IA : Intelligence Artificielle
KPMG : Klynveld, Peat, Marwick et Goerdeler
PFE : Projet de Fin d'Etudes
LLM : Large Language Model
NLP : Natural Language Processing
BERT : Bidirectional Encoder Representations from Transformers
GPT : Generative Pre-trained Transformer
LlaMa : Large Language Model Meta AI
RNN : Recurrent Neural Network
CNN : Convolutional Neural Network
NLU : Natural Language Understanding
FAQ : Frequently Asked Questions
IAg : Intelligence Artificielle Générative
PLM : Modèle de Langage Pré-entraîné
RAG : Retrieval-Augmented Generation
FS : Few-Shot
GPU : Graphics Processing Unit
SFT : Supervised Fine-Tuning
PEFT : Parameter Efficient Fine-Tuning
LoRa : Low Rank Adaptation
RLHF : Reinforcement Learning from Human Feedback
DPO : Direct Preference Optimization
KL : Kullback-Leibler
PDF : Portable Document Format
LCS : Longest common subsequence
NLI : Natural Language Inference
PWC : PricewaterhouseCoopers
EY : Ernst & Young
USD : United States Dollar
DZD : Dinar Algérien
AAS : Accountancy Assistance Services
IT : Information Technology
SLR : Systematic Literature Review
PPO : Proximal Policy Optimization
TII : Technology Innovation Institute
API : Application Programming Interface
CUDA : Compute Unified Device Architecture
CSV : Comma-Separated Values
BDD : Base De Données

Introduction générale

L'intelligence artificielle (IA), avec son potentiel disruptif, suscite un intérêt croissant dans le monde académique et professionnel. Spécifiquement, l'IA générative trouve une application notable dans le domaine des services financiers, où les technologies de génération de texte commencent à jouer un rôle transformateur. Ces technologies ne se contentent pas de produire du texte, mais facilitent également des interactions complexes et personnalisées via des chatbots, redéfinissant ainsi l'engagement client dans de nombreux secteurs.

KPMG, leader mondial du conseil, exploite ces avancées pour renforcer son offre dans le secteur bancaire. Le cabinet est reconnu pour son expertise approfondie et son engagement à fournir des solutions innovantes qui répondent aux défis complexes de ses clients. Chez KPMG, l'accent est mis sur l'utilisation de données sophistiquées et d'analyses prédictives pour optimiser les stratégies opérationnelles de ses clients.

Dans le cadre de ce mémoire, nous nous penchons sur le développement d'un chatbot avancé basé sur l'IA, conçu pour améliorer significativement la qualité du service client de KPMG par des réponses automatisées et pertinentes. Le rapport détaillera toutes les phases de conception, développement, et évaluation de ce projet, illustrant ainsi notre démarche complète pour ce projet de fin d'études (PFE).

Contexte

Les progrès récents en intelligence artificielle (IA) ont permis de développer des modèles de langage naturel capables de répondre instantanément aux questions dans divers domaines. Bien que performants, ces modèles ne sont généralement pas spécialisés. Notre projet se concentre sur la création d'un chatbot basé sur l'IA, spécifiquement conçu pour générer des réponses précises et pertinentes aux problématiques posées par les clients bancaire de KPMG. Ce chatbot utilise un modèle de langage pré-entraîné, enrichi par des données spécifiques, afin d'améliorer significativement la communication avec les clients.

Problématique

Dans un contexte où les interactions numériques deviennent primordiales dans le secteur bancaire, les entreprises cherchent constamment à améliorer l'efficacité et la personnalisation du service client. Face à l'augmentation des demandes de conseils financiers complexes et personnalisés, KPMG cherche à optimiser ses services à travers l'intégration de solutions technologiques avancées. Les défis à relever sont multiples : Comment l'intelligence artificielle peut-elle être exploitée pour répondre de manière précise et pertinente aux requêtes des clients bancaires, surpassant ainsi les performances des modèles de langage conventionnels, tout en améliorant l'efficacité opérationnelle et la satisfaction client ? Plus spécifiquement, quelles stratégies et technologies sont nécessaires pour assurer que les réponses fournies soient non seulement précises et contextuellement pertinentes, mais aussi adaptées aux méthodes de travail et aux techniques spécifiques utilisées chez KPMG ?

Objectifs

Notre travail vise à répondre aux problématiques posées en atteignant les objectifs suivants :

1. **Concevoir un chatbot basé sur l'IA** spécifiquement adapté pour répondre avec précision aux besoins des clients bancaires de KPMG.
2. **Adapter le chatbot aux méthodes de travail et aux techniques spécifiques** utilisées chez KPMG, assurant une intégration harmonieuse avec les processus opérationnels existants.
3. **Évaluer la performance du chatbot**, en mesurant la précision des réponses pour évaluer l'efficacité globale de l'outil.
4. **Surpasser les performances des modèles de langage traditionnels**, en offrant une précision et une expertise accrues dans les réponses liées au secteur bancaire.

Organisation du mémoire

Ce mémoire est structuré en trois parties principales, chacune abordant un aspect essentiel de notre étude et présentant une synthèse approfondie du travail effectué. Chaque partie vise à répondre à notre problématique et à atteindre les objectifs fixés.

Première Partie : État de l'art

Cette section établit les bases théoriques et techniques nécessaires pour comprendre et appliquer les technologies avancées utilisées dans notre projet. Elle se compose de :

- **Chapitre 1 : Fondements de l'Intelligence Artificielle** - Ce chapitre introduit les principes fondamentaux de l'IA, avec un focus particulier sur l'apprentissage profond et le traitement automatique du langage naturel (NLP), explorant leur utilisation dans le développement de chatbots.
- **Chapitre 2 : Modèles de Langage à Grande Échelle (LLMs)** - Ce chapitre propose une présentation détaillée des LLMs, examinant leur architecture, les défis de leur mise en œuvre, et les stratégies d'amélioration de leurs performance telles que le prompt engineering, le fine-tuning, l'apprentissage par renforcement et la technologie RAG (Retrieval-Augmented Generation).

Deuxième Partie : État des lieux

Cette partie présente le contexte opérationnel et commercial de KPMG, contextualisant l'application du projet. Elle inclut :

- **Chapitre 3 : Présentation de KPMG** - Ce chapitre décrit l'entreprise, ses divisions et l'organigramme de la branche algérienne.
- **Chapitre 4 : Étude de l'existant** - Ce chapitre analyse des besoins des clients bancaires de KPMG, étude des appels d'offres et des préparatifs pour la conception de la solution de chatbot.

Troisième Partie : Conception de la solution

Cette dernière partie détaille la conception et l'implémentation de la solution de chatbot, exploitant les technologies discutées précédemment :

- **Chapitre 5 : Présentation de la solution** - Ce chapitre offre une vue exhaustive de la conception du chatbot basé sur l'IA générative, décrivant les outils et technologies utilisés, et retraçant les étapes du développement, de la base de données à la mise en production.
- **Chapitre 6 : Déploiement de la solution** - Dans ce chapitre, nous avons détaillé les étapes de la création de l'interface utilisateur de notre solution, ainsi que les différentes fonctionnalités offertes par celle-ci.

Première partie

État des lieux

Chapitre 1

Présentation de l'organisme d'accueil

1.1 Introduction

Cette section offre un aperçu détaillé de KPMG, un acteur clé du secteur de l'audit et du conseil. Nous débiterons par une présentation générale de l'entreprise, suivie d'un examen de l'organigramme de KPMG Algérie pour illustrer sa structure organisationnelle et départementale. Nous détaillerons également les principaux services offerts par l'entreprise.

1.2 Présentation de KPMG

1.2.1 KPMG International

KPMG est un réseau mondial de cabinets de services d'audit et de conseil, exerçant dans près de 156 pays, avec un effectif global de quelque 207 000 employés. Chaque cabinet membre est rattaché à KPMG International, une coopérative basée en Suisse, tout en conservant une indépendance légale et opérationnelle.

Considéré comme l'un des quatre géants du secteur, connus sous le nom de Big Four, aux côtés de Deloitte, PWC et EY, KPMG joue un rôle de premier plan dans l'industrie mondiale de l'audit et du conseil. La firme est vue comme une référence dans le monde des affaires et compte parmi ses clients 60% des 1 000 premières entreprises mondiales, dont notamment **Google**, **Samsung**, **Philips** ou encore **Wal-Mart**. En 2018, son chiffre d'affaires s'élevait à 28,96 milliards USD.

1.2.2 KPMG Algérie

KPMG Algérie SPA, affilié au réseau mondial KPMG International Coopérative, fut parmi les pionniers des cabinets d'audit et de conseil en Algérie. S'inscrivant dans un mouvement de libéralisation de secteurs clés de l'économie, il est devenu en 2002 le premier des *Big Four* à s'établir en Algérie. En mai 2009, dans le but d'étendre sa couverture nationale et de se rapprocher de sa clientèle de l'ouest du pays, un bureau fut inauguré à Oran. Dix-sept ans après sa fondation, le cabinet comptait environ 120 collaborateurs et un chiffre d'affaires avoisinant les 800M DZD.

Aujourd'hui, KPMG Algérie offre des services d'audit et de conseil à un large éventail d'entreprises nationales et multinationales opérant dans divers secteurs, consolidant ainsi sa position de leader sur le marché algérien. L'organisation s'appuie sur une expertise sectorielle approfondie pour proposer des solutions adaptées à des domaines vitaux de l'économie :

- **Finance** : banques, assurances.
- **Industrie** : automobile, construction, produits industriels, pharmacie.
- **Technologie** : informatique, électronique, télécommunication.
- **Consommation** : agroalimentaire, biens de consommation, distribution.

- **Loisirs** : tourisme, hôtellerie.

1.3 Organigramme de KPMG Algérie

KPMG Algérie Spa est établi dans un immeuble situé au lot 94 de la zone d'affaires de Bab Ezzouar à Alger, où son siège abrite la direction générale ainsi que l'ensemble de ses départements. L'entreprise est structurée en cinq départements distincts :

- **Département Fiscal et Juridique (Tax & Legal)** : Ce département est constitué de 10 juristes et 14 fiscalistes.
- **Département d'Assistance en comptabilité (AAS : Accountancy Assistance Services)** : Il rassemble 20 consultants.
- **Département Audit** : Ce département compte 32 auditeurs.
- **Département Conseil (Advisory)** : Il comprend 30 consultants, y compris l'équipe Data/IT auquel on est affecté durant notre stage.
- **Département Administration** : Ce département comprend trois services : Comptabilité & Finances, Ressources Humaines, Moyens Généraux-Informatique, ainsi que le secrétariat. Il est composé d'un effectif total de 12 personnes.

La direction de chaque département est confiée à un Associé (Partner). Tandis que le Partner Président Directeur Général est à la tête du cabinet et des départements d'administration, et d'audit. Ce dernier se charge aussi du bureau d'Oran dont la principale mission est de vendre les services de KPMG dans la région. La figure 1.1 démontre une schématisation de l'organigramme de KPMG Algérie.

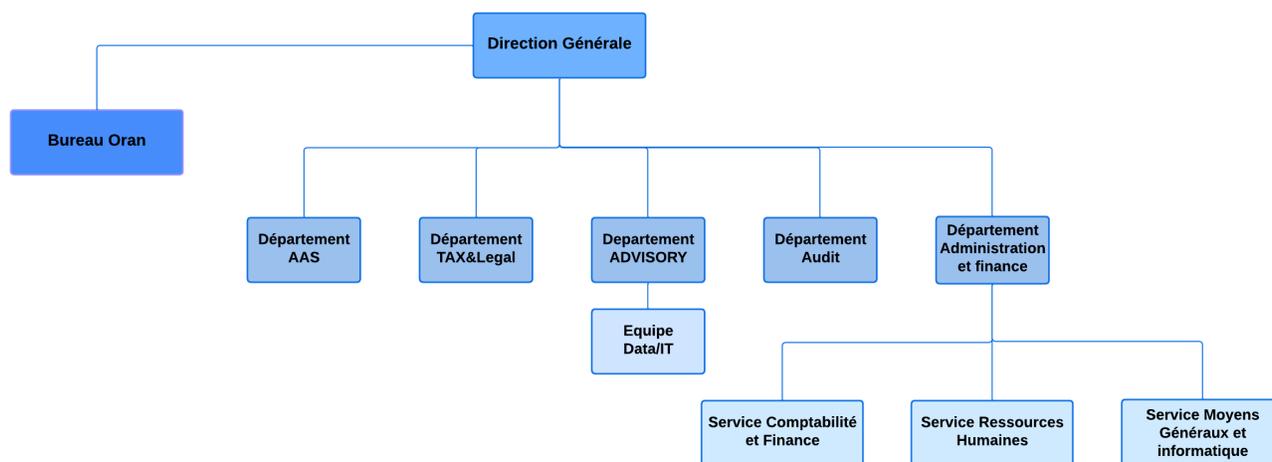


FIGURE 1.1: Organigramme de KPMG Algérie

1.4 Principaux services

1.4.1 Fiscal et juridique

L'environnement législatif en Algérie évolue rapidement en réponse aux changements économiques, s'adaptant aux tendances actuelles ainsi qu'aux exigences régionales et mondiales. Cette adaptation ne se limite pas seulement à l'introduction de nouvelles lois, mais inclut également d'importants changements dans l'interprétation des règlements existants.

Dans ce contexte dynamique, KPMG offre une gamme de services juridiques et fiscaux :

- **Services juridiques** :

- Création de sociétés.
- Études sur les modalités d'implantation et la réalisation d'opérations.
- Revue juridique de conformité.
- Assistance contractuelle (financement, assistance technique, travail, etc.).
- Secrétariat juridique.
- **Services fiscaux :**
 - Audit fiscal.
 - Revue des déclarations fiscales.
 - Revue fiscale dans le cadre de l'exécution des contrats.
 - Due diligence.
 - Développement, amélioration et suivi des processus fiscaux.
 - Assistance à la mise en place de procédures de conformité fiscale.

1.4.2 Audit financier

Au travers de ses services d'audit, KPMG joue un rôle crucial en renforçant la confiance des acteurs économiques quant à la pertinence et la fiabilité des informations financières. Les activités principales des équipes d'audit chez KPMG comprennent :

- L'audit des états financiers afin d'émettre un avis sur leur conformité et sincérité, selon les normes comptables algériennes ou internationales.
- La documentation et l'analyse des processus ainsi que de l'environnement de contrôle interne dans le cadre d'audits informatisés, incluant le test de ces systèmes et la mise en place de procédures de contrôle analytique ou substantif, utilisant des outils avancés pour l'interrogation de données.
- La formulation de recommandations pour corriger les lacunes des contrôles internes identifiées lors des audits, et la contribution à la restitution de ces informations de manière à créer de la valeur ajoutée pour la direction ou les instances de gouvernance.

1.4.3 Conseil (Advisory)

Pour améliorer le fonctionnement et la performance des entreprises, le cabinet KPMG propose des recommandations et des solutions couvrant plusieurs domaines clés, organisés comme suit :

Transactions et Restructuration (Transaction Services) :

- Assistance à l'identification et à l'évaluation des risques lors de fusions, cessions ou rapprochements d'entreprises.
- Due diligence pour des transactions d'achat ou de vente.
- Assistance pour les data rooms.
- Identification des risques.

Conseil en Comptabilité (Accounting Advisory Services) :

- Gestion de la comptabilité.
- Préparation des déclarations fiscales et sociales.
- Reporting.
- Préparation des états financiers.
- Conversion aux normes IFRS/US GAAP.

Conseil en Management : Le cabinet accompagne les entreprises dans leurs projets de transformation, touchant à l'organisation, à la fonction finance et aux systèmes d'information.

Conseil en Management des Risques : Il offre des solutions pour évaluer et améliorer les dispositifs de contrôle interne et de gestion des risques des entreprises.

1.4.4 Conseil en informatique (IT Advisory)

Le conseil en informatique chez KPMG fournit un support essentiel aux autres domaines de conseil, en se focalisant principalement sur les aspects suivants :

Conseil en Management des Risques :

- Audit Interne et Gouvernance : Le cabinet aide ses clients à intégrer la dimension technologique dans leur audit interne, gestion des risques, et gouvernance.
- Sécurité des SI : Assistance aux clients pour protéger et sécuriser leurs actifs technologiques.
- Contrôle Interne IT et Attestation : Assistance pour évaluer et optimiser le dispositif de contrôle interne des systèmes d'information.

Conseil en Management :

- Accompagnement de la DSI : Soutien dans la gestion des systèmes d'information, incluant l'élaboration de la stratégie IT, du schéma directeur, et de sa gouvernance.
- Transformation Technologique : Aide à la définition et mise en œuvre de projets de transformation technologique, y compris la sélection de solutions informatiques, l'assistance à la maîtrise d'ouvrage, l'assurance qualité, et la conduite du changement.

Ces services sont essentiellement proposés par le département Advisory, plus précisément à travers les services d'IT Advisory, dans le contexte de l'accompagnement à la transformation technologique.

1.5 Structure organisationnelle

Pour saisir la structure du cabinet et les dynamiques hiérarchiques entre les collaborateurs, il est crucial d'examiner la répartition des responsabilités et les pratiques de communication interne. Afin de clarifier les rôles spécifiques des employés au sein des différents départements (à l'exception du Département Administration), nous détaillons ci après (figure 1.2) les différents grades en vigueur chez KPMG Algérie.

- Junior débutant
- Junior confirmé
- Senior (2 années : Senior 1 et Senior 2)
- Senior Expérimenté
- Assistant Manager
- Manager
- Partner

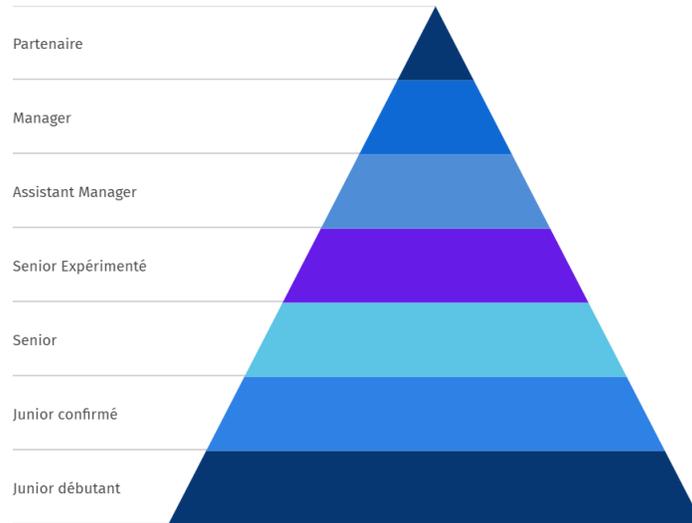


FIGURE 1.2: L'hierarchie des grades chez KPMG Algérie

Le cabinet est dirigé par le Président Directeur Général, qui est à la tête de tous les départements. Chaque département est à son tour dirigé par un collaborateur titulaire du grade de Partner, ce dernier est à la tête de tous les Managers de son département. En dernier lieu, Chaque collaborateur titulaire d'un autre grade est rattaché à un Manager, qui en est le responsable. La figure 1.3 illustre cette structure :

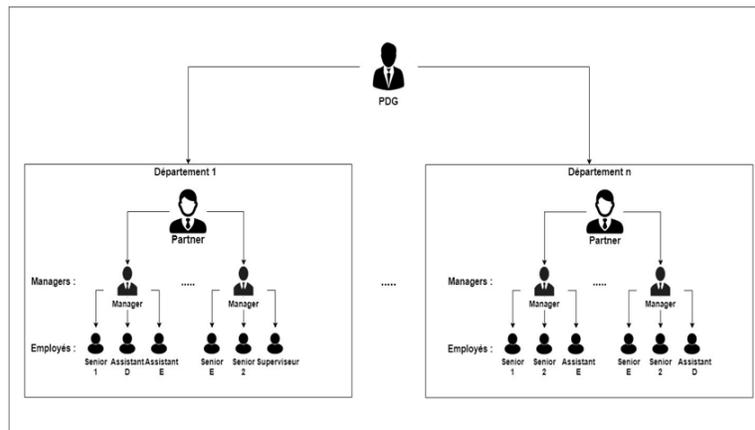


FIGURE 1.3: Structuration hiérarchique de KPMG Algérie

1.6 Structure départementale

Pour comprendre la structure organisationnelle de KPMG, il est important de noter que les responsables de projets sont généralement les managers, ou, dans de rares cas, le Partner. Il est possible qu'un employé participe à un projet dirigé par un manager à qui il n'est pas directement rattaché. Pour la durée du projet, cet employé sera alors sous l'autorité du manager responsable du projet. Cette configuration illustrée dans la figure 1.4 offre plusieurs avantages à KPMG :

- Les collaborateurs peuvent participer simultanément à plusieurs projets.
- L'échange de connaissances entre employés est fortement encouragé.
- La spécialisation des employés tend à accroître leur motivation.

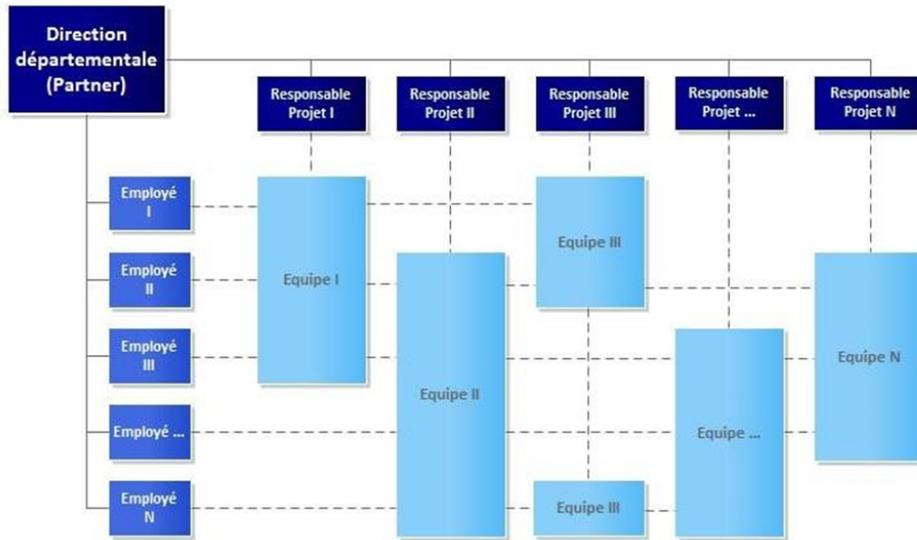


FIGURE 1.4: Structure départementale de KPMG Algérie

1.7 Conclusion

La recherche effectuée dans ce chapitre nous a permis de cerner efficacement le cadre de travail de l'organisme d'accueil, de définir précisément son domaine d'action, et de saisir sa structure organisationnelle et départementale. Maintenant que nous avons établi l'environnement de travail, nous sommes prêts à nous plonger dans l'étude approfondie du système existant.

Chapitre 2

Étude de l'existant

2.1 Introduction

Dans ce chapitre, nous allons présenter le système actuel des appels d'offres avec les clients de KPMG et analyser la nécessité de notre solution dans ce contexte. Nous explorerons également les solutions similaires existantes déjà sur le marché, en mettant un accent particulier sur les solutions d'IA générative dans le domaine bancaire. Cette étude nous permettra de situer notre approche par rapport aux solutions existantes et de souligner les avantages spécifiques que notre solution peut apporter.

2.2 Clients de KPMG dans le domaine bancaire

Depuis plusieurs décennies, KPMG s'est imposé comme le partenaire de référence dans le secteur de la banque et de la finance. Les équipes de KPMG interviennent à la fois en tant qu'auditeurs et conseillers auprès de l'ensemble des acteurs bancaires français, couvrant l'intégralité de leurs métiers. L'implication de KPMG dans les différentes instances professionnelles et autorités de tutelle renforce sa position de leader.

2.2.1 Secteur de la Fintech

La Fintech, combinaison de la technologie et de la finance, constitue un facteur clé de création de valeur. Innovante et disruptive, la Fintech redessine les contours de la finance à l'échelle mondiale et accélère la mutation de l'industrie financière. En tant qu'acteur majeur dans l'écosystème des Fintech, KPMG accompagne l'ensemble des parties prenantes dans leurs projets d'innovation, de transformation et d'évolution de leur modèle économique. KPMG couvre ainsi tous les métiers du secteur, notamment les néobanques, les services de paiement, d'investissement, de financement, l'assurtech, la regtech et la blockchain.

2.2.2 Secteur de Customer & Operations Banque

KPMG accompagne les banques de la définition à l'exécution de leurs projets de transformation. Les banques poursuivent actuellement une dynamique forte de transformation autour de cinq priorités clés :

1. Repenser les parcours et processus clients pour répondre aux nouvelles attentes ;
2. Accélérer l'innovation et les synergies avec les Fintech ;
3. Transformer les contraintes réglementaires en opportunités ;
4. Faire évoluer les lignes métiers et les fonctions internes ;
5. Intégrer l'humain dans la transformation.

KPMG accompagne les banques tout au long de leurs initiatives de transformation en mobilisant les talents et expertises nécessaires pour co-construire des solutions adaptées au contexte et à la culture de chaque client. Cela inclut :

- La définition de leur stratégie et de leurs modèles opérationnels.
- L'exécution et la sécurisation des programmes de transformation métier et pluridisciplinaire.

2.2.3 Secteur de la Blockchain & Crypto-actifs

KPMG propose une offre de services pour accompagner ses clients autour des technologies blockchain et des actifs numériques. Le rapprochement avec les équipes de Blockchain Partner, leader français du conseil sur les technologies blockchain, a permis en 2021 de constituer une équipe de professionnels dédiés aux technologies blockchain et aux actifs numériques. L'offre KPMG blockchain & crypto-actifs comprend notamment :

- L'élaboration d'une stratégie et d'une feuille de route ;
- La gestion de projet et le cadrage ;
- La gestion des actifs numériques au sein d'une organisation ;
- La gestion de la cybersécurité, des attestations, des risques et de la conformité ;
- L'accompagnement réglementaire, comptable et fiscal en matière de blockchain.

2.2.4 Secteur du Advisory Bank

KPMG aide les banques dans leurs transformations, la gestion des risques et la mise en conformité. En bénéficiant d'un accès aux meilleures pratiques du marché grâce à un réseau mondial pluridisciplinaire, KPMG répond aux défis suivants :

- Adaptation aux évolutions réglementaires :
 - Mise en place des nouvelles normes et réglementations françaises et internationales ;
 - Mise en œuvre des reportings réglementaires ;
 - Sécurité financière : KYC, lutte contre le blanchiment et contre le financement du terrorisme, sanctions/embargos.
- Gestion des risques et conformité :
 - Gestion des risques : crédit, marché, opérationnel, liquidité, valorisation ;
 - Gouvernance, audit et contrôles internes, contrôle permanent.
- Transformations métiers et organisationnelles :
 - Amélioration de l'efficacité opérationnelle ;
 - Optimisation des processus ;
 - Transformation des fonctions finance et risques et optimisation des processus comptables et financiers ;
 - Intervention dans des situations d'urgence.

2.3 Les appels d'offres des clients de KPMG

Habituellement, lorsqu'un client lance un appel d'offre chez KPMG et que cette mission est officiellement prise en charge, plusieurs étapes sont suivies de manière systématique :

— **Lancement et cadrage**

Cette première étape comprend le cadrage du projet, la définition du périmètre d'intervention, l'identification des parties prenantes, et la validation du planning du projet. Cette phase initiale est cruciale pour établir des bases solides et des objectifs clairs pour la mission.

— **Étude de marché et analyse**

Dans cette étape, une étude de marché est effectuée ainsi qu'une analyse approfondie du problème du client. Cette analyse permet de comprendre les enjeux spécifiques et de définir les besoins exacts

du client, en identifiant les opportunités et les défis potentiels.

— **Choix de la solution**

Cette phase comprend l'élaboration du cahier des charges de la mission et une analyse approfondie des solutions existantes. L'objectif est de faire un choix final de la solution à proposer, en s'appuyant sur une évaluation rigoureuse des options disponibles et en tenant compte des spécificités du projet et des attentes du client.

— **Élaboration d'une feuille de route**

Cette étape inclut la préparation d'une feuille de route pour la concrétisation de la solution, ainsi que l'élaboration du budget prévisionnel. La feuille de route détaille les étapes à suivre pour la mise en œuvre de la solution, les ressources nécessaires et les échéances.

2.4 Défis principaux de ce système

Bien que le système actuel des appels d'offres chez KPMG soit une stratégie clé pour la réussite de leurs missions, un défi majeur réside dans le fait que les consultants sont facturés pour des heures bien définies. Cela signifie que si un client souhaite poser une question à un consultant, celui-ci n'est pas disponible à tout moment, notamment en dehors des heures de travail, les jours fériés, ou même pendant les périodes de congés. De plus, l'élaboration de la feuille de route est une étape qui pourrait être automatisée sans nécessiter l'intervention des consultants.

2.5 Revue de la littérature

Ces derniers mois, les grands modèles de langage (LLMs) comme ChatGPT [11], BERT [12] et LLaMa [13], ont atteint des résultats de pointe dans le domaine du traitement du langage naturel (NLP) et ont révolutionné divers autres domaines. Ces modèles, basés sur l'architecture du transformeur, ont été entraînés sur des données linguistiques génériques à l'échelle du web et peuvent comprendre les instructions en langage naturel et générer des réponses de type humain.

Grâce à leur capacité à répondre de manière semblable à celle des humains, les LLMs ont réalisé des avancées significatives dans le domaine de l'IA conversationnelle, améliorant grandement les performances des chatbots et des assistants virtuels [14]. La création de chatbots spécialisés est devenue plus facile en ajustant des grands modèles de langage génériques pré-entraînés pour des tâches spécifiques ou en utilisant la génération augmentée par récupération (RAG) pour enrichir les modèles avec des connaissances spécifiques au domaine [15].

2.5.1 Les chatbots

La conception de chatbots utilise diverses méthodes, allant des systèmes basés sur des règles aux chatbots basés sur l'IA générative, comme ChatGPT-3.5 ou ChatGPT-4. Ces derniers sont basés sur l'architecture transformers. Cependant, ils rencontrent fréquemment le problème des "hallucinations" [16], où ils génèrent des réponses qui sont syntaxiquement et sémantiquement correctes mais déconnectées de la réalité et basées sur de fausses hypothèses, surtout lorsqu'ils traitent des questions spécifiques à un domaine [17].

Pour pallier ces défis, des chatbots orientés vers des domaines spécifiques ont été développés en affinant des modèles pré-entraînés via des techniques comme le fine-tuning supervisé [19]. Par exemple dans cette étude [3], une recherche a été entreprise pour la création d'un chatbot pour un cabinet comptable en utilisant le fine-tuning supervisé sur plusieurs LLMs, de même, un LLM conçu pour la recommandation a été mis œuvre en utilisant la méthode Fine Tuning [23]. Dans cette étude également [23], un LLM a été développé avec la méthode Fine Tuning pour automatiser les dernières étapes d'exécution d'un processus de revue systématique de la littérature (SLR) qui implique la synthèse des connaissances. Bien que cette

méthode soit largement utilisée, elle nécessite des ressources considérables et une grande quantité de données pour un entraînement efficace.

2.5.2 Reinforcement learning with human feedback

L'une des méthodes de fine-tuning les plus notables est l'apprentissage par renforcement avec feedback humain (RLHF), qui a montré des résultats impressionnants, notamment après sa première utilisation par OpenAI pour leur modèle ChatGPT [24]. Cette méthode a été largement utilisée pour construire des chatbots spécialisés dans divers domaines. Par exemple, Raithubot [25] est un chatbot spécifique à l'agriculture créé en entraînant le modèle Pythia avec RLHF. Dans le domaine de l'éducation, le LLM MM-PhyRLHF [26] a démontré de très bons résultats pour répondre à des problèmes de physiques complexes par rapport aux méthodes existantes. Cette approche intègre les retours humains pendant la phase d'apprentissage, augmentant la capacité du modèle à répondre efficacement tout en minimisant les hallucinations. L'algorithme le plus connu pour RLHF est l'optimisation de politique proximale (PPO) [27], qui utilise un modèle de récompense pour guider le modèle à produire les réponses correctes .

2.5.3 Direct Preference Optimization

Bien que RLHF ait suscité un grand intérêt pour l'amélioration des LLMs, cette méthode reste une procédure complexe et instable, notamment en ce qui concerne l'entraînement du modèle de récompense et la recherche du bon modèle reflétant les préférences humaines [39]. En tant qu'alternative, l'optimisation directe des préférences (DPO) a été introduite [28]. La DPO est une méthode plus simple qui élimine le modèle de récompense et entraîne le modèle de base pour prédire directement la réponse attendue [40]. Cette méthode a montré d'excellents résultats pour affiner les LLMs et réduire les hallucinations [29]. Le modèle Phoenix [30] et le modèle Zephyr [31] sont deux exemples de la mise en œuvre de la DPO, démontrant des progrès significatifs dans l'alignement des modèles de langage et permettant de rivaliser avec des modèles de plus grande taille.

2.5.4 Génération augmentée par récupération (RAG)

Bien que les méthodes de fine-tuning aient montré de grandes capacités et des résultats surprenants, elles restent coûteuses et nécessitent des ressources énormes. Par conséquent, une nouvelle approche, retrieval augmented generation (RAG), a été introduite, montrant de très bons résultats dans la réduction des hallucinations [32], [60] des LLMs et la construction de chatbots spécialisés dans des domaines spécifiques [34]. Par exemple, le MufassirQAS [35] LLM est un chatbot conçu en utilisant la méthode RAG pour répondre à des questions islamiques en utilisant ChatGPT-3.5 turbo. Dans cette méthode, les connaissances ne sont pas nécessairement stockées dans les paramètres du modèle mais sont explicitement acquises de manière plug-and-play, ce qui permet une grande évolutivité [36].

2.5.5 Chatbots dans le domaine bancaire

Les chatbots FAQ ont connu une augmentation, surtout avec les LLMs et les différentes méthodes introduites chaque jour. Des chatbots conçus pour répondre à des questions spécifiques à un domaine ont été développés. Par exemple, le premier service de chatbot dans le secteur financier a été Erica de Bank of America, introduit en mai 2017. Erica fournissait des réponses textuelles et vocales simples, incluant les détails des transactions, les montants limites et les soldes de compte. En Inde, Un LLM de conseil financier a été développé par une équipe intégrant des expertises juridiques, financières et de gestion dans une plate-forme cohérente que les conseillers financiers peuvent appliquer pour guider efficacement leurs clients [38]. Cependant, le domaine du conseil bancaire manque encore de chatbots dédiés. Certains modèles ont été améliorés en exploitant des ensembles de données spécialisés tels que le dataset. [Banking77](#), bien que

cet ensemble de données reste assez général et ne contienne pas de questions très spécifiques sur le conseil bancaire.

2.6 Énoncé de la problématique

La problématique initiale à laquelle KPMG fait face réside dans l'amélioration de son service client, plus précisément dans l'efficacité du système des appels d'offres. Pour cela, notre problématique peut être résumée par les questions suivantes :

- Comment répondre en temps réel aux besoins des clients de KPMG en utilisant les dernières tendances technologiques ?
- Comment exploiter les technologies de l'IA générative pour améliorer le service client de KPMG, tout en économisant du temps et des coûts ?

2.7 Conclusion

Ce chapitre a mis en évidence l'état actuel du système d'appel d'offres chez KPMG, en précisant les différents secteurs du domaine bancaire avec lesquels l'entreprise travaille, les étapes d'un appel d'offres et les divers services offerts à ses clients. Cette étude nous a également permis de cerner la problématique et de souligner les différents points à améliorer dans ce système.

Une revue de la littérature a montré que, bien que de nombreuses recherches et développements aient été réalisés, les méthodes d'améliorations des modèles existantes requièrent souvent de grandes quantités de données et de ressources, ou bien les modèles sont entraînés sur des données qui ne sont pas toujours adaptées aux questions spécifiques du conseil bancaire. Nous proposons deux approches pour concrétiser notre solution : en utilisant soit la méthode Direct Preference Optimization (DPO) soit les Retrieval-Augmented Generation (RAG) pour améliorer les performances d'un modèle de langage pré-entraîné avec des données spécifiques à l'entreprise. Cette spécificité distingue notre solution des offres existantes sur le marché.

Dans le chapitre suivant, nous explorerons les différentes notions du domaine de L'intelligence artificielle générative avec ses différentes techniques et méthodes.

Deuxième partie

État de l'art

Chapitre 3

Fondements de l'intelligence artificielle

3.1 Introduction

Ce chapitre explore les fondements de l'intelligence artificielle (IA), une technologie pivotale transformant de nombreux secteurs. Nous commençons par définir l'IA, soulignant son importance et son évolution, puis nous nous penchons sur le deep learning, une technique essentielle qui permet des avancées significatives en permettant un traitement complexe des données. Le traitement du langage naturel (NLP) est examiné comme un domaine crucial qui permet aux machines de comprendre et de produire le langage humain, essentiel pour le développement de chatbots. Ces derniers représentent des applications pratiques du NLP, facilitant les interactions quotidiennes entre humains et machines. Ensemble, ces technologies démontrent comment l'IA façonne et continuera de révolutionner notre interaction avec le numérique.

3.2 Définition de l'IA

L'intelligence artificielle (IA) est un domaine de l'informatique qui vise à créer des machines capables de réaliser des tâches qui, traditionnellement, nécessitent l'intelligence humaine. Ces tâches incluent la prise de décision, la résolution de problèmes, la compréhension du langage humain, et la reconnaissance visuelle et sonore. L'IA intègre des disciplines telles que les mathématiques, la psychologie, la philosophie, la neurologie, et bien sûr, l'informatique.

L'objectif de l'IA n'est pas seulement de simuler ou de reproduire l'intelligence humaine mais aussi de développer des systèmes capables d'améliorer leur propre performance sans intervention humaine directe. Cela est souvent réalisé par des processus d'apprentissage automatique et de réseautage neuronal, où les machines s'adaptent et apprennent de nouvelles stratégies à partir des données qu'elles traitent.

3.3 L'apprentissage profond (Deep Learning)

Le deep learning ou apprentissage profond est un sous-ensemble de l'intelligence artificielle et une branche spécialisée du machine learning. Contrairement aux méthodes traditionnelles de machine learning, qui nécessitent souvent une ingénierie manuelle des caractéristiques, le deep learning apprend des représentations de données de manière automatique. Il utilise des architectures complexes de réseaux de neurones profonds qui permettent à la machine d'apprendre et de faire des prédictions ou des décisions basées sur des données, sans suivre strictement des règles prédéterminées. Cette capacité d'apprentissage automatique lui permet d'exécuter des tâches de manière plus intuitive et efficace, particulièrement dans des domaines tels que la reconnaissance visuelle et le traitement du langage naturel.

3.3.1 Fonctionnement du Deep Learning

Le deep learning fonctionne grâce à des réseaux de neurones artificiels, inspirés par le cerveau humain, qui comprennent plusieurs couches de neurones. Chaque couche reçoit des informations de la couche précédente, les traite via des fonctions d'activation, et les transmet à la suivante comme il est illustré dans la Figure 3.1. Ce processus permet au système de reconnaître des patterns allant de simples à complexes. Par exemple, il peut apprendre à identifier d'abord les lettres, puis les mots dans un texte, ou les formes avant de reconnaître des visages dans une image.

L'apprentissage se réalise par rétropropagation : le réseau ajuste les poids des connexions neuronales en fonction des erreurs entre les prédictions et les véritables résultats. Ce mécanisme d'amélioration continue permet au modèle de devenir plus précis au fil du temps. Un modèle bien entraîné est capable de généraliser à partir de nouveaux exemples, c'est-à-dire de reconnaître des éléments ou des concepts jamais rencontrés durant la phase d'apprentissage, en se basant sur les connaissances acquises. La diversité et la qualité des données d'entraînement sont essentielles pour la performance globale du réseau.

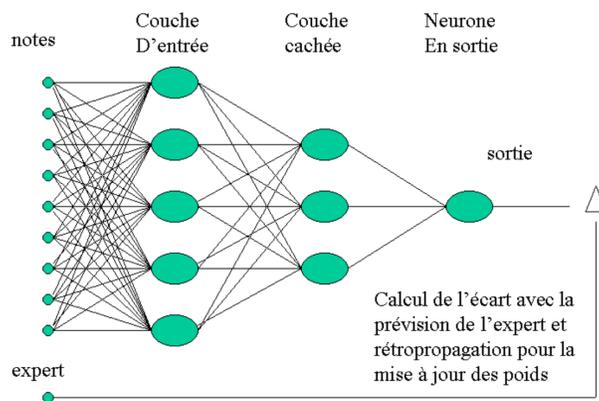


FIGURE 3.1: Réseau de neurones artificiels [6]

3.3.2 Applications du Deep Learning

Le deep learning a révolutionné de nombreux domaines grâce à sa capacité à traiter et à analyser de grandes quantités de données avec une précision remarquable. Parmi les applications les plus notables, on trouve :

- **Vision par ordinateur** : Le deep learning permet aux machines de reconnaître des objets, des visages, et même des émotions dans des images et des vidéos. Cette technologie est largement utilisée dans des applications telles que la surveillance de sécurité, où elle peut identifier des comportements suspects, ou dans le secteur de la santé pour l'analyse d'images médicales. Les réseaux de neurones convolutifs (CNNs) sont particulièrement efficaces pour traiter les données visuelles et sont au cœur de ces progrès.
- **Traitement du langage naturel** : Dans le NLP, le deep learning a permis de grandes avancées dans la compréhension et la génération du langage humain. Cela inclut la traduction automatique, la génération de réponses textuelles, et l'analyse de sentiment. Les modèles comme BERT et GPT, basés sur l'architecture Transformer, ont établi de nouveaux standards pour l'efficacité des chatbots et des assistants personnels intelligents.
- **Création de chatbots** : Le deep learning a également transformé le développement de chatbots, les rendant plus interactifs, intuitifs et capables de fournir des réponses pertinentes. Grâce à l'analyse avancée du langage naturel, les chatbots peuvent comprendre des requêtes complexes et fournir des

réponses personnalisées, améliorant ainsi l'engagement client et la satisfaction dans des domaines tels que le service clientèle et les services financiers.

Ces applications montrent comment le deep learning peut être utilisé pour améliorer l'automatisation et la précision dans des tâches variées, offrant des solutions innovantes qui étaient auparavant hors de portée des techniques informatiques traditionnelles.

3.4 Le traitement automatique du langage naturel (NLP)

Le NLP, pour Natural Language Processing ou Traitement du Langage Naturel, est une branche interdisciplinaire qui fusionne l'informatique et la linguistique. Elle se concentre sur l'interaction entre les ordinateurs et le langage humain, avec pour objectif principal de permettre aux machines de comprendre, manipuler et générer du langage naturel de manière efficace. Le NLP permet ainsi aux systèmes informatiques d'interpréter, de répondre et de simuler le langage humain d'une manière qui est à la fois utile et significative pour les utilisateurs. Cette capacité est cruciale pour de nombreuses applications modernes, telles que les assistants virtuels, les systèmes de traduction automatique et les outils d'analyse de texte, qui aident à simplifier les interactions entre les humains et les machines, améliorant ainsi l'accessibilité et l'efficacité des technologies de l'information.

3.4.1 Techniques et Méthodes Clés en NLP

Le Traitement Automatique du Langage Naturel (NLP) emploie une variété de techniques pour décomposer et comprendre le langage humain. Ces méthodes fondamentales incluent :

- **Tokenisation** : Ce processus divise le texte en unités plus petites, appelées tokens, qui peuvent être des mots, des phrases ou des symboles. Cette étape est essentielle pour préparer les données pour les analyses plus complexes.
- **Analyse syntaxique** : Elle implique l'identification de la structure grammaticale d'une phrase, en utilisant des règles pour analyser la relation entre les mots. Cela permet aux machines de comprendre comment les mots dans une phrase s'articulent entre eux pour former un sens.
- **Analyse sémantique** : Cette méthode cherche à interpréter le sens des mots dans leur contexte spécifique. Elle va au-delà de la structure grammaticale pour comprendre les nuances du langage, y compris les ambiguïtés et les variations dans les significations.
- **Word Embeddings** : Les embeddings, ou plongements lexicaux, transforment les tokens en vecteurs de nombres qui capturent des aspects contextuels et sémantiques des mots. Cela permet aux modèles de traiter le texte dans un espace vectoriel dense, où des mots similaires ont des représentations numériques proches, facilitant ainsi des tâches telles que la recherche sémantique et la classification de texte.

Outre ces techniques de base, le NLP s'appuie également sur des modèles avancés pour traiter et générer le langage :

- **Modèles statistiques** : Ces modèles utilisent des techniques statistiques pour comprendre et anticiper les structures linguistiques. Ils peuvent, par exemple, calculer la probabilité qu'un mot en suive un autre dans une phrase, ce qui est fondamental pour des tâches comme la correction automatique et la complétion de texte.
- **Modèles basés sur le deep learning** : Avec l'avènement du deep learning, le NLP a fait des avancées significatives. Des réseaux de neurones profonds, tels que les réseaux de neurones convolutifs (CNN) et récurrents (RNN), sont utilisés pour traiter des séquences de texte longues et complexes. Plus récemment, les architectures de type Transformer, qui utilisent des mécanismes d'attention pour améliorer la qualité de la modélisation du langage, ont révolutionné la traduction automatique, la génération de texte, et la compréhension du langage naturel. [44]

3.4.2 Application du NLP dans le Développement de Chatbots

Le Traitement Automatique du Langage Naturel (NLP) est crucial pour le développement des chatbots, en permettant aux machines de comprendre, traiter, et générer le langage humain. Le NLP englobe plusieurs sous-disciplines essentielles telles que la reconnaissance vocale, la compréhension du langage naturel (NLU), la génération de texte, et l'analyse des sentiments, chacune contribuant à des aspects différents de la fonctionnalité des chatbots.

- **Compréhension du langage naturel (NLU)** : Le NLU est au cœur de l'interaction des chatbots, leur permettant de déchiffrer le contexte et l'intention derrière les paroles ou les textes des utilisateurs. Cette sous-discipline utilise des techniques d'apprentissage automatique pour analyser la syntaxe et la sémantique, permettant au chatbot de comprendre des requêtes complexes telles que "Quel temps fait-il dehors?" et de déterminer si l'utilisateur demande une prévision météorologique plutôt que de faire une remarque générale.
- **Génération de texte** : Essentielle pour la réponse des chatbots, cette technologie emploie des modèles de langage avancés pour formuler des réponses cohérentes et contextuellement appropriées. Par exemple, un chatbot peut répondre à la question météorologique mentionnée précédemment par "Le ciel est clair avec une température de 23 degrés." Ces réponses sont générées par des modèles qui calculent la probabilité des suites de mots, souvent basés sur des réseaux de neurones ou des architectures Transformer. [45]
- **Reconnaissance vocale** : Cette technologie transforme le langage parlé en texte, permettant aux chatbots de traiter des commandes vocales. Par exemple, dans un environnement de service client, un chatbot peut recevoir des instructions vocales d'un utilisateur et les convertir en requêtes textuelles qui seront ensuite analysées pour une réponse appropriée.
- **Analyse des sentiments** : Cette fonction permet aux chatbots de reconnaître et de réagir aux émotions exprimées dans les messages des utilisateurs, ce qui est crucial pour maintenir une interaction empathique et personnalisée. L'analyse des sentiments peut aider un chatbot à ajuster le ton de sa réponse en fonction de l'humeur perçue de l'utilisateur, améliorant ainsi l'engagement et la satisfaction du client.

3.5 Chatbots

À l'ère de la transformation numérique, la communication est essentielle tant pour les entreprises que pour les clients. Cependant, l'interaction interhumaine n'est pas toujours réalisable, efficace ou souhaitable. C'est là qu'interviennent les chatbots. Les chatbots, ou agents conversationnels, sont des programmes informatiques qui simulent des interactions humaines en utilisant le traitement du langage naturel (NLP) et l'apprentissage automatique. Ils se divisent en deux catégories : les chatbots basés sur des règles, qui suivent des commandes spécifiques, et les chatbots basés sur l'IA, capables d'apprendre et de s'adapter au fil du temps.

Employés dans divers secteurs, les chatbots améliorent l'efficacité opérationnelle et la satisfaction clientèle en répondant aux questions, en exécutant des tâches et en offrant des recommandations personnalisées. Ils sont intégrés sur de multiples plateformes comme les sites web, les applications mobiles et les réseaux sociaux, facilitant les interactions entreprises-clients. Les progrès récents en NLP et en IA ont augmenté leur popularité et sophistication, les rendant essentiels dans l'engagement et le service client modernes.

3.5.1 Types de Chatbots

Dans le développement de chatbots, il est crucial de comprendre les deux principales catégories : les chatbots basés sur des règles et ceux basés sur l'intelligence artificielle.

- **Chatbots Basés sur des Règles** : Ces chatbots fonctionnent selon des scénarios préprogrammés et répondent aux requêtes des utilisateurs à travers un ensemble fixe de règles et de parcours d'utilisateur. Ils sont capables de gérer des interactions simples, telles que répondre à des FAQ ou guider un utilisateur dans des processus prédéfinis. Cependant, les chatbots basés sur des règles ne peuvent pas aller au-delà de leur programmation initiale et ne s'adaptent pas aux situations non prévues.
- **Chatbots Basés sur l'IA** : Ces chatbots utilisent des technologies avancées telles que l'apprentissage automatique et le NLP pour comprendre le langage naturel et les intentions des utilisateurs. Contrairement aux chatbots basés sur des règles, ils peuvent apprendre des interactions passées et améliorer continuellement leur capacité à répondre de manière plus naturelle et contextuelle. Ces chatbots sont particulièrement utiles pour des applications nécessitant une compréhension plus profonde et une plus grande flexibilité, comme le support client personnalisé, la génération de leads, ou même la fourniture de conseils personnalisés.[45]

Dans le paysage évolutif des chatbots, en plus des chatbots basés sur des règles et ceux basés sur l'intelligence artificielle, il est essentiel de considérer les chatbots spécialisés dans un domaine spécifique :

- **Spécialisation dans un Domaine Spécifique** : Les chatbots spécialisés, tels que ceux utilisés dans le secteur bancaire, sont conçus pour gérer des tâches et des requêtes spécifiques à ce domaine, offrant des réponses précises et conformes aux réglementations financières. Ils possèdent une connaissance approfondie des termes financiers et des pratiques bancaires, ce qui les rend particulièrement efficaces pour des services spécialisés.

3.5.2 Intelligence Artificielle Générative et Chatbots

L'intelligence artificielle générative (IAg) désigne une branche de l'intelligence artificielle qui permet aux machines de créer du contenu original et de générer de nouvelles idées de manière autonome. Dans le domaine des chatbots, cette technologie est particulièrement utile pour élaborer des réponses qui semblent naturelles et humaines.

L'IAg s'appuie sur des techniques avancées d'apprentissage automatique, notamment les réseaux de neurones profonds, pour analyser et apprendre à partir de vastes ensembles de données textuelles. En identifiant des motifs et des tendances dans les données, elle peut produire des réponses textuelles pertinentes et cohérentes. Par exemple, dans le cadre des chatbots, l'IAg peut utiliser l'histoire des interactions d'un utilisateur pour mieux comprendre ses besoins et personnaliser les réponses en conséquence. Cela rend les chatbots non seulement plus réactifs mais également capables d'engager des conversations plus complexes et significatives.

L'application de l'IAg dans les chatbots transforme leur fonctionnement : au lieu de se limiter à des réponses préprogrammées, les chatbots peuvent maintenant participer activement à des échanges en générant des réponses qui s'adaptent dynamiquement au contexte de la conversation. Cette capacité de génération de contenu enrichit l'expérience utilisateur, rendant les interactions avec les chatbots plus fluides et agréables. [45]

3.5.3 Avantages et Défis de l'IA Générative dans le développement des chatbot

L'intelligence artificielle générative (IAg) apporte des avantages significatifs dans le développement des chatbots, tout en présentant certains défis qui nécessitent une attention particulière.

- **Avantages** :
 - **Réponses Naturelles et Humaines** : Permet aux chatbots de produire des réponses qui semblent plus naturelles et humaines, améliorant ainsi l'expérience utilisateur et rendant les interactions plus agréables.
 - **Adaptabilité et Apprentissage Continu** : Les chatbots peuvent s'adapter et apprendre de chaque interaction, ce qui les rend progressivement plus efficaces et plus pertinents pour l'utilisateur.

- **Défis :**
 - **Exigence de Grandes Quantités de Données :** L'IA générative nécessite de vastes ensembles de données pour l'apprentissage, ce qui peut être coûteux et complexe en termes de collecte et de traitement.
 - **Risques de Réponses Inappropriées :** Peut parfois générer des réponses inattendues ou inappropriées, y compris des réponses offensantes ou non pertinentes, en raison des biais dans les données d'entraînement.
 - **Nécessité de Contrôle et de Modération :** Il est crucial de mettre en place des mécanismes de contrôle et de modération robustes pour atténuer les risques de réponses inappropriées et assurer que les interactions restent respectueuses et pertinentes.

3.6 Conclusion

Ce chapitre a posé les bases théoriques nécessaires à la compréhension des technologies sous-jacentes à notre projet de fin d'études. Nous avons exploré les principes de l'intelligence artificielle, le fonctionnement et les applications du deep learning, les techniques avancées de traitement automatique du langage naturel, et les divers aspects des chatbots. Forts de ces connaissances, nous orientons notre projet vers la création d'un chatbot innovant en utilisant l'intelligence artificielle générative, visant à améliorer l'interaction utilisateur et à répondre efficacement aux besoins spécifiques des clients de KPMG.

Chapitre 4

Large Language Models (LLMs)

4.1 Introduction

Les Grands Modèles de Langage (LLMs) ont profondément transformé le domaine de l'intelligence artificielle et du traitement automatique du langage. Ce sont des modèles d'IA, souvent basés sur l'architecture Transformer, conçus pour comprendre et générer le langage humain. Entraînés sur de vastes corpus textuels, ils excellent dans une multitude de tâches linguistiques, allant de la classification de texte à la génération de contenu. Leur efficacité provient de l'intégration de technologies avancées telles que l'attention, l'apprentissage par transfert, et l'élargissement des réseaux neuronaux.[10]

Dans ce chapitre, nous aborderons toutes les notions relatives aux LLMs, y compris leur architecture, leurs applications, les techniques et approches d'amélioration, ainsi que les méthodes d'évaluation des résultats d'un LLM. Ces éléments permettront de mieux comprendre comment ces modèles fonctionnent et comment ils peuvent être optimisés pour diverses applications linguistiques.

4.2 Terminologies sur les LLM

Avant de détailler les différentes notions des large language models (LLMs), il est nécessaire de présenter les diverses terminologies pertinentes dans ce domaine.

- **Température** : Un hyperparamètre qui contrôle l'aléatoire des résultats du modèle. Une température de 0 signifie que le modèle produira toujours le jeton de probabilité le plus élevé.
- **Fenêtre de Contexte** : La fenêtre de contexte est le nombre de jetons (tokens) pris en compte lors de la prédiction du jeton suivant.
- **Modalité** : Une catégorie de données de haut niveau. Par exemple, les nombres, le texte, les images, les vidéos et l'audio sont cinq modalités différentes.
- **Tokens** : Un jeton est une unité fondamentale de texte qu'un grand modèle de langage (LLM) utilise pour traiter et comprendre le langage. Par exemple, dans la phrase « J'aime la science des données », il y a quatre jetons : « J' », « aime », « science » et « données ».
- **Prompt** : Un prompt est une entrée spécifique donnée au modèle pour guider son comportement et générer le texte souhaité. C'est une instruction ou une requête qui aide le LLM à comprendre la tâche ou le contexte auquel il doit répondre. Elle peut être une phrase, un paragraphe ou même juste quelques mots-clés, selon la tâche à accomplir.
- **Paramètres** : Les paramètres sont les valeurs apprises qu'un modèle acquiert au cours de l'entraînement pour faciliter les prédictions ou les classifications sur de nouvelles données. Dans les réseaux de neurones, ces paramètres sont couramment désignés comme les poids et les biais, dictant comment les données d'entrée sont transformées en prédictions de sortie.
- **Modèle de Base** : Un modèle de base est un grand modèle de langage (LLM), parfois appelé modèle de langage pré-entraîné (PLM), qui est suffisamment robuste pour servir de base pouvant être

utilisée tel quel ou ajustée/adaptée pour être utilisée dans de nouveaux domaines.

4.3 Les applications des LLM

Avec un tel éventail d'applications, les grands modèles de langage peuvent servir à différentes finalités.[9]

- **Chatbots** : Amélioration de l'interaction utilisateur grâce à des réponses plus naturelles et contextuelles.
- **Biologie computationnelle** : Accélération des recherches en biologie grâce à l'analyse rapide des données génétiques.
- **Programmation informatique** : Assistance aux développeurs en générant automatiquement du code ou en suggérant des corrections.
- **Raisonnement et décision** : Aide à la résolution de problèmes complexes nécessitant analyse et déduction.
- **Robotique et agents incarnés** : Amélioration des interactions et de l'autonomie des robots grâce à une meilleure compréhension des commandes naturelles.
- **Sciences sociales et psychologie** : Analyse des comportements humains et modélisation des interactions sociales pour des études plus approfondies.
- **Génération de données synthétiques** : Création de données pour entraîner d'autres modèles sans compromettre la confidentialité ou l'intégrité des données réelles.

4.4 Les défis des LLM

Les modèles de langage offrent de multiples avantages et applications pratiques. Cependant, en tant que technologies avancées, ils sont également sujets à divers défis.[9]

- **Jeu de données impénétrables** : Difficulté à évaluer la qualité des très grands jeux de données utilisés pour l'entraînement.
- **Dépendance aux tokenizers** : Les tokenizers peuvent introduire des biais et limiter la compréhension multilingue.
- **Coûts élevés de pré-entraînement** : Les ressources nécessaires pour entraîner des LLMs sont considérables, tant en termes financiers qu'en consommation d'énergie.
- **Surcoût du fine-tuning** : Adapter un modèle pré-entraîné à des tâches spécifiques peut être coûteux et complexe.
- **Latence élevée lors de l'inférence** : La taille et la complexité des LLMs peuvent entraîner des retards significatifs lors de leur utilisation en production.
- **Longueur de contexte limitée** : Les LLMs ont du mal à gérer de très longs contextes, ce qui peut restreindre leur utilisation dans certains scénarios.
- **Fragilité des prompts** : Sensibilité à la formulation exacte des prompts, affectant la stabilité des réponses.
- **Hallucinations** : Tendance des modèles à générer des informations fausses ou non vérifiées.
- **Comportements non alignés** : Difficultés à faire correspondre les comportements du modèle avec les attentes humaines ou éthiques.
- **Connaissances obsolètes** : Incapacité à intégrer automatiquement les nouvelles connaissances ou informations.
- **Evaluations fragiles** : Manque de robustesse des méthodes d'évaluation des performances des modèles.
- **Manque de reproductibilité** : Difficultés à reproduire les résultats des LLMs à cause de leur complexité et de leur dépendance à des configurations spécifiques.

4.5 Architecture des LLMs :

Dans le domaine de la génération de texte, les modèles qui utilisent l'architecture des Transformers se sont avérés extrêmement performants. Ces modèles possèdent une capacité remarquable à détecter et interpréter les relations à distance entre les mots dans une phrase, une capacité essentielle pour cette tâche. L'introduction du mécanisme de self-attention [41] par l'architecture Transformer a marqué une avancée révolutionnaire dans le traitement automatique du langage naturel (NLP).

4.5.1 Structure Générale des LLMs

L'architecture encodeur-décodeur [42], [43], [47] se compose de deux parties principales : l'encodeur et le décodeur. Comme le montre la figure 4.1. Chaque partie a un rôle spécifique dans le traitement de l'information :

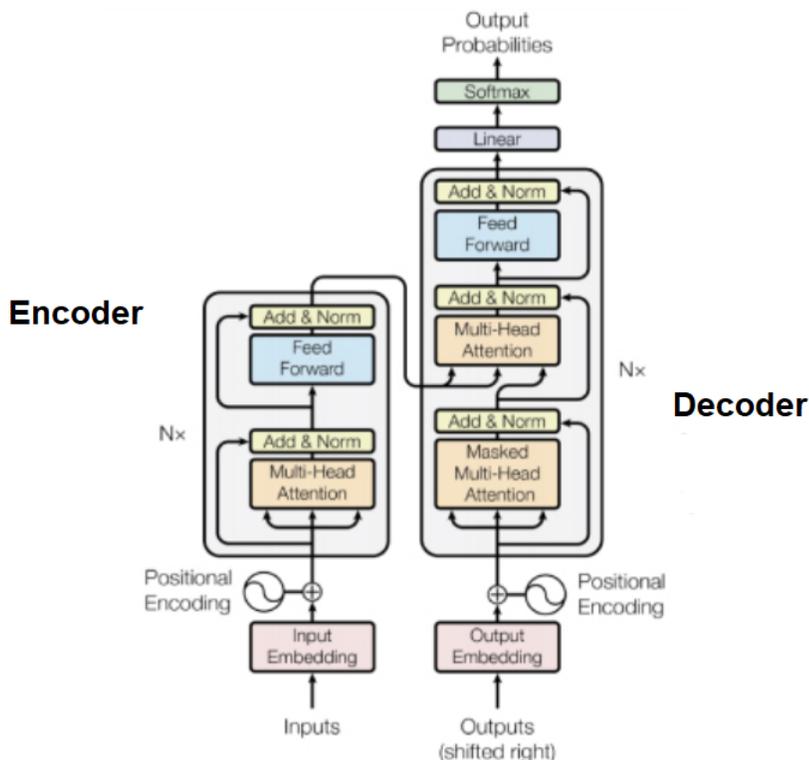


FIGURE 4.1: L'architecture du modèle transformateur [41]

Encodeur

L'encodeur [49] prend en entrée une séquence de mots ou de phrases et les convertit en une série de représentations vectorielles. Ces représentations contiennent les informations contextuelles de l'entrée. Dans le modèle Transformer, l'encodeur est constitué de plusieurs couches identiques empilées les unes sur les autres. Chaque couche comprend principalement deux sous-couches : une sous-couche de self-attention multi-têtes et une sous-couche de réseau feed-forward. Les connexions résiduelles autour de chaque sous-couche, suivies d'une normalisation, permettent de préserver l'information et d'accélérer la convergence du modèle.

Décodeur

Le décodeur [50] génère la sortie en se basant sur les représentations fournies par l'encodeur. Il est également composé de plusieurs couches, similaires à celles de l'encodeur, mais avec une couche supplémentaire d'attention qui aide à se concentrer sur les parties pertinentes de l'entrée de l'encodeur pour chaque étape de la génération de la sortie. Cette attention inter-couches permet au décodeur de prendre en compte l'ensemble de l'entrée tout en générant chaque mot de sortie, améliorant ainsi la cohérence et la pertinence du texte généré.

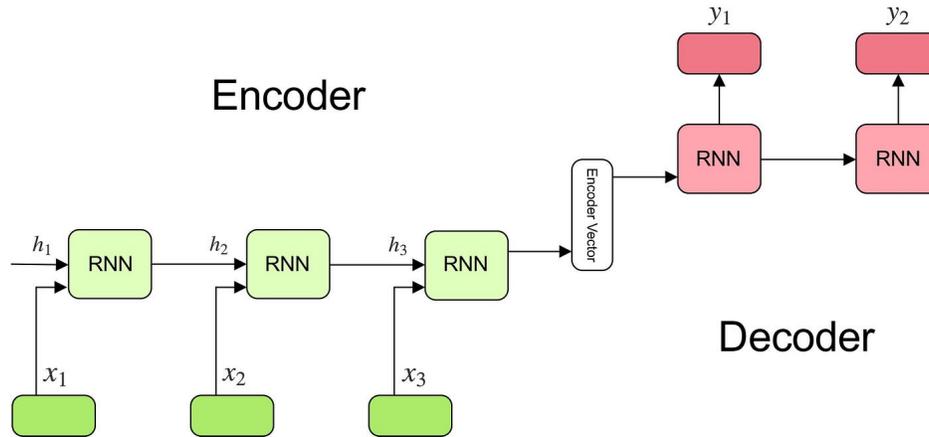


FIGURE 4.2: Les modèles encodeur décodeur [61]

4.5.2 Composants de Base des LLMs

L'architecture transformer repose principalement sur les couches détaillées ci-dessous. La figure 4.3 illustre l'interaction entre ces couches.

Couche d'Embedding

Chaque mot ou token d'entrée est converti en un vecteur dense qui capture sa signification sémantique et syntaxique. Ces embeddings sont essentiels pour permettre au modèle de traiter le texte de manière significative.

Mécanisme d'Attention (self attention mechanism)

Au cœur des LLMs se trouve le mécanisme de self-attention [48], qui permet au modèle de pondérer l'importance relative des différents mots dans une phrase. Il fonctionne en transformant la séquence d'entrée en trois vecteurs (query, key, value), et calcule une somme pondérée des valeurs, basée sur la similarité entre les vecteurs de query et de key. Ce processus aide le modèle à se concentrer sur les informations pertinentes et à capturer des dépendances à longue portée.

Couches Cachées (Feed-Forward)

Après le traitement par self-attention, les sorties sont passées à travers des réseaux feed-forward qui modifient les représentations. Ces couches sont répétées plusieurs fois dans le modèle, permettant la conceptualisation d'abstractions de plus haut niveau.

Attention Multi-Têtes (Multi-head Attention)

L'attention multi-têtes améliore le mécanisme de self-attention en divisant les embeddings en plusieurs sous-ensembles traités parallèlement. Cette technique permet au modèle de se concentrer simultanément sur différentes parties d'une séquence d'entrée, enrichissant ainsi la représentation contextuelle du texte.

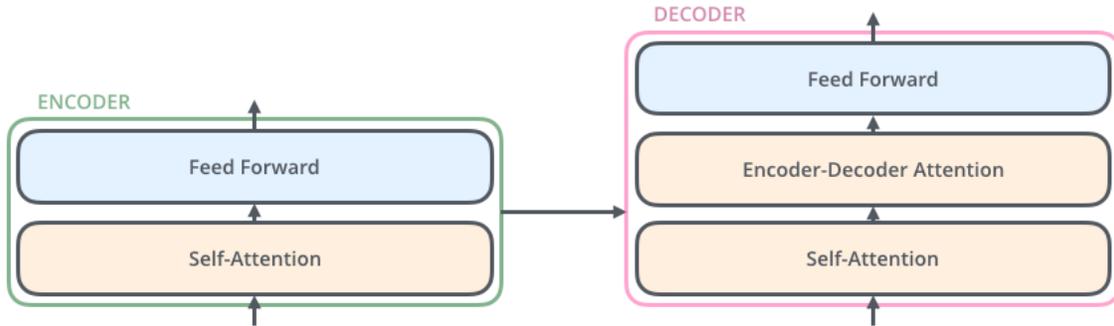


FIGURE 4.3: Les couches de l'architecture transformer [62]

4.6 Amélioration des LLM (Large Language Models)

Comme nous l'avons vu dans la section précédente sur les modèles de langage à grande échelle, ou LLM, ceux-ci représentent une avancée significative dans le domaine du traitement du langage naturel. Cependant, pour exploiter pleinement leur potentiel, il est essentiel de recourir à des techniques spécifiques d'amélioration qui optimisent leur performance et leur applicabilité dans divers contextes. Cette section se concentre sur les méthodologies avancées utilisées pour affiner les LLM, incluant le Prompt Engineering, le fine-tuning, l'apprentissage par renforcement (RL) et la génération augmentée par récupération (RAG). Nous explorerons comment ces techniques contribuent à rendre les LLM plus précis, flexibles et efficaces pour des applications réelles.

4.6.1 L'ingénierie de prompts (Prompt Engineering)

Le prompt engineering est une discipline émergente qui se concentre sur le développement et l'optimisation de prompts pour maximiser l'efficacité des modèles de langage (LMs) dans diverses applications et domaines de recherche. Cette pratique est essentielle pour exploiter pleinement les capacités des grands modèles de langage (LLMs) et pour comprendre leurs limitations. Les chercheurs et les développeurs emploient le prompt engineering pour améliorer la performance des LLMs sur une variété de tâches, allant de la réponse à des questions simples à des opérations de raisonnement plus complexes, comme le raisonnement arithmétique.

L'utilisation de prompts simples peut produire des résultats significatifs, mais la qualité de ces derniers dépend largement de la quantité et de la précision des informations fournies au modèle. Un prompt efficace peut inclure des éléments tels que des instructions spécifiques, des questions, du contexte pertinent, des entrées utilisateur, et des exemples illustratifs. Ces éléments sont cruciaux pour orienter précisément le modèle, permettant ainsi d'obtenir des réponses plus précises et adaptées aux besoins spécifiques de l'utilisateur. [46]

2.6.1.1 Techniques de Prompt Engineering

Le prompt engineering consiste à exploiter les capacités des modèles de langage de grande envergure (LLMs) pour formuler des prompts qui traduisent précisément les objectifs souhaités. Cette approche

permet de tirer parti d'une diversité de techniques pour engendrer une variété infinie de contenus, des articles de presse minutieusement élaborés aux poèmes qui reflètent un style et un ton spécifiques. Parmi les méthodologies couramment employées, l'apprentissage avec peu d'exemples (few-shot learning) :

a. Le Few-Shot Learning

Dans le domaine du traitement automatique du langage (NLP), le few-shot learning permet aux modèles de langage de grande envergure, qui ont été pré-entraînés sur d'importants ensembles de données textuelles, de généraliser leur capacité à comprendre et à exécuter des tâches connexes mais non vues auparavant avec seulement quelques exemples, sans mises à jour des poids.

Cette méthode implique de donner au modèle K exemples de contexte et de complétion, suivis d'un exemple final de contexte pour lequel le modèle doit générer la complétion. Nous fixons généralement K dans une fourchette de 10 à 100, selon le nombre d'exemples qui peuvent s'adapter dans la fenêtre de contexte du modèle.

Les avantages principaux incluent une réduction significative du besoin de données spécifiques à la tâche et une adaptation rapide à de nouvelles tâches sans nécessiter un réentraînement complet. Comme l'explique un article de recherche [51] :

“Few-Shot (FS) is the term we will use in this work to refer to the setting where the model is given a few demonstrations of the task at inference time as conditioning, but no weight updates are allowed. ... The main advantages of few-shot are a major reduction in the need for task-specific data and reduced potential to learn an overly narrow distribution from a large but narrow fine-tuning dataset.”

Composantes du Few-Shot Learning :

- **Description de la Tâche** : Une brève explication de ce que le modèle est censé accomplir, par exemple, “Traduire de l'anglais vers le français”.
- **Exemples** : Plusieurs exemples sont fournis pour montrer au modèle le type de prédiction attendu, tels que “sea otter => loutre de mer”.
- **Prompt** : Le début d'un nouvel exemple que le modèle doit compléter en générant le texte manquant, par exemple, “cheese => ”.

L'exemple présenté dans la Figure 4.4 met en évidence le concept.

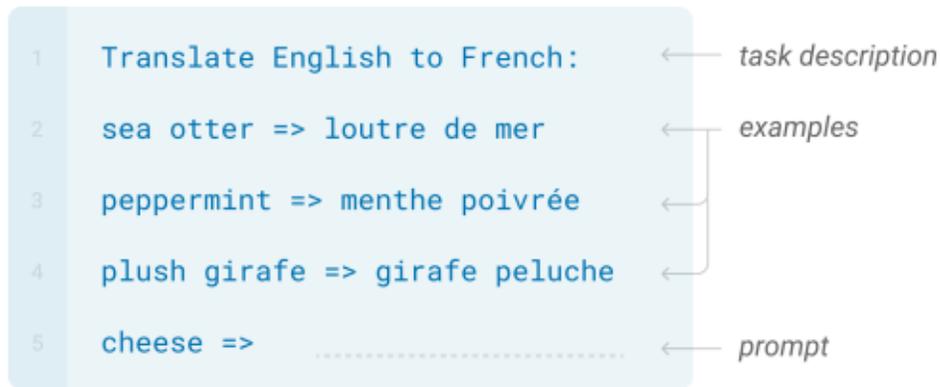


FIGURE 4.4: Illustration d'un exemple du Few-shot [51]

b. Autres Techniques de Prompt Engineering

D'autres techniques sont également utilisées pour enrichir les interactions avec les modèles de langage. Chacune possède des spécificités qui contribuent à diversifier et améliorer les réponses générées par les LLMs :

- **One-Shot Learning (Apprentissage avec un seul exemple)** : Le One-Shot Learning est similaire au few-shot, il implique de donner un seul exemple au modèle, accompagné d'une description de la tâche, utile pour des tâches nécessitant un style ou un ton spécifique. La Figure 4.5 démontre clairement le principe à travers un exemple.

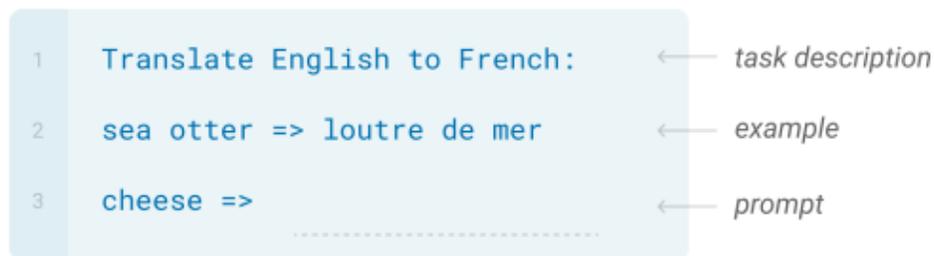


FIGURE 4.5: Illustration d'un exemple du One-Shot [51]

- **Zero-Shot Learning (Apprentissage sans exemple)** : Le Zero-Shot Learning ne fournit aucune démonstration ; le modèle reçoit uniquement une instruction décrivant la tâche, il est idéal pour des tâches générales. De même, l'exemple fourni dans la Figure 4.6 illustre le principe du zero-shot.

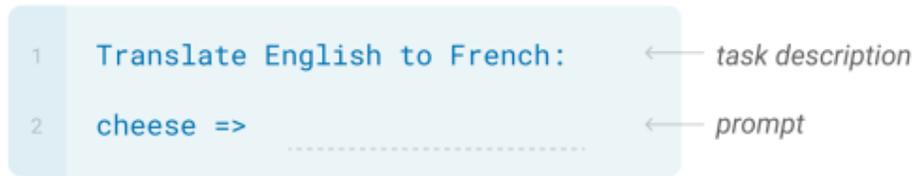


FIGURE 4.6: Illustration d'un exemple du Zero-shot [51]

- **Chain-of-thought prompts (Prompts en chaîne de pensée)** : Une technique qui décompose les problèmes complexes en étapes plus simples pour améliorer la logique du modèle.
- **Contextual augmentation (Augmentation contextuelle)** : Cette méthode enrichit les prompts avec des informations contextuelles pour augmenter la pertinence des réponses. [51]

2.6.1.2 Applications du Prompt Engineering

Le few-shot learning est particulièrement bénéfique dans les scénarios où collecter de grandes quantités de données d'entraînement est impraticable ou coûteux. Il permet une flexibilité significative dans l'application des LLM à une variété de tâches en réduisant le besoin de données tout en maintenant une performance élevée. De plus le prompt engineering facilite une adaptation rapide des solutions d'IA à de nouveaux domaines, sans nécessiter de vastes ensembles de données pour un réentraînement complet, accélérant ainsi le déploiement de technologies adaptées aux besoins réels.

Le prompt engineering joue aussi un rôle crucial dans l'évaluation et la sélection de modèles de langage adaptés, cela implique la mise en place d'une série de tests avec les méthodes du prompt engineering où différents modèles sont sollicités pour répondre à des prompts variés qui reflètent les scénarios d'utilisation réels. Le modèle qui produit les réponses les plus précises, pertinentes et contextuellement adaptées est alors choisi pour une intégration plus poussée. Cette approche permet non seulement de garantir la performance du modèle choisi en termes de précision, de pertinence et de cohérence des réponses mais aussi d'affiner davantage les prompts pour maximiser l'efficacité du modèle dans des conditions d'utilisation variées. Il aide également à réduire les coûts et les ressources en choisissant le modèle le plus efficace, évitant ainsi l'usage de modèles surdimensionnés inefficaces pour des applications spécifiques.

4.6.2 Ajustement fin (fine-tuning)

Le fine-tuning est une technique de deep learning qui permet d'adapter un modèle pré-entraîné à des tâches spécifiques en ajustant ses paramètres. Cette méthode est utilisée pour affiner les capacités des modèles de l'IAg, afin qu'ils répondent mieux aux exigences spécifiques sans nécessiter un entraînement complet. Elle est particulièrement utile pour les modèles ayant de nombreux paramètres, comme les grands modèles de langage (LLMs) utilisés en traitement automatique du langage (NLP). En ré-entraînant un modèle sur des données spécifiques, les entreprises peuvent améliorer l'efficacité du modèle, produisant des résultats plus rapides et mieux adaptés à des applications telles que le support client ou domaines bancaires.

Le fine-tuning est une des possibilités permettant de conserver un fort pouvoir de généralisation des LLM avec des objectifs métiers sur un secteur (banque, assurance, médical) précis.

Il faut tout de même garder à l'esprit que le fine-tuning nécessite des compétences techniques en Data, ainsi que de la puissance de calcul et des GPU à disposition afin de pouvoir obtenir des résultats cohérents.

2.6.2.1 Processus et Méthodologies du Fine-tuning

L’ajustement fin, ou *fine-tuning*, commence par utiliser les poids d’un modèle pré-entraîné comme base pour un entraînement supplémentaire sur un ensemble de données spécifique. Ce processus peut impliquer diverses méthodes d’apprentissage, telles que supervisé, par renforcement, ou semi-supervisé, adaptées aux cas d’utilisation spécifiques du modèle.

Les données utilisées reflètent les tâches particulières et le domaine pour lequel le modèle est optimisé, permettant de transmettre des connaissances précises. Le fine-tuning peut mettre à jour tous les poids du réseau (Ajustement fin complet) ou se concentrer seulement sur certains (Parameter Efficient Fine-Tuning), pour optimiser les performances tout en évitant que les connaissances antérieures sont perdues, ou bien mettre à jour les poids du modèle en se basant sur le score d’un modèle de récompense (Apprentissage par Renforcement ‘à partir de Feedback Humain)

- **Ajustement fin complet**

Cette méthode actualise l’ensemble des poids du réseau neuronal, ce qui est conceptuellement similaire au pré-entraînement mais diffère principalement par l’état initial des paramètres et l’ensemble de données utilisé. Pour prévenir l’oubli catastrophique, on ajuste soigneusement les hyperparamètres comme le taux d’apprentissage, assurant que le modèle conserve une bonne généralisation tout en s’adaptant à de nouvelles tâches.

Le Supervised Fine-Tuning (SFT) est une forme spécifique d’ajustement fin complet qui utilise des ensembles de données annotées pour entraîner le modèle de manière plus ciblée.

- **Parameter Efficient Fine-Tuning (PEFT)**

Le fine-tuning complet est très gourmand en ressources de calcul, rendant cette approche souvent trop coûteuse et impraticable. Le PEFT consiste à actualiser uniquement un sous-ensemble de paramètres sélectionnés, réduisant ainsi les besoins en ressources computationnelles et en mémoire. Ces méthodes PEFT sont généralement plus stables que l’ajustement fin complet, en particulier pour les applications en traitement du langage naturel. [52]

Cette approche inclut diverses techniques :

- **Ajustement fin partiel** : Mise à jour seulement de certains paramètres critiques, souvent les couches externes, tout en ”gelant” les autres. Cela diminue la charge computationnelle tout en conservant la pertinence du modèle pour la tâche cible.
- **Ajustement fin additif** : Ajout de nouvelles couches ou paramètres au modèle existant, formant des modules adaptateurs qui sont ensuite entraînés indépendamment des poids pré-entraînés, lesquels restent inchangés.
- **Low Rank Adaptation (LoRA)** : LoRA utilise la reparamétrisation pour optimiser les poids, transformant les matrices de grande dimension en représentations de rang inférieur. Cette technique réduit le nombre de paramètres à entraîner, accélérant l’ajustement fin et diminuant la mémoire nécessaire pour stocker les mises à jour. [53]

- **Apprentissage par Renforcement à partir de Feedback Humain (RLHF)**

Outre les méthodes traditionnelles de fine-tuning telles que l’ajustement fin complet et le Parameter Efficient Fine-Tuning (PEFT), il est essentiel de reconnaître l’importance de l’Apprentissage par Renforcement à partir de Feedback Humain (RLHF). Le RLHF partage les principes de base du fine-tuning, en ce sens qu’il utilise également la mise à jour des poids d’un modèle pré-entraîné. Toutefois, il diffère dans sa capacité à ajuster dynamiquement ces poids en fonction des retours directs sur les réponses générées par le modèle. Cela permet une adaptation plus précise du modèle aux attentes et préférences des utilisateurs, en améliorant la qualité et la pertinence des interactions [54]. Cette technique sera explorée plus en détail dans l’Annexe B 5.5, mettant en lumière son application pratique pour améliorer la pertinence et la précision des interactions modelées par LLM.

Bien que l’apprentissage par renforcement avec feedback humain (RLHF) soit devenu une méthode populaire pour l’ajustement fin efficace des grands modèles de langage pour réaliser des tâches

spécifiques, cette approche présente un problème majeur : son coût computationnel élevé. RLHF affine les modèles de langage en utilisant le feedback humain pour former un modèle de récompense, qui guide ensuite le modèle de langage vers la génération de réponses préférées. Bien que ce processus soit efficace, il est confronté à des défis tels que la nécessité de développer des modèles de récompense à grande échelle équivalents au modèle original, entraînant d'importantes demandes computationnelles.

Après avoir examiné le processus et les méthodologies du fine-tuning, notamment la méthode de Renforcement de l'Apprentissage Humain (RLHF), il est essentiel de considérer une alternative qui s'adapte encore mieux aux exigences spécifiques de notre projet. Cette alternative est l'Optimisation Directe des Préférences (DPO), une méthode distincte qui offre une approche différente et potentiellement moins coûteuse pour l'ajustement fin des modèles de langage à grande échelle. Cette nouvelle méthode sera introduite et détaillée dans la subsection suivante, mettant en lumière comment elle répond aux défis posés par RLHF tout en offrant une efficacité comparable.

4.6.3 Optimisation de Préférence Direct (DPO)

Pour répondre au besoin computationnel de RLHF, nous pouvons passer à une méthode plus récente appelée Optimisation de Préférence Directe ou DPO. Contrairement au RLHF, la DPO simplifie le processus en éliminant le besoin d'un modèle de récompense. Elle fonctionne en créant un ensemble de données de paires de préférences humaines, chaque paire contenant un prompt et deux réponses possibles — une préférée et une non préférée. Le LLM est ensuite ajusté pour maximiser la probabilité de générer des réponses préférées et minimiser la probabilité de générer des réponses non préférées.

2.6.3.1 Processus d'implémentation du DPO :

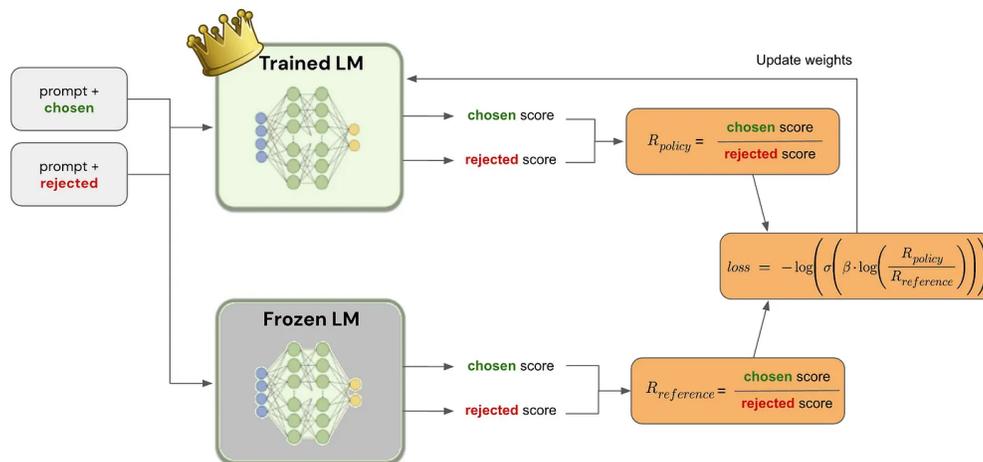


FIGURE 4.7: Ajustement fin du modèle de langage principal en utilisant la DPO [58]

Pour implémenter l'approche de l'Optimisation Directe des Préférences (DPO), les étapes suivantes, illustrées dans la Figure 4.7, sont cruciales :

1. **Préparation du Modèle** : Deux copies du modèle de langage (LM) sont préparées : une avec des poids figés pour servir de référence, et une autre avec des poids entraînaibles pour le fine-tuning.
2. **Évaluation des Réponses** : Chaque réponse générée par le LM est évaluée à l'aide des deux copies du modèle. Les réponses reçoivent un score basé sur la probabilité de produire la réponse attendue.
3. **Calcul des Scores** : Pour un prompt donné, les scores pour les réponses choisies (préférées) et les réponses rejetées sont calculés. Cette évaluation est faite en utilisant une classification binaire simple, où les réponses sont soit acceptées soit rejetées en fonction de leur pertinence.

4. **Calcul de la Perte en DPO** : La perte est calculée en utilisant la fonction de perte DPO, $\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}})$ illustré dans l'équation 4.1. Elle est structurée pour évaluer et ajuster la probabilité que les réponses générées par le modèle soient les préférées, en utilisant les entrées x et les réponses y_w (gagnantes) ou y_l (perdantes). [28]

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right] \quad (4.1)$$

Analyse de l'Équation DPO :

- **Espérance** $-\mathbb{E}$: La valeur attendue est calculée sur un ensemble de données contenant des triplets de (entrée x , résultat gagnant y_w , résultat perdant y_l).
- **Fonction logistique** $\log \sigma$: Cette fonction sigmoïde est utilisée pour convertir les logits en probabilités, assurant que les sorties sont normalisées entre 0 et 1.
- **Logits des politiques** $\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}$ **et** $\log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$: Ces termes comparent les probabilités de générer des réponses préférées et moins préférées selon la nouvelle politique π_θ par rapport à une politique de référence π_{ref} . Le terme β pondère l'importance de ces comparaisons.
- **Divergence KL et préférences** : La divergence de Kullback-Leibler (KL) est utilisée pour mesurer la distance entre les probabilités générées par la politique actuelle et celles de la politique de référence, ajoutant un terme de régularisation qui pénalise les déviations importantes.
- **Objectif** : L'objectif est de minimiser cette perte, ce qui encourage le modèle à favoriser les résultats ayant des récompenses élevées tout en restant proche de la politique de référence, limitant ainsi les modifications radicales qui pourraient dégrader la performance du modèle.

la formule 4.1 illustre l'équilibre entre l'exploration de nouvelles réponses et l'exploitation des réponses qui sont déjà connues pour être efficaces, ce qui est essentiel dans les applications de DPO pour garantir que les modèles restent fiables et alignés avec les préférences humaines. [28]

4.6.4 Retrieval-Augmented Generation (RAG)

La technologie de Génération Augmentée par Récupération (RAG) représente une avancée majeure dans le domaine de l'intelligence artificielle, développée initialement par des chercheurs de Meta AI, il consiste en une fusion innovante entre les modèles de génération de texte et les techniques de récupération d'informations. Cette approche permet aux modèles de langage de tirer parti des vastes bases de données ou corpus pour enrichir leurs réponses, offrant ainsi des réponses plus précises et informées. RAG combine efficacement les capacités de réponse directe d'un modèle de langage avec la richesse des informations disponibles dans des documents externes, ce qui est particulièrement utile dans des applications telles que les chatbots, où la précision et la pertinence de l'information sont cruciales. Cette méthode est essentielle pour améliorer l'interaction des utilisateurs avec les systèmes basés sur l'IA, en fournissant des réponses non seulement cohérentes mais aussi profondément ancrées dans des faits vérifiables et des détails spécifiques.

2.6.4.1 Pourquoi Utiliser la RAG ?

La Génération Augmentée par Récupération (RAG) améliore substantiellement les modèles de langage à grande échelle (LLMs), en les dotant de capacités pour surmonter plusieurs défis intrinsèques :

- **Manque d'informations spécifiques** : Contrairement aux LLMs traditionnels, qui se limitent aux données sur lesquelles ils ont été entraînés, les RAG utilisent des bases de données externes pour fournir des réponses précises et adaptées au contexte.
- **Réduction des hallucinations** : Les LLMs peuvent produire des réponses basées sur des faits non véridiques. Les RAG, en intégrant des sources vérifiables, minimisent ce risque en ancrant les réponses dans la réalité.

- **Amélioration de la pertinence des réponses** : Les RAG enrichissent les capacités génératives des LLMs avec la précision de mécanismes de recherche de données spécialisés, assurant des réponses nuancées et adaptées aux besoins spécifiques des utilisateurs.

Cette méthode transforme l'application des LLMs en les rendant plus aptes à fournir des réponses informées et précises, crucial pour des applications exigeant une grande fiabilité comme les assistants virtuels et les systèmes de support client.

2.6.4.2 Processus d'implémentation des RAG

La Génération Augmentée par Récupération (RAG) est mise en œuvre à travers un processus structuré en plusieurs phases clés :

1. Préparation :

- *Collecte de Données* : Rassemblement des documents nécessaires, comme des manuels d'utilisateur, PDFs, bases de données de produits et FAQs qui seront utilisés pour générer des réponses.

2. Stockage :

- *Découpage et Encodage des Données (Data chunking and Document embeddings)* : Division des documents en segments plus petits (chunks) et transformation de ces textes en représentations vectorielles, ou embeddings, qui permettent de traiter sémantiquement les informations.
- *Stockage des embeddings* : Sauvegarde des embeddings dans une base de données vectorielle pour permettre une récupération rapide et efficace.

3. Récupération :

- *Traitement des Requêtes Utilisateurs* : Conversion des requêtes des utilisateurs en embeddings en utilisant le même modèle que pour les documents.
- *Recherche et Récupération* : Calcul des scores de similarité entre le vecteur de la requête et ceux des chunks, sélectionnant ainsi les segments les plus pertinents.

4. Réponse :

- *Génération de Réponses* : Intégration des informations récupérées et de la requête initiale dans un modèle de langage pour générer une réponse cohérente et contextuellement adaptée.

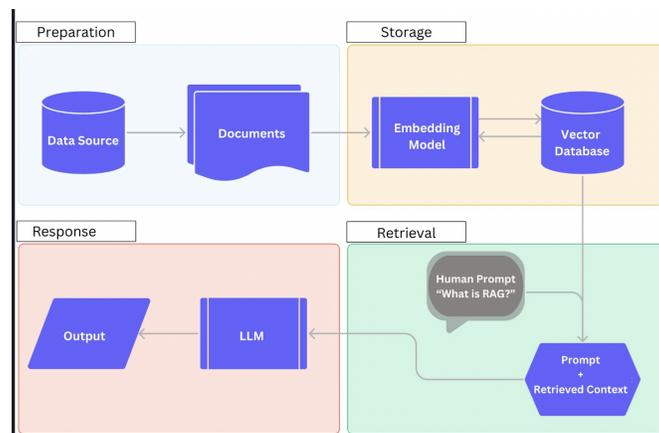


FIGURE 4.8: Processus d'implémentation des RAG [59]

Les étapes décrites ci-dessus et illustrer dans la Figure 4.8 correspondent à l'implémentation de la méthode Naive RAG, qui est l'une des formes de base de la Génération Augmentée par Récupération. Il existe d'autres améliorations et variantes de RAG qui intègrent des techniques plus complexes et des optimisations supplémentaires pour répondre à des besoins spécifiques [33]. Cependant, dans le cadre de ce mémoire, nous avons choisi d'implémenter le Naive RAG, car cette approche correspond le mieux à nos objectifs actuels et aux ressources disponibles.

2.6.4.3 RAG vs. Fine-tuning

Lorsqu'on explore les différences entre la Génération Augmentée par Récupération (RAG) et le fine-tuning, il est essentiel de comprendre dans quels scénarios chaque méthode excelle. Les recherches indiquent que le RAG est particulièrement utile pour intégrer de nouvelles connaissances dans les modèles, alors que le fine-tuning est efficace pour améliorer la performance et l'efficacité des modèles par l'optimisation de la connaissance interne, le format de sortie, et la capacité à suivre des instructions complexes. Ces approches ne sont pas mutuellement exclusives et peuvent se compléter dans un processus itératif visant à améliorer l'utilisation des modèles de langage à grande échelle (LLMs) pour des applications exigeantes en connaissances et évolutives nécessitant un accès à des connaissances rapidement changeantes et des réponses personnalisées qui respectent un format, un ton et un style spécifiques. De plus, l'ingénierie de prompts peut également être utilisée pour optimiser les résultats en exploitant les capacités inhérentes du modèle. la Figure 4.9 illustre les différentes caractéristiques de RAG comparées à d'autres méthodes d'optimisation de modèles :

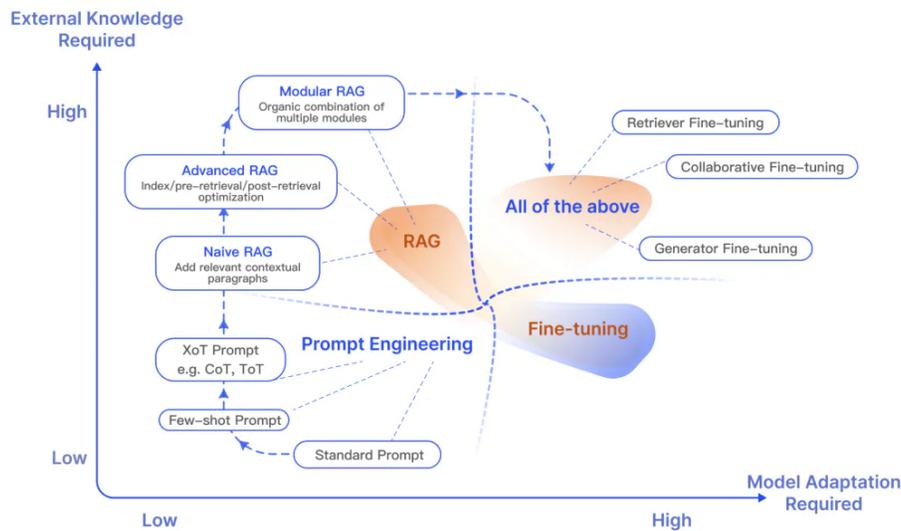


FIGURE 4.9: Comparaison des RAG avec d'autres méthodes d'optimisation [32]

Le tableau 4.1 fournit une analyse comparative détaillée des caractéristiques distinctives entre le méthode des RAG (Retrieval-Augmented Generation) et les modèles soumis à un ajustement fin, mettant en lumière les différences et les similitudes clés entre ces approches :

Caractéristique	RAG	Fine-Tuning
Mises à jour des connaissances	Actualisation directe des données récupérées, adaptée aux environnements dynamiques.	Nécessite une reformation pour les mises à jour, stocke des données statiques.
Connaissance externe	Utilise efficacement les ressources externes pour accéder à des documents et bases de données.	S'aligne avec la connaissance externe acquise par pré-entraînement.
Traitement des données	Minimise le traitement des données grâce à une récupération ciblée.	Dépend de la qualité des datasets pour l'amélioration significative de la performance.
Personnalisation du modèle	Intègre des connaissances externes mais peut ne pas personnaliser entièrement le comportement ou le style du modèle.	Permet des ajustements spécifiques au comportement du LLM, style d'écriture, et connaissances domaines.
Interprétabilité	Haute interprétabilité grâce à la traçabilité des sources de données.	Moindre interprétabilité, fonctionne comme une "boîte noire".
Ressources computationnelles	Dépend des ressources pour soutenir les stratégies de récupération et nécessite la maintenance de la source de données.	Nécessite des ressources computationnelles pour l'entraînement de haute qualité et la curation des données.
Exigences de latence	Peut entraîner une latence accrue due à la récupération des données.	Répond sans latence après le fine-tuning.
Réduction des hallucinations	Moins sujet aux hallucinations car les réponses sont basées sur des preuves concrètes.	Peut réduire les hallucinations par un entraînement ciblé sur des données spécifiques.
Questions éthiques et de confidentialité	Soucis éthiques liés à la récupération de texte depuis des bases de données externes.	Des préoccupations peuvent survenir en raison de la sensibilité du contenu des données d'entraînement.

TABLE 4.1: Tableau comparant les caractéristiques entre les modèles RAG et les modèles ajustés finement [32]

4.7 Évaluation des LLMs

L'évaluation des modèles de langage (LLMs) [63], [64] est une étape cruciale pour mesurer la qualité et la pertinence des résultats produits par le modèle. Quelle que soit la méthode d'évaluation utilisée, elle peut être effectuée de manière subjective en demandant à des experts de noter la qualité des réponses ou de manière objective en utilisant des mesures telles que la précision, la cohérence, la lisibilité, la pertinence, la perplexité, la robustesse, et la variabilité. Ces résultats sont utilisés pour ajuster le modèle et améliorer sa performance.

4.7.1 Types d'Évaluations

- **Évaluation Humaine** : Dans cette technique, un groupe de personnes évalue la qualité du texte généré par le modèle. Les évaluateurs humains peuvent être des experts du domaine ou des personnes ordinaires. Les évaluations sont généralement basées sur des critères tels que la clarté, la cohérence, la pertinence et la fluidité du texte.
- **LLM-comme-Juge** : Cette méthode utilise un autre LLM pour évaluer les sorties du modèle testé [63]. Cette approche a été trouvée pour refléter largement les préférences humaines pour certains

cas d'usage.

- **Évaluation Automatique** : Les métriques automatiques peuvent être utilisées pour évaluer la qualité du texte généré. Les métriques d'évaluation automatique les plus courantes incluent la précision, le rappel, et la perplexité. Ces métriques sont souvent utilisées pour évaluer des aspects spécifiques de la génération de textes, tels que la précision de la grammaire ou la cohérence thématique.

4.7.2 Métriques d'évaluation automatique

Pour évaluer les performances des LLMs il existe plusieurs métriques d'évaluation, dans cette partie on va présenter les métriques les plus utilisées.

2.7.2.1 ROUGE Score

Le ROUGE Score, ou Recall-Oriented Understudy for Gisting Evaluation, est un ensemble de métriques utilisées pour évaluer la qualité des modèles de traduction et de résumé de documents. Il calcule le F1-score basé sur le nombre de mots consécutifs communs entre le texte de référence et le texte généré. Le score varie de 0 à 1, un score proche de zéro indiquant une faible similarité entre le candidat et les références, et un score proche de un indiquant une forte similarité.

Le ROUGE Score est basé sur le concept des n-grams, qui sont des séquences de n mots. Par exemple, un 1-gram est un seul mot, un 2-gram est une paire de mots, etc.

Les différents types de métriques ROUGE incluent :

- ROUGE-N : Mesure le chevauchement des n-grams entre les résumés système et de référence. Par exemple, ROUGE-1 se réfère au chevauchement des unigrammes (chaque mot), tandis que ROUGE-2 se réfère au chevauchement des bigrammes (deux mots consécutifs).
- ROUGE-L : Basé sur la longueur de la plus longue sous-séquence commune (LCS). Il calcule la moyenne harmonique pondérée combinant le score de précision et le score de rappel. Il ne nécessite pas de correspondances consécutives mais des correspondances en séquence.
- ROUGE-W : Une statistique basée sur LCS pondérée qui favorise les LCS consécutifs.
- ROUGE-S : Une statistique de co-occurrence basée sur les skip-bigrammes. Un skip-bigramme est toute paire de mots dans l'ordre de la phrase.
- ROUGE-SU : Une statistique de co-occurrence basée sur les skip-bigrammes plus les unigrammes.

Formules Mathématiques des ROUGE scores

Les formules suivantes (4.2),(4.3),(4.4),(4.5),(4.6),(4.7),(4.8),(4.9),(4.10) sont utilisées pour calculer les scores ROUGE :

- **Rouge 1**

$$\text{Rouge-1}_p = \frac{\text{nombre des unigrammes superposés}}{\text{nombre total des unigrammes dans le texte généré}} \quad (4.2)$$

$$\text{Rouge-1}_r = \frac{\text{nombre des unigrammes superposés}}{\text{nombre total des unigrammes dans le texte de référence}} \quad (4.3)$$

$$\text{Rouge-1}_f = \frac{2(\text{Rouge-1}_p \text{Rouge-1}_r)}{\text{Rouge-1}_p + \text{Rouge-1}_r} \quad (4.4)$$

- **Rouge 2**

$$\text{Rouge-2}_p = \frac{\text{nombre des bigrammes superposés}}{\text{nombre total des bigrammes dans le texte généré}} \quad (4.5)$$

$$\text{Rouge-2}_r = \frac{\text{nombre des bigrammes superposés}}{\text{nombre total des bigrammes dans le texte de référence}} \quad (4.6)$$

$$\text{Rouge-2}_f = \frac{2(\text{Rouge-2}_p \text{Rouge-2}_r)}{\text{Rouge-2}_p + \text{Rouge-2}_r} \quad (4.7)$$

— Rouge L

$$\text{Rouge-L}_p = \frac{\text{sous-séquence commune la plus longue (LCS)}}{\text{nombre total des n-grams dans le texte généré}} \quad (4.8)$$

$$\text{Rouge-L}_r = \frac{\text{sous-séquence commune la plus longue (LCS)}}{\text{nombre total des n-grams dans le texte de référence}} \quad (4.9)$$

$$\text{Rouge-L}_f = \frac{2(\text{Rouge-L}_p \text{Rouge-L}_r)}{\text{Rouge-L}_p + \text{Rouge-L}_r} \quad (4.10)$$

2.7.2.2 Evaluation de l'allucination du modèle

Utilisation du modèle vectara/hallucination_evaluation_model

Le modèle HHEM [65] est un modèle open source créé par Vectara, conçu pour détecter les hallucinations dans les LLMs. Il est particulièrement utile dans le contexte de la création d'applications de génération augmentée par récupération (RAG), où un ensemble de faits est résumé par un LLM, mais le modèle peut également être utilisé dans d'autres contextes.

Ce modèle a été entraîné en utilisant la classe Cross-Encoder de SentenceTransformers. Il produit une probabilité de 0 à 1, 0 représentant une hallucination et 1 représentant une cohérence factuelle. Les prédictions peuvent être seuilées à 0.5 pour déterminer si un document est cohérent avec sa source.

Données d'Entraînement

Ce modèle est basé sur *microsoft/deberta-v3-base* et a été initialement entraîné sur des données NLI (Natural Language Inference) pour déterminer l'entaillement textuel. Ensuite, il a été affiné sur des ensembles de données de résumés avec des échantillons annotés pour la cohérence factuelle, y compris FEVER, Vitamin C et PAWS.

4.8 Conclusion

Dans ce chapitre, nous avons exploré en détail les Grands Modèles de Langage (LLMs), en abordant leur architecture, leurs applications, ainsi que les différentes approches d'amélioration des performances des LLMs. Nous avons étudié des méthodes allant du fine-tuning à l'apprentissage par renforcement avec feedback humain, et enfin à la Génération Augmentée par Récupération (RAG). De plus, nous avons examiné les diverses méthodes d'évaluation des résultats des LLMs. Cette analyse approfondie nous permet de comprendre comment les LLMs fonctionnent, comment ils peuvent être optimisés pour diverses tâches linguistiques et comment évaluer efficacement leur performance.

Dans le chapitre suivant, nous explorerons la solution proposée pour relever les défis présentes dans la problématique et améliorer le processus de l'offre de KPMG dans le secteur bancaire. Notre objectif est de proposer une solution intégrant l'intelligence artificielle générative avec l'utilisation de données internes à KPMG, permettant des réponses à la fois rapides et pertinentes.

Troisième partie

Conception de la solution

Chapitre 5

Présentation de la solution

5.1 Introduction

Face au besoin de l'entreprise d'améliorer le service client et d'attirer davantage de clients bancaires, notre solution est basée sur le développement d'un chatbot, conçu pour répondre automatiquement aux clients de KPMG et proposer des solutions étape par étape à leurs problèmes, en utilisant des techniques avancées d'IA générative. La conception de ce chatbot repose sur une analyse approfondie de diverses méthodes pour identifier celle qui correspond le mieux à nos besoins et objectifs, ainsi qu'à nos ressources disponibles.

Ce chapitre se concentre sur la mise en œuvre concrète de la solution développée dans ce mémoire, en commençant par une présentation des étapes du développement de notre solution puis une explication en détail de chaque étape, en mettant l'accent sur les choix méthodologiques, les techniques utilisées et les résultats obtenus.

5.2 Structure de la solution

Cette section présente la structure de notre solution ainsi que le pipeline proposé pour sa réalisation, en détaillant les différentes approches et méthodes envisagées. La Figure [5.1](#) illustre schématiquement les étapes suivies pour notre solution.

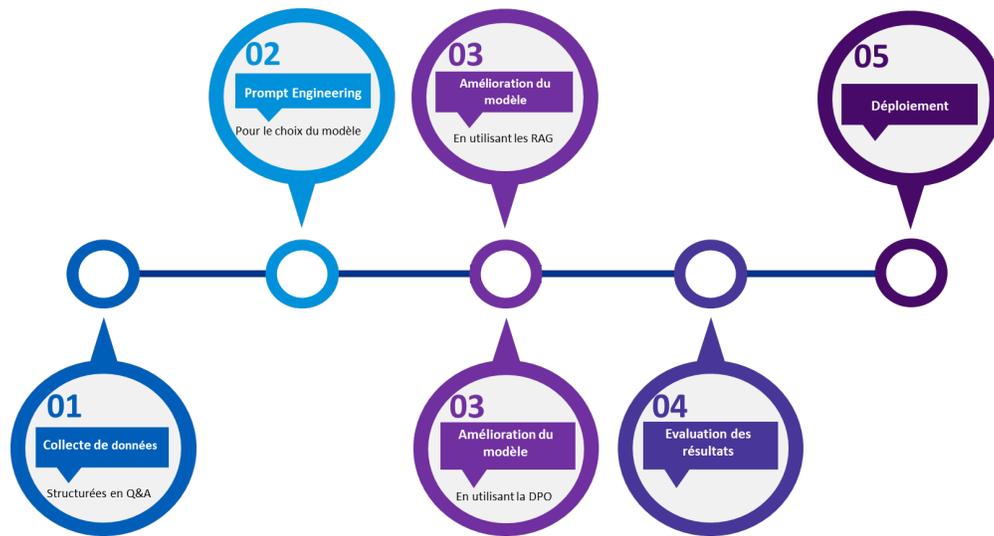


FIGURE 5.1: Schématisation des étapes de la solution

Pour construire notre chatbot, nous nous sommes appuyés sur un modèle de langage à grande échelle (LLM) pré-entraîné, en améliorant ses capacités pour répondre spécifiquement à nos besoins grâce à des techniques avancées d'IA générative. Voici les étapes clés de la mise en œuvre de notre solution :

1. **Collecte de données** : Initialement, nous avons entrepris de rassembler un ensemble de données spécifiquement adapté à nos objectifs. Cela impliquait d'obtenir des exemples de questions posées par les clients bancaires de KPMG et les réponses attendues que notre chatbot devrait générer.
2. **Choix du LLM** : Face à l'abondance des LLMs performants disponibles, choisir le bon modèle adapté à nos données et besoins était complexe. Nous avons utilisé la méthode du prompt engineering pour tester plusieurs LLMs sur un échantillon de nos questions/réponses, sélectionnant ainsi les modèles les plus performants pour notre projet.
3. **Choix de la méthode d'amélioration du LLM** : Nous avons opté pour deux approches reconnues pour leur efficacité et leur faisabilité avec nos ressources. La première méthode consiste à entraîner le LLM en utilisant Direct Preference Optimization (DPO) avec notre base de données pour qu'il apprenne à générer des données similaires. La seconde méthode intègre Retrieval-Augmented Generation (RAG) pour permettre au LLM d'extraire et d'utiliser des données pertinentes dans ses réponses.
4. **Évaluation des résultats** : Après l'implémentation des méthodes, une évaluation rigoureuse et une interprétation des résultats sont essentielles pour choisir l'approche la plus adaptée.
5. **Déploiement de la solution** : En fonction de la méthode qui a donné les meilleur performance, nous l'utiliseront pour finalisé notre solution. Le chatbot sera alors déployé dans un environnement de production pour utilisation dans des scénarios réels.

5.3 Conception de la solution

Dans cette section, nous allons présenter en détail les étapes que nous avons suivies pour concrétiser notre solution, les résultats obtenus, ainsi que le déploiement de la solution.

5.3.1 Collecte de données

Nous avons commencé par la collecte des données nécessaires à la conception de notre solution. A cet effet, la structure de données la plus appropriée pour notre cas est une base de données composée de questions-réponses.

5.3.1.1 Formats de la Base de Données

La méthode DPO (Direct Preference Optimization) requiert un format de données comprenant des questions, des réponses attendues et des réponses rejetées.

- **Questions** : Ce sont des cas réels de problématiques proposées par des clients.
- **Réponse attendue** : Il s'agit de la solution proposée par le cabinet KPMG au client. Étant donné que le chatbot ne remplacera pas un consultant, nous avons structuré les réponses comme étant les étapes de mise en œuvre de la solution, en mentionnant les outils et les technologies utilisés.
- **Réponse rejetée** : Ce sont des réponses de mauvaise qualité par rapport aux réponses issues des cas réels de notre base de données. Une explication détaillée sur la collecte de ces réponses sera fournie ultérieurement.

Pour les RAG (Retrieval-Augmented Generation), nous utilisons les mêmes données, à l'exception des réponses rejetées. La base de données se compose d'exemples de questions que les clients peuvent poser et des réponses conçues à partir des cas réels de missions.

5.3.1.2 Collecte de données

Pour collecter les données nécessaires à l'implémentation de nos méthodes, nous avons adopté plusieurs approches :

— Données Internes

Ce sont des données internes à KPMG. Pour les collecter, nous avons lancé un formulaire à destination des consultants de KPMG ayant travaillé sur des missions avec des clients du secteur bancaire. Nous leur avons expliqué notre projet et le format des solutions proposées, sous forme d'étapes en mentionnant les technologies utilisées. Cela représente la valeur ajoutée de notre chatbot par rapport à une solution proposée par un modèle de langage pré-entraîné qui n'a pas été entraîné avec des données spécifiques à KPMG.

— Données Externes

Étant donné que les données internes collectées chez KPMG n'étaient pas suffisantes, nous avons décidé de collecter des données externes adaptées à notre cas. Nous avons consulté les sites web des cabinets de conseil dans le domaine bancaire et trouvé des cabinets publiant des résumés de missions accomplies. Les résumés contenaient la présentation du problème du client qu'on a reformulé sous forme d'une question que le client pourrait poser au chatbot, ainsi que le résumé de la solution proposée en étapes avec les outils utilisés qu'on a reformulé comme une réponse dans un format proposé par le chatbot.

— Génération des réponses rejetées de la base de données

Pour produire les réponses rejetées de notre base de données, nous avons initialement opté pour la génération des réponses à l'aide d'un petit modèle mais les réponses étaient de très mauvaise qualité donc inutilisables même comme mauvaises réponses, et aussi la majorité des modèles avec un petit nombre de paramètres ont une limite dans la taille du texte généré, ce qui a entraîné l'obtention de réponses incomplètes.

Pour cela nous avons opté pour une deuxième méthode qui est la génération des réponses avec GPT-3.5 Turbo, nous avons passé les questions de notre base de données à GPT-3.5 Turbo en lui demandant de générer des réponses de mauvaises qualités suite à ces questions.

A la fin, nous avons consolidé les données collectées dans une base de données pour entamer la phase suivante, à savoir le prompt engineering, afin de choisir le meilleur modèle pour concrétiser notre solution.

5.3.2 Prompt Engineering

5.3.2.1 Pourquoi utiliser le prompt engineering ?

Nous avons choisi d'adopter le prompt engineering pour diverses raisons fondamentales :

- Évaluation des performances des modèles : Cette méthode nous permet de tester les capacités des modèles sur nos données, facilitant ainsi l'évaluation de leur efficacité et aidant à choisir le modèle le plus adapté pour nos méthodes futures.
- Absence de nécessité d'entraînement : Le prompt engineering ne requiert pas un entraînement préalable du modèle, ce qui réduit significativement le temps et les ressources nécessaires pour obtenir des résultats.
- Facilité d'implémentation : Cette approche est relativement simple à mettre en place, permettant une intégration rapide dans nos processus sans exigences techniques complexes.
- Réduction des coûts : En évitant les longs processus d'entraînement et les ressources computationnelles intensives, le prompt engineering contribue à une diminution notable des coûts associés au développement et à la maintenance des modèles de langage.

5.3.2.2 Modèles utilisés

Dans cette étude, plusieurs modèles de traitement du langage ont été évalués. Chacun présente des caractéristiques distinctes qui influencent leur performance dans des tâches spécifiques de compréhension et de génération de texte. Voici une brève description de chaque modèle utilisé :

- **Llama3 8B** : Proposé par Meta AI, ce modèle de 8 milliards de paramètres utilise une architecture transformer. Il est entraîné sur un large corpus de textes provenant du web et supporte une fenêtre contextuelle typique des modèles GPT (jusqu'à 8,192 tokens), et maîtrise plusieurs langues, incluant l'anglais, le français et l'espagnol [66].
- **Llama3 70B** : Version plus grande du Llama3, développée par Meta AI, avec 70 milliards de paramètres, conçue pour des tâches complexes. Ce modèle utilise une fenêtre contextuelle étendue à 8,192 tokens et maîtrise les mêmes langues que la version 8B [67].
- **Mistral-7B-Instruct** : Développé par Mistral AI, ce modèle de 7 milliards de paramètres est optimisé pour suivre des instructions précises, utilisant une architecture transformer. Il est entraîné sur des données variées et supporte une fenêtre contextuelle de 4096 tokens, en gérant principalement l'anglais [68].
- **Mixtral-8x7B** : Une innovation de Mistral AI, ce modèle est composé de 8 modèles experts et d'un modèle de routage, totalisant 45 milliards de paramètres. Cependant, seulement 12 milliards de paramètres sont activés pour le traitement de chaque token, ce qui permet de réduire la charge de calcul tout en bénéficiant de l'expertise spécialisée de chaque sous-modèle. Le Mixtral-8x7B supporte une fenêtre contextuelle de 32k tokens. Il maîtrise plusieurs langues, dont l'anglais, le français, l'italien, l'allemand et l'espagnol [71].
- **Gemma 7B** : Créé par Google, ce modèle de 7 milliards de paramètres est conçu pour générer des réponses informatives et détaillées. Il est entraîné sur un mélange de textes académiques et techniques, et supporte une fenêtre contextuelle de 8,192 tokens aussi [69].
- **Falcon 7B** : Développé par TII UAE, ce modèle utilise une architecture transformer pour offrir des réponses contextuellement appropriées. Il est spécifiquement entraîné pour interagir en utilisant des dialogues humain-machine dans diverse langue : Anglais, allemand, espagnol, français (et capacités limitées en italien, portugais, arabe...) [70].

Ces descriptions offrent un aperçu des capacités et des objectifs de conception de chaque modèle, fournissant un contexte pour comprendre leurs performances variées dans notre analyse comparative.

5.3.2.3 Application du Few-Shot Learning

Pour pratiquer le prompt engineering efficacement, il est essentiel de pouvoir interagir avec des modèles de langage sans nécessiter l'accès à leurs poids internes. Nous avons opté pour deux approches distinctes :

- Utilisation de la clé API de HuggingFace : Cette plateforme héberge une quantité impressionnante de modèles de langage en open source. Nous pouvons soit télécharger les modèles pour une utilisation locale, soit grâce à une clé API, les appeler directement sans nécessiter un téléchargement complet.

Cela permet de tester leurs performances et d'envoyer des prompts directement, bien que l'accès à toutes les capacités du modèle puisse être restreint. Des notions plus détaillées sur HuggingFace seront explorées dans l'Annexe A 5.5. Les modèles que nous avons utilisés via Hugging Face incluent : **Mistral 7b-instruct, Falcon 7b-instruct.**

- Utilisation des modèles hébergés sur Groq Cloud : Groq Cloud est une plateforme d'inférence AI développée par Groq, spécialement conçue pour les modèles de traitement de langage volumineux, offrant une interface rapide et des performances de calcul optimisées pour les besoins en temps réel. Pour utiliser les modèles sur Groq Cloud, il faut d'abord obtenir une clé API via la console de développeur de Groq. Ensuite, télécharger la bibliothèque Groq pour Python, puis utiliser cette bibliothèque pour effectuer des appels API et interagir avec les modèles disponibles sur la plateforme. Des notions plus détaillées sur Groq Cloud seront explorées dans l'Annexe A 5.5. Dans le cadre de notre projet, nous avons utilisé sur cette plateforme les modèles suivants : **Lama3 70B, Mixtral 8x7B, Gemma 7B et Lama3 8b.**

Après avoir appelé un modèle, la prochaine étape consiste à configurer un apprentissage par quelques exemples (few-shot learning) pour guider le modèle dans la génération de réponses. On procède comme suit :

1. **Création de l'ensemble d'exemples (Examples Set-up)** : On définit une liste d'exemples qui contient deux dictionnaires. Chaque dictionnaire représente un exemple qui comprend une question posée par un professionnel bancaire et la réponse correspondante. Ces exemples sont destinés à guider le modèle sur le type de réponse attendue pour des questions similaires.
Le nombre d'exemples dans cette liste détermine le type de few-shot learning utilisé par exemple un seul exemple constitue un one-shot learning, tandis que cinq exemples forment un five-shot learning.
2. **Définition du Modèle de Prompt** : Un modèle de prompt '*PromptTemplate*' est créé, qui prend en paramètres '*input-variables*' (ici, "question" et "Réponse") et un template de formatage. Ce template organise comment le prompt (Question) et la réponse (Réponse) doivent être formatés lors de leur utilisation.
3. **Configuration du Few-Shot Prompt Template** : On instancie un '*FewShotPromptTemplate*' en utilisant la liste d'exemples et le modèle de prompt. Ce template prépare le formatage des exemples pour aider le modèle à comprendre le contexte et la structure attendue des réponses.
4. **Formation d'une Chaîne d'Invocation** : On forme une chaîne d'invocation en liant le template '*FewShotPromptTemplate*' à un processus de génération de réponses. Cette chaîne est utilisée pour invoquer le modèle avec une nouvelle entrée, en se basant sur les exemples formatés pour guider la réponse du modèle.
5. **Affichage du Résultat** : On extrait et affiche directement le résultat. Cela montre comment le modèle, guidé par les exemples fournis, répond à de nouvelles questions dans un domaine spécifique.

5.3.2.4 Évaluation des résultats

a) Métriques utilisées

Notre évaluation se concentre principalement sur le score ROUGE-1 F pour sa capacité à évaluer la précision et le rappel des unigrammes. En complément, nous utilisons l'ensemble des scores ROUGE (ROUGE-1, ROUGE-2, et ROUGE-L) pour une analyse plus complète des performances, en mettant en lumière la précision, le rappel, et les F1-scores des unigrammes, bigrammes, et des séquences les plus longues. Cela nous permet de mieux comprendre la capacité des modèles à générer des réponses contextuellement appropriées.

b) Configuration des Arguments

Après plusieurs tests, nous avons fixé les paramètres de chaque modèle de la manière suivante : `max_length=5000`, `temperature=0.4`. Ces paramètres ont été choisis pour optimiser la génération des réponses.

c) Méthodologie d'Évaluation

Pour réaliser une évaluation complète et comparative des différents modèles, nous avons adopté l'approche suivante :

1. Évaluation Répétée des Réponses du Modèle :
 - Générer trois réponses du modèle pour chaque prompt.
 - Calculer le score ROUGE pour chaque réponse générée.
 - Déterminer le score ROUGE moyen à partir des trois réponses.
2. Évaluation des Réponses du Modèle sur Plusieurs Questions :
 - Sélectionner trois questions aléatoires de notre base de données qui n'ont pas été utilisées comme exemples dans les shots du modèle.
 - Calculer le score ROUGE moyen pour chaque question en utilisant la méthode précédente.
3. Évaluation des Réponses du Modèle avec Variation du Nombre de Shots :
 - Varier le nombre de shots de 0 à 5.
 - Évaluer les performances du modèle sur les trois questions à chaque niveau de shots.
4. Synthèse des Résultats :
 - Compiler un tableau récapitulatif affichant le score ROUGE moyen pour chaque question à différents niveaux de shots, permettant une comparaison directe entre les modèles.

Cette méthodologie nous permet de mesurer de manière rigoureuse l'efficacité de chaque modèle à apprendre et reproduire la structure des réponses en fonction du nombre d'exemples fournis (shots), tout en vérifiant la consistance des réponses générées face à des questions nouvelles et variées.

d) Évaluation des modèles

1. Llama3 8b :

- Le modèle est performant même avec zero shot, se comportant comme un assistant.
- Dès un shot, le modèle adopte la structure de réponse attendue, mentionnant explicitement qu'il s'agit d'un chatbot de KPMG et en incorporant des phrases standardisées en introduction et conclusion.
- Après trois shots, le modèle peut devenir trop précis, répondant parfois à toutes les questions passées en shots au lieu de se concentrer uniquement sur la question posée.
- À cinq shots, le modèle présente des signes d'hallucination, où il génère des réponses incohérentes ou hors contexte.



FIGURE 5.2: Graphiques du modèle Llama3 8B

La Figure 5.2 présente deux graphiques représentant l'évaluation des résultats des réponses générées par le modèle Llama3 8b. L'interprétation de ces graphiques permet de remarquer les aspects suivants :

- L'augmentation notable des scores ROUGE-1 F entre zéro et un shot (de 0.21 à 0.28 pour les valeurs les plus élevées) montre que le modèle intègre bien les exemples fournis, améliorant ainsi la pertinence de ses réponses.
- Avec zéro shot, le modèle montre des résultats modestes, environ 0.2, mais atteint un pic à deux shots, suggérant un équilibre optimal entre apprentissage et précision.
- Les scores ROUGE-1 F augmentent initialement avec le nombre de shots, mais commencent à diminuer après deux shots, ce qui coïncide avec le début des hallucinations observées à partir de trois shots.
- La variation des scores ROUGE-1 R entre les questions dans le deuxième graphe pourrait indiquer une sensibilité du modèle aux types de questions, tandis que la stabilité observée pour les autres métriques suggère une uniformité dans la gestion des bigrammes et des séquences plus longues.

2. Llama3-70b :

- Le modèle est relativement performant même avec zero shot, se comportant comme un assistant.
- Dès un shot, le modèle adopte clairement la structure de réponse souhaitée, mentionnant qu'il agit comme un chatbot de KPMG et intégrant des phrases standardisées en introduction et en conclusion.
- À mesure que le nombre de shots augmente, les réponses deviennent plus précises, mieux structurées et contiennent plus de détails.

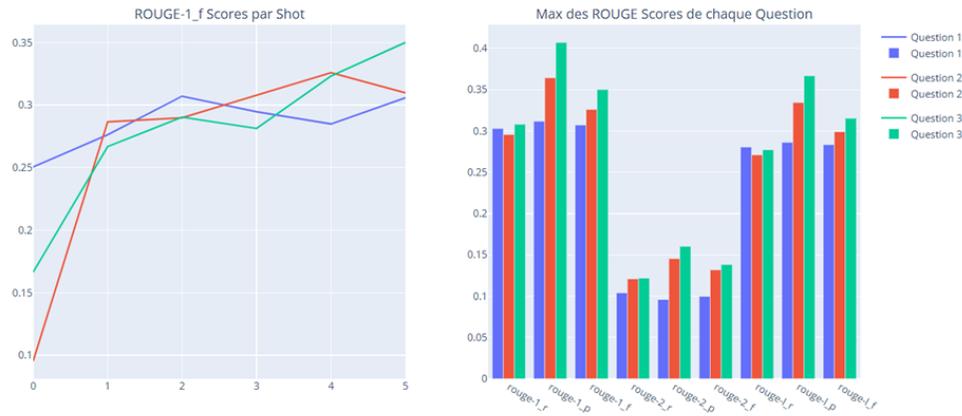


FIGURE 5.3: Graphiques du modèle Llama3 70B

La Figure 5.3 présente deux graphiques représentant l'évaluation des résultats des réponses générées par le modèle Llama3 70b. L'interprétation de ces graphiques permet de remarquer les aspects suivants :

- Les scores ROUGE en zero shot sont relativement bas et varient significativement entre les questions avec des scores initiaux de 0.09, 0.16, et 0.25 respectivement pour les trois questions. Cela indique une adaptation initiale moins efficace aux requêtes sans exemples préalables.
- Il y a une augmentation notable des scores ROUGE-1 F entre zéro et un shot, ce qui démontre une amélioration significative de la qualité des réponses lorsque le modèle est guidé par au moins un exemple.
- Après un shot, les scores montrent une certaine stabilité, avec des valeurs fluctuantes entre 0.28 et 0.35 pour les trois questions. Cette stabilité suggère que le modèle atteint un plateau de performance où des shots additionnels n'apportent pas de bénéfice significatif.
- Les variations des scores ROUGE-1 P entre les questions, alors que les autres scores ROUGE restent stables, pourraient indiquer que le modèle gère de manière uniforme la structure des réponses et le rappel des informations, mais que la précision des termes spécifiques varie selon le contexte de la question.

3. Mixtral-8x7b :

- Le modèle est efficace même avec zero shot, offrant des réponses concrètes.
- Dès un shot, le modèle adopte la structure de réponse souhaitée en intégrant des phrases standardisées en introduction et en conclusion.
- À mesure que le nombre de shots augmente, les réponses deviennent plus précises et mieux structurées.

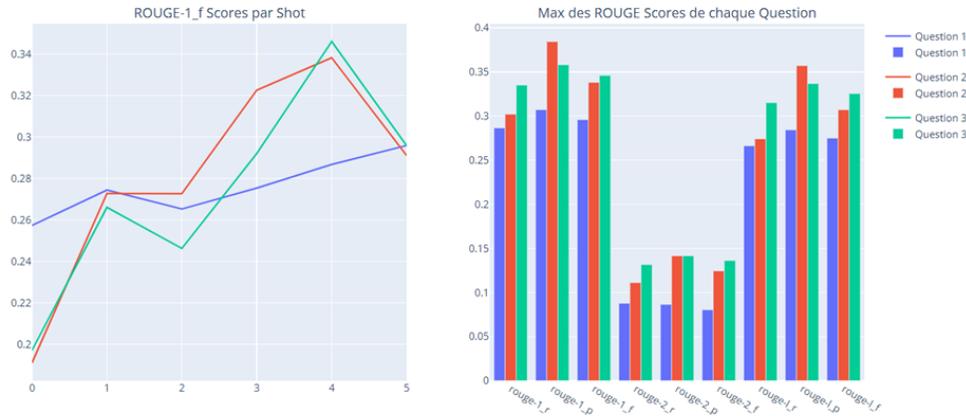


FIGURE 5.4: Graphiques du modèle Mixtral-8x7b

La Figure 5.4 présente deux graphiques représentant l'évaluation des résultats des réponses générées par le modèle Mixtral-8x7b. L'interprétation de ces graphiques permet de remarquer les aspects suivants :

- En zero shot, le modèle affiche des résultats modestes, avec des scores respectivement de 0.19, 0.20, et 0.25 pour les trois questions.
- Une augmentation significative des scores est observée pour les questions 2 et 3, avec un pic de performance à quatre shots, atteignant des scores de 0.33 pour la question 2 et 0.34 pour la question 3, avant de diminuer légèrement à cinq shots.
- La question 1 diverge par rapport aux autres, avec des performances inférieures dès un shot, indiquant peut-être une moindre adaptation du modèle à cette question spécifique.
- Les variations entre les scores ROUGE (ROUGE-1 R, P, et F) dans le deuxième graphique révèlent des différences de performance entre les questions, ce qui suggère des variations dans la précision des termes spécifiques utilisés et dans la manière dont le modèle capture les informations essentielles des prompts.

4. Gemma 7b :

- En zero shot, le modèle génère des réponses longues et bien structurées, mais celles-ci ne répondent pas à l'attente d'un comportement typique d'assistant, ce qui pourrait indiquer une compréhension insuffisante du contexte.
- Après deux shots, bien que les réponses ne se comportent toujours pas comme celles d'un assistant, le modèle commence à incorporer des phrases des exemples fournis, indiquant une amélioration dans l'apprentissage de la structure des réponses.
- À partir de trois shots, le modèle montre des signes d'hallucination, répondant à toutes les questions fournies en shot simultanément sans respecter la structure de réponse souhaitée.

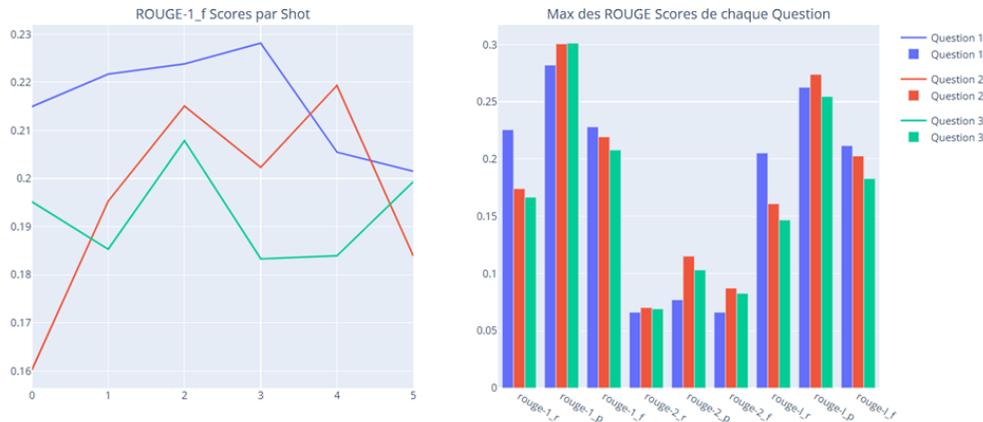


FIGURE 5.5: Graphiques du modèle Gemma 7b

La Figure 5.5 présente deux graphiques représentant l'évaluation des résultats des réponses générées par le modèle Gemma 7b. L'interprétation de ces graphiques permet de remarquer les aspects suivants :

- Le graphique "ROUGE-1 F Scores par Shot" montre une fluctuation des scores au fil des shots. Initialement, les scores commencent relativement bas (0.16 à 0.21), avec une augmentation notable pour la question 2 vers le shot 2, atteignant environ 0.21, ce qui peut refléter une meilleure adaptation du modèle à cette question avec l'introduction d'exemples.
- Cependant, les scores pour les questions 1 et 3 restent inférieurs et présentent une variabilité plus marquée, avec une tendance à la baisse pour la question 3 à partir du troisième shot, suggérant des difficultés dans la gestion de cette question en particulier.
- Le graphique "Max des ROUGE Scores de chaque Question" révèle que les scores maximaux pour chaque métrique ROUGE varient significativement entre les questions. Pour la question 3, les scores sont généralement inférieurs, surtout en ROUGE-1 F, ROUGE-2 F, et ROUGE-L F, indiquant une capacité réduite à reproduire des réponses fidèles et structurées pour cette question.

5. Mistral-7B-Instruct :

- Zero shot : Le modèle est capable de générer des réponses pertinentes d'emblée, démontrant une compréhension de base des questions même sans exemples préalables.
- Un shot : Des réponses excessivement longues et parfois erronées sont observées, suggérant des premiers signes d'hallucination.
- Deux shots : Le modèle adopte la structure de réponse souhaitée avec efficacité, bien que les réponses entre les différentes questions commencent à se ressembler notablement.
- Trois shots : Stabilisation des réponses qui continuent à ressembler à celles obtenues avec deux shots, indiquant une optimisation effective.
- Quatre et cinq shots : Amélioration continue avec des réponses plus détaillées, cependant, à cinq shots pour la question 2, on observe une régression due à des hallucinations.



FIGURE 5.6: Graphiques du modèle Mistral-7B-Instruct

La Figure 5.6 présente deux graphiques représentant l'évaluation des résultats des réponses générées par le modèle Mistral-7B-Instruct. L'interprétation de ces graphiques permet de remarquer les aspects suivants :

- Le graphique ROUGE-1 F Scores par Shot montre une amélioration initiale des performances pour toutes les questions jusqu'à deux shots, suivie par une stabilisation ou une légère augmentation. La performance pour la question 1 fluctue notablement, avec un pic à 2 shots, indiquant un ajustement initial puis une diminution potentielle due à des hallucinations.
- Le graphique Max des ROUGE Scores de chaque question illustre une hétérogénéité dans les performances maximales selon les différentes métriques ROUGE. Notamment, pour la question 1 qui montre des variations plus larges, révélant des réponses plus détaillées pour cette question spécifique.

6. Falcon-7b-Instruct :

- Zero shot : Le modèle génère des réponses aléatoires avec des signes clairs d'hallucination, indiquant une difficulté à générer des réponses pertinentes sans exemples préalables.
- Un shot : Bien que le modèle intègre certaines phrases des exemples fournis, les hallucinations persistent, compromettant la cohérence des réponses.
- Deux shots : Le modèle commence à montrer des signes de détérioration plus sévère, ne finissant pas ses réponses et continuant à se détériorer au-delà de ce point.

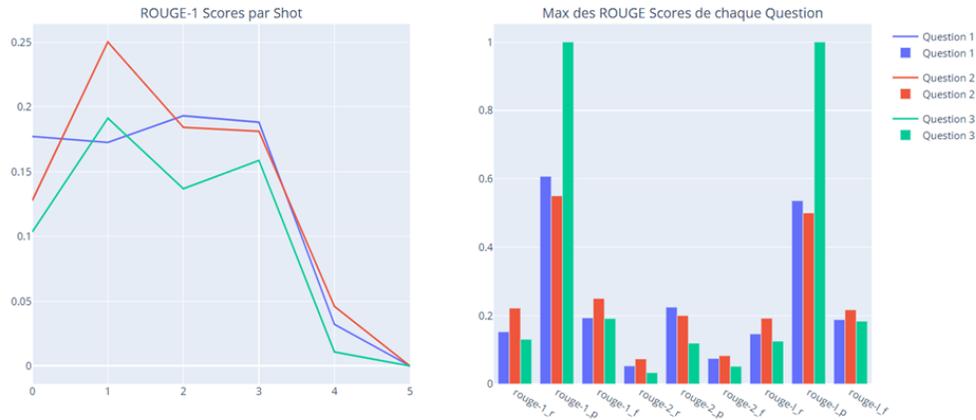


FIGURE 5.7: Graphiques du modèle Falcon-7b-Instruct

La Figure 5.7 présente deux graphiques représentant l'évaluation des résultats des réponses générées par le modèle Falcon-7b-Instruct. L'interprétation de ces graphiques permet de remarquer les aspects suivants :

- Les scores ROUGE-1 F commencent à un niveau relativement bas en zero shot, avec une légère augmentation après un shot, mais commencent à chuter drastiquement après deux shots. Cette chute continue jusqu'à atteindre des valeurs proches de zéro à cinq shots, illustrant une incapacité croissante du modèle à fournir des réponses cohérentes.
- Le deuxième graphique montre une performance maximale très disparate. Les scores maximaux pour ROUGE-1 F et ROUGE-L F pour la question 1 sont exceptionnellement élevés comparés aux autres métriques, ce qui pourrait suggérer des cas spécifiques où le modèle réussit à bien répondre malgré une tendance générale à l'échec. Pour les autres questions, les scores sont beaucoup plus faibles, indiquant une performance généralement médiocre sur ces types de questions.
- Le pic à 1 dans le score ROUGE-P observé pour la question 3 est dû au fait que le modèle n'a généré aucun mot à cinq shots, résultant en un calcul de précision qui, techniquement, atteint 100% puisque aucune erreur de précision n'a été commise, bien que cette situation soit plus indicative d'une anomalie de calcul plutôt que d'une performance effective du modèle.

e) Comparaison entre les résultats des modèles

Moyenne du ROUGE-1 F des Questions pour Chaque Modèle en Fonction du Nombre de Shots

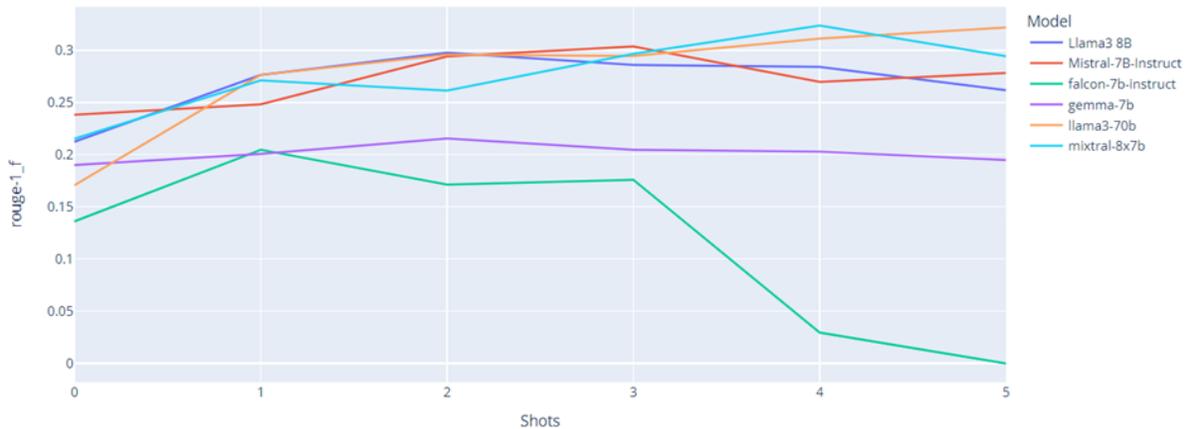


FIGURE 5.8: Graphique qui compare les résultats des modèles par rapport au nombre de shots

La Figure 5.8 illustre la variation des performances des modèles pour chaque shot en se concentrant sur la moyenne des scores ROUGE-1-F des 3 questions :

- Le modèle **Mistral-8x7B** atteint son pic de performance avec 4 shots, atteignant une valeur de 0.323, suivi de près par le modèle **Llama3-70B** avec un score de 0.321 à 5 shots.
- Les modèles **Falcon-7b-Instruct** et **Gemma-7b** montrent des fluctuations plus marquées, avec une tendance générale à la baisse après deux ou trois shots. Cela pourrait indiquer des problèmes d'hallucination ou une diminution de la pertinence des réponses, comme observé précédemment.
- Les autres modèles (**Llama3 8B**, **Llama3-70B**, **Mistral-7B-Instruct**, et **Mixtral-8x7B**) affichent des performances très proches et constantes, sans baisses drastiques. Ils maintiennent des performances relativement élevées et stables même après leur pic, ce qui démontre leur robustesse et leur fiabilité.

Scores ROUGE Maximaux par Question pour Chaque Modèle

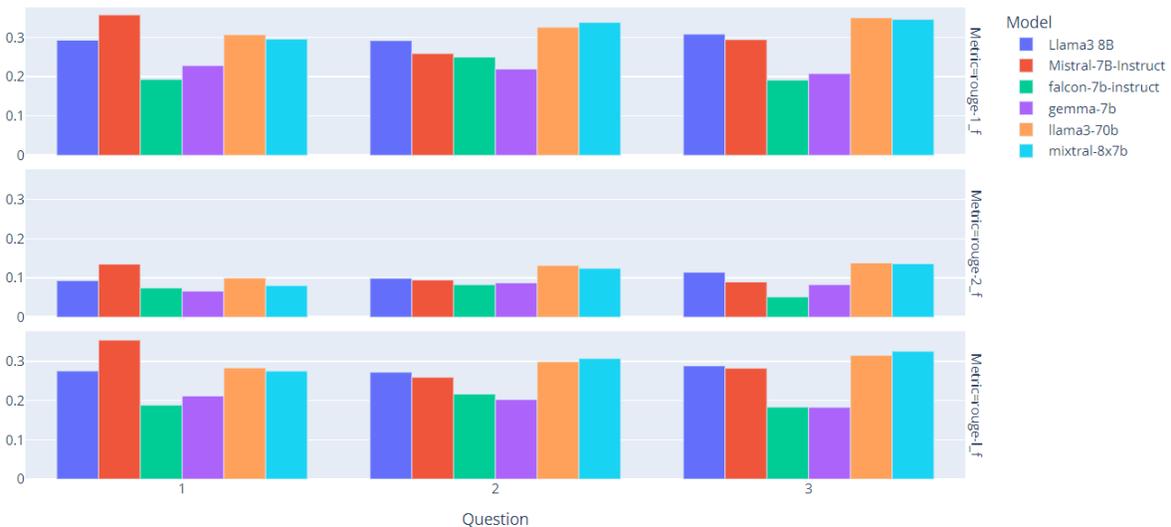


FIGURE 5.9: Graphique qui compare les résultats des modèles par rapport au questions

La Figure 5.9 illustre également la variation des performances des modèles pour chaque question en se concentrant sur les scores ROUGE-F maximum obtenus :

- Pour la question 1, le modèle **Mistral-7B-Instruct** se distingue par les meilleures performances, démontrant sa capacité à traiter efficacement cette question spécifique avec des scores ROUGE-F supérieurs.
- Pour les questions 2 et 3, les meilleurs scores sont atteints par les modèles **Llama3-70B** et **Mixtral-8x7B**, qui affichent des performances très proches l'un de l'autre, soulignant leur compétence dans le traitement de ces types de questions plus complexes.
- Le modèle **Llama3 8B** présente une stabilité remarquable dans ses résultats à travers les trois questions, ce qui souligne sa robustesse et sa fiabilité généralisée, indépendamment de la nature de la question.
- Les modèles **Falcon-7b-Instruct** et **Gemma-7b** se trouvent régulièrement en bas du classement pour toutes les questions, ce qui peut indiquer des lacunes dans leur capacité à généraliser ou à tirer pleinement parti des exemples fournis dans les prompts.

5.3.2.5 Conclusion de l'Analyse Comparative des Modèles

Cette analyse comparative révèle plusieurs points clés sur la performance des différents modèles de traitement du langage :

1. **Variabilité des Performances selon les Shots :** Les modèles présentent des seuils de performance optimaux à différents niveaux de shots, soulignant que le nombre optimal de shots varie d'un modèle à l'autre. Cette variabilité suggère que le nombre de shots est un paramètre crucial qui doit être ajusté spécifiquement pour chaque modèle plutôt que d'être considéré de manière uniforme.
2. **Indépendance de la Performance par rapport à la Capacité du Modèle :** Il est observé que la capacité nominale d'un modèle (mesurée en milliards de paramètres) ne détermine pas nécessairement sa performance. Des modèles avec une capacité moindre, comme certains modèles de 7 milliards de paramètres, peuvent surpasser ceux de 70 milliards de paramètres dans certains cas, indiquant que l'efficacité de l'apprentissage et l'adaptation au prompt sont souvent plus critiques que la simple quantité de paramètres.
3. **Classement des Modèles :** Les quatre modèles les plus performants — Llama3-70B, Llama3-8B, Mistral-7B-Instruct, et Mixtral-8x7B — démontrent des performances comparables et élevées sans un vainqueur clair entre eux. En revanche, les modèles Gemma-7B et Falcon-7b-Instruct montrent des performances nettement inférieures, ce qui justifie leur exclusion des phases ultérieures de développement et de déploiement.

Stratégie Future : Dans la poursuite de notre projet, nous avons sélectionné les quatre modèles les plus performants — Llama3-70B, Llama3-8B, Mistral-7B-Instruct, et Mistral-8x7B — pour déterminer celui qui répondra le mieux à nos besoins spécifiques. La prochaine étape consistera à affiner les modèles choisis en adoptant deux approches stratégiques : la Direct Preference Optimization (DPO) et la Retrieval-Augmented Generation (RAG). Ces méthodologies visent à améliorer la précision et l'efficacité des modèles selon nos besoins spécifiques.

5.3.3 Direct Preference Optimization (DPO)

5.3.3.1 Pourquoi utiliser la méthode de la Direct Preference Optimization (DPO) ?

- Inclusion du feedback humain : La méthode DPO intègre les retours humains, ce qui permet une meilleure orientation du modèle vers les résultats souhaités. Cette inclusion directe du feedback améliore l'adaptabilité et la précision du modèle en répondant aux attentes spécifiques des utilisateurs.

- Moins de données nécessaires : Comparée au fine-tuning supervisé traditionnel, la méthode DPO nécessite moins de données. Cette réduction de la quantité de données requises facilite et accélère le processus d'entraînement tout en maintenant une haute performance du modèle.
- Absence de modèle de récompense : Contrairement aux autres méthodes d'affinement des modèles de langage large (LLMs) utilisant le renforcement (Reinforcement Learning), la méthode DPO ne nécessite pas de modèle de récompense pour guider le modèle vers les réponses attendues. En DPO, l'algorithme apprend directement à produire la bonne réponse sans passer par l'entraînement d'un modèle de récompense, réduisant ainsi les ressources nécessaires et simplifiant le processus d'entraînement.

5.3.3.2 Schématisation de la solution avec DPO

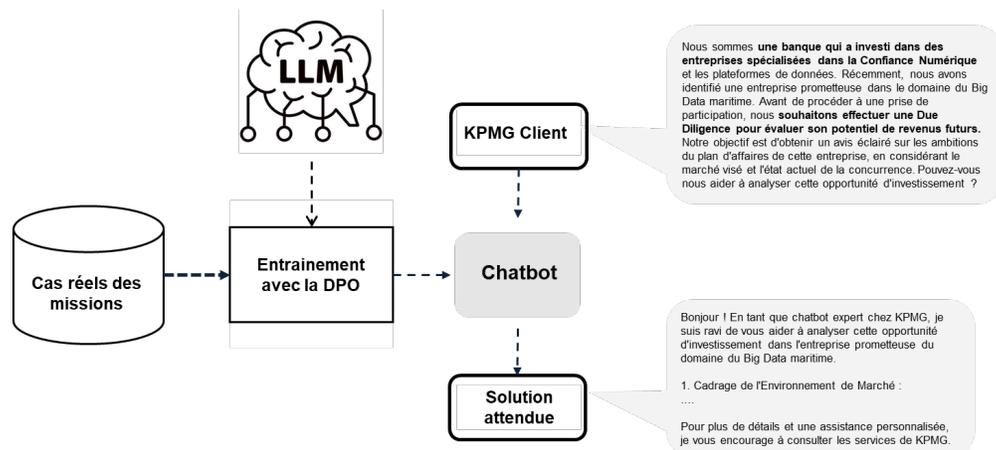


FIGURE 5.10: Schématisation de la solution avec DPO

La Figure 5.10 illustre le schéma de mise en œuvre de la solution utilisant la méthode DPO (Direct Preference Optimization), démontrant comment le modèle de langage (LLM) est entraîné avec des cas réels de missions et intégré dans un chatbot pour fournir des solutions attendues aux clients de KPMG.

5.3.3.3 Implémentation de la DPO

Pour implémenter l'approche de la Direct Preference Optimization nous avons suivi les étapes suivantes :

1. Choix du Modèle : Plusieurs modèles pré-entraînés ont été testés à l'aide du *prompt engineering*. Les modèles ayant donné les meilleurs résultats sont :
 - Mixtral 8x7 B
 - Llama 8B
 - Llama 70B
 - Mistral 7B
2. Données Utilisées : La base de données utilisée est structurée au format suivant : *prompt, réponse attendue, réponse rejetée*. Ce format est essentiel pour le déploiement de la méthode DPO.
3. Caractéristiques de la Machine : Le GPU utilisé est de la série *NVIDIA GeForce RTX 3060* avec 12 Go de VRAM comme démontré dans La Figure 5.11.
4. Configuration de l'Environnement CUDA : Pour utiliser le GPU, l'environnement CUDA a été configuré avec *PyTorch*. Un environnement virtuel a été créé sur *Jupyter Notebook* et CUDA avec *PyTorch* a été installé via la commande suivante :

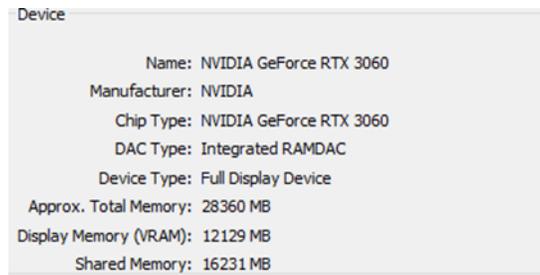


FIGURE 5.11: Les caractéristiques de la carte graphique de la machine

```
conda install pytorch==2.2.1 torchvision==0.17.1 pytorch-cuda=12.1 -c pytorch
-c nvidia
```

Cette configuration permet l'utilisation du GPU lors de l'entraînement.

5. Installation des Bibliothèques Nécessaires : Les bibliothèques suivantes ont été installées pour l'entraînement :
 - Datasets : Pour manipuler les données sous forme d'objets Dataset.
 - Transformers : Pour télécharger le modèle via Hugging Face.
 - TRL : Fournit des outils pour entraîner des modèles de langage avec l'apprentissage par renforcement.
 - PEFT : Permet le fine-tuning du modèle de base avant de l'entraîner avec la DPO.
 - Bitsandbytes : Un wrapper Python pour les fonctions CUDA personnalisées, incluant des optimiseurs 8 bits et des fonctions de quantification.
6. Importation des Bibliothèques : Les bibliothèques nécessaires pour l'exécution du code ont été importées dans l'environnement de développement.
7. Importation du Dataset : Les données ont été chargées à partir d'un fichier CSV à l'aide de la bibliothèque *Datasets* et mises au format adéquat pour l'entraînement.
8. Division des Données : Les données ont été divisées en ensemble d'entraînement (70%) et ensemble de validation (30%) à l'aide de la fonction *train_test_split*. Cette division permet d'évaluer les performances du modèle sur des données non vues pendant l'entraînement.
9. Téléchargement du Modèle : Le modèle pré-entraîné a été téléchargé depuis *Hugging Face*. La configuration de l'appareil pour l'entraînement a également été réalisée, en utilisant un GPU si disponible, sinon un CPU.
10. Préparation du Tokenizer : Un token de padding a été ajouté au tokenizer pour uniformiser la longueur des séquences de texte. Les embeddings du modèle ont ensuite été redimensionnés pour inclure cet ajout.
11. Définition des Arguments d'Entraînement : Les paramètres d'entraînement ont été définis à l'aide de la classe *TrainingArguments*. Cela inclut le nombre d'époques, la taille des lots, le nombre de *steps* de *warmup*, la dégradation du poids, etc.
12. Initialisation du DPOTrainer : Le *DPOTrainer* a été initialisé avec le modèle, la configuration PEFT, les arguments d'entraînement et les jeux de données. Le *DPOTrainer* gère toute la boucle d'entraînement, y compris la mise à jour des paramètres du modèle et le suivi des métriques.
13. Entraînement du Modèle : Le modèle a été entraîné en utilisant la méthode *train* de l'instance *Trainer*. Cette méthode parcourt toutes les époques, met à jour les paramètres du modèle à chaque étape et suit les métriques.

Ces étapes détaillent le processus méthodologique qu'on a suivi pour la conception de notre solution en utilisant la méthode de la Direct Preference Optimization.

5.3.3.4 Configuration de l'Entraînement du Modèle

L'entraînement du modèle a été réalisé en utilisant une configuration soigneusement choisie pour maximiser l'efficacité et la précision des prédictions. Les paramètres clés de cette configuration sont les suivants :

- Nombre d'époques (num_train_epochs) : L'entraînement a été réalisé sur 5 époques, permettant au modèle de passer plusieurs fois sur l'ensemble des données d'entraînement pour affiner ses prédictions.
- Taille des lots (batch_size) : Une taille de lot de 2 a été utilisée tant pour l'entraînement que pour l'évaluation, ce qui facilite la gestion de la mémoire et peut aider à éviter l'overfitting.
- Optimiseur (optim) : L'optimiseur utilisé était 'paged_adamw_32bit', une variante d'AdamW optimisée pour une utilisation efficace de la mémoire, idéale pour des modèles de grande taille.
- Planificateur de taux d'apprentissage (lr_scheduler_type) : Un planificateur de type 'cosine' a été sélectionné pour ajuster le taux d'apprentissage selon une décroissance cosinusoidale au cours des époques, facilitant une convergence douce et évitant les sauts brusques dans l'optimisation.

5.3.3.5 Évaluation des résultats de l'entraînement

Le tableau 5.1 montre les résultats obtenus après l'entraînement de chaque epoch :

epoch	Training Loss	Validation Loss	Rewards/chosen	Rewards/rejected	Logps/rejected	Logps/chosen
1	0.119200	0.003018	7.899302	-2.862248	-256.505493	-516.633911
2	0.000000	0.002115	7.958516	-4.972060	-277.603638	-516.041687
3	0.000000	0.001991	7.996623	-5.165757	-279.540588	-515.660645
4	0.000000	0.001915	8.029451	-5.262581	-280.535828	-515.323236
5	0.000000	0.001860	8.053390	-5.339847	-281.281525	-515.093018

TABLE 5.1: Résultats obtenus après chaque epoch d'entraînement

La Figure 5.12 montre un graphique représentant la variation du loss d'entraînement et du loss de validation.

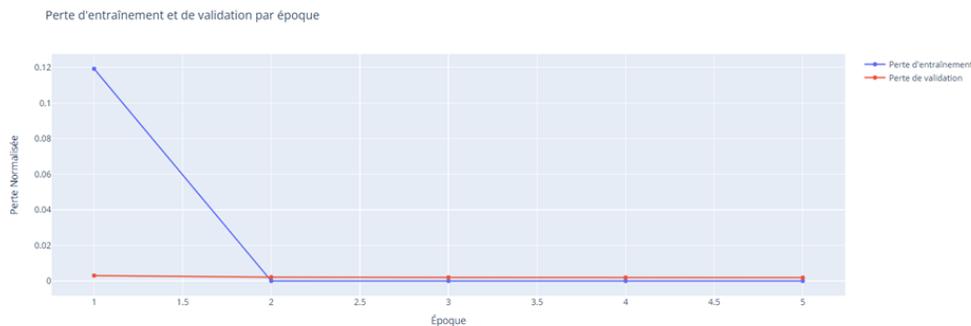


FIGURE 5.12: Graphique montrant la perte d'entraînement et de validation par epoch.

Afin d'obtenir une meilleure visualisation des variations, nous avons opté pour normaliser les résultats en utilisant la normalisation Min-Max. La Figure 5.13 présente ces données normalisées.

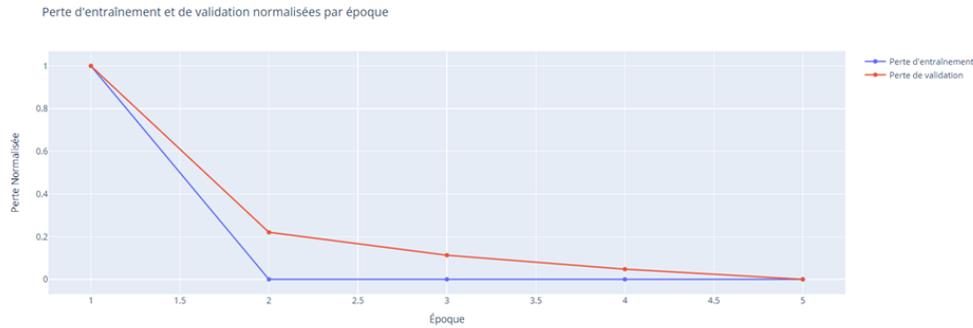


FIGURE 5.13: Graphique montrant la perte d'entraînement et de validation normalisées par epoch.

En interprétant les graphes, nous remarquons les points suivants :

1. **Convergence rapide de la perte d'entraînement** : La perte d'entraînement diminue brusquement de la première à la deuxième époque puis reste à zéro pour les époques suivantes. Cela indique que le modèle a rapidement appris à minimiser la fonction de perte sur les données d'entraînement.
2. **Diminution régulière de la perte de validation** : La perte de validation diminue de manière constante au fil des époques, bien que plus graduellement que la perte d'entraînement. Cela suggère que le modèle améliore sa performance sur des données non vues tout au long du processus d'entraînement.

En approfondissant l'évaluation de ces résultats, nous constatons les aspects critiques suivants :

- **Préoccupations de surapprentissage** : Puisque la perte d'entraînement atteint zéro dès la deuxième époque et y reste, tout en observant que la perte de validation continue de diminuer, il est crucial d'évaluer pour le surapprentissage. Bien que la diminution continue de la perte de validation soit un signe positif, la perte d'entraînement nulle dès un stade précoce pourrait indiquer que le modèle a parfaitement mémorisé les données d'entraînement plutôt que d'apprendre à généraliser.
- **Complexité du modèle et taille des données** : La réduction rapide de la perte d'entraînement à zéro suggère que le modèle pourrait être trop complexe par rapport à la taille des données d'entraînement. Avec un jeu de données petit, cette complexité pourrait conduire à un surapprentissage, bien que la tendance de la perte de validation ne soutienne pas actuellement cette préoccupation. Des tests supplémentaires avec plus de données ou l'utilisation de techniques telles que la validation croisée pourraient fournir des informations plus profondes.
- **Amélioration des récompenses et de la confiance dans les décisions** : L'augmentation des récompenses pour les choix acceptés et l'amélioration des log-probabilités pour les choix rejetés dans le tableau, indiquent une meilleure capacité du modèle à différencier les bonnes des mauvaises décisions au fil du temps, renforçant sa fiabilité dans des scénarios d'application réels.

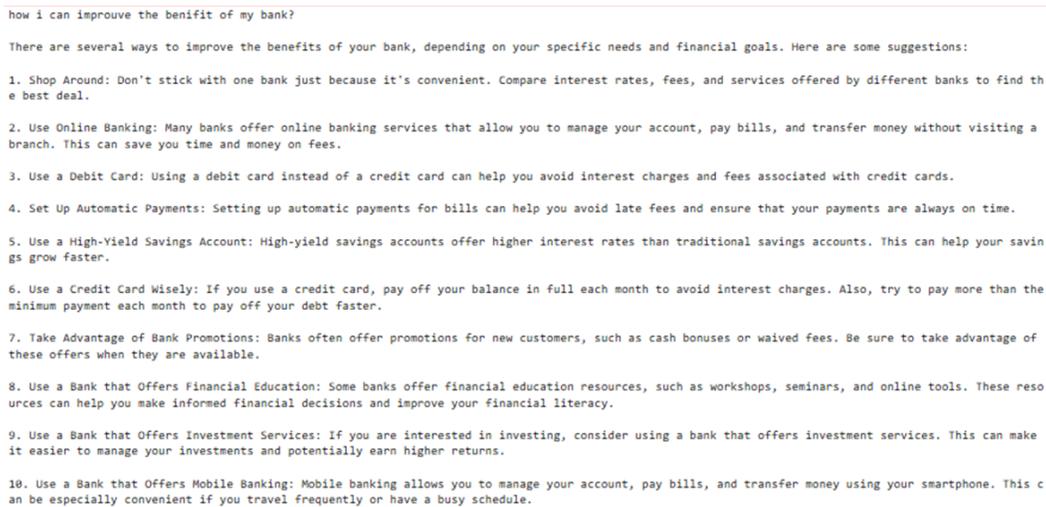
En résumé, les résultats de l'entraînement démontrent un processus de formation du modèle potentiellement efficace, avec un apprentissage rapide et de bonnes performances sur les données de validation. Cependant, l'indication d'un possible surapprentissage en raison d'une perte d'entraînement nulle nécessite une approche prudente. De plus, l'incorporation d'une validation croisée et de tests sur un ensemble de données plus large et plus varié pourrait offrir des insights plus significatifs sur la performance réelle du modèle. La progression des métriques telles que les récompenses et les log-probabilités souligne également la compétence du modèle à faire des choix précis, ce qui est essentiel pour son utilisation efficace dans des scénarios pratiques.

5.3.3.6 Évaluation des réponses du modèle :

Pour évaluer les performances du modèle Mistral 7B Instruct après l'entraînement, nous avons adopté une approche bimodale consistant à tester le modèle avec des questions issues de notre base de données

interne (Question 1, 2 et 3) ainsi que des questions externes (Question 4 et 5). Cette méthode permet une évaluation exhaustive, en explorant à la fois la capacité du modèle à généraliser au-delà des données d'entraînement et à fournir des réponses précises pour des requêtes connues. De plus, nous avons utilisé le score ROUGE pour mesurer la similarité entre les réponses du modèle et les réponses attendues. Nous avons également calculé le score d'hallucination pour vérifier la cohérence des réponses du modèle et s'assurer qu'il ne génère pas de contenu inapproprié ou incorrect.

- **Réponses à des questions externes de la base de donnée :** Le modèle a démontré une capacité notable à traiter et à répondre de manière structurée et détaillée à des questions nouvelles, non présentes dans la base de données. Les réponses étaient non seulement cohérentes et bien formulées, mais également présentées de manière séquentielle, ce qui témoigne de la capacité du modèle à fonctionner efficacement comme un assistant intelligent, comme illustrer dans la Figure 5.14.



```
how i can improve the benefit of my bank?

There are several ways to improve the benefits of your bank, depending on your specific needs and financial goals. Here are some suggestions:

1. Shop Around: Don't stick with one bank just because it's convenient. Compare interest rates, fees, and services offered by different banks to find the best deal.

2. Use Online Banking: Many banks offer online banking services that allow you to manage your account, pay bills, and transfer money without visiting a branch. This can save you time and money on fees.

3. Use a Debit Card: Using a debit card instead of a credit card can help you avoid interest charges and fees associated with credit cards.

4. Set Up Automatic Payments: Setting up automatic payments for bills can help you avoid late fees and ensure that your payments are always on time.

5. Use a High-Yield Savings Account: High-yield savings accounts offer higher interest rates than traditional savings accounts. This can help your savings grow faster.

6. Use a Credit Card Wisely: If you use a credit card, pay off your balance in full each month to avoid interest charges. Also, try to pay more than the minimum payment each month to pay off your debt faster.

7. Take Advantage of Bank Promotions: Banks often offer promotions for new customers, such as cash bonuses or waived fees. Be sure to take advantage of these offers when they are available.

8. Use a Bank that Offers Financial Education: Some banks offer financial education resources, such as workshops, seminars, and online tools. These resources can help you make informed financial decisions and improve your financial literacy.

9. Use a Bank that Offers Investment Services: If you are interested in investing, consider using a bank that offers investment services. This can make it easier to manage your investments and potentially earn higher returns.

10. Use a Bank that Offers Mobile Banking: Mobile banking allows you to manage your account, pay bills, and transfer money using your smartphone. This can be especially convenient if you travel frequently or have a busy schedule.
```

FIGURE 5.14: La réponse du modèle à une question extérieure de la base de données

- **Réponses aux questions de la base de données :** Lorsqu'interrogé avec des questions de la base de données, le modèle a produit des réponses extrêmement cohérentes, structurées en plusieurs étapes et intégrant des informations spécifiques extraites directement de la base de données. Ces réponses non seulement respectent la structure attendue mais démontrent également la capacité du modèle à réutiliser efficacement les données apprises. De plus, son comportement s'apparente à celui d'un assistant qualifié, prêt à offrir des réponses informatives et bien organisées, Figure 5.15.

We are currently interested in exploring mobile payment solutions in the French market, specifically for our Chinese expatriate clients using these services to pay their electricity bills. As the R&D Department, we aim to understand if the popular mobile payment systems in China can be adapted to the French context while complying with local regulations. Our goal is to find alternatives to credit card payments or direct debits. Can you assist us in studying the available options in the French market?

Absolutely, we'd be delighted to assist you in exploring mobile payment solutions for your Chinese expatriate clients in the French market. Here's a structured approach to address your needs:

1. Market Analysis: We will conduct a comprehensive analysis of the French market, focusing on mobile payment solutions and their adoption rates among the French population.
2. Regulatory Compliance: We will ensure that any proposed solutions comply with French regulations, particularly those related to data protection and financial transactions.
3. Adaptation of Chinese Solutions: We will investigate the feasibility of adapting popular Chinese mobile payment systems to the French context.
4. Focus on Electricity Bills: We will specifically focus on solutions that facilitate the payment of electricity bills, ensuring seamless integration with existing systems.

Here's a detailed breakdown of our approach:

Market Analysis:

- Analysis of the French mobile payment market
- Identification of the main players and their market shares
- Analysis of the adoption rates of mobile payment solutions among the French population

Regulatory Compliance:

- Analysis of French regulations related to data protection and financial transactions
- Definition of compliance requirements for mobile payment solutions

Adaptation of Chinese Solutions:

- Analysis of popular Chinese mobile payment systems
- Identification of the key features that could be adapted to the French context
- Definition of the adaptation strategy

Focus on Electricity Bills:

- Analysis of the current process for paying electricity bills
- Identification of the key challenges and opportunities for mobile payment solutions
- Definition of a strategy for integrating mobile payment solutions into the existing process

We will deliver a comprehensive report detailing our findings and recommendations for implementing mobile payment solutions for your Chinese expatriate clients in the French market.

FIGURE 5.15: La réponse du modèle à une question de la base de données

Le Tableau 5.2 montre les résultats des scores ROUGE obtenus après avoir testé le modèle sur cinq questions :

Question	rouge-1_r	rouge-1_p	rouge-1_f	rouge-2_r	rouge-2_p	rouge-2_f	rouge-lr	rouge-lp	rouge-lf
1	0.381356	0.292208	0.330882	0.127389	0.076336	0.095465	0.322034	0.246753	0.279412
2	0.321678	0.310811	0.316151	0.110000	0.083333	0.094828	0.272727	0.263514	0.268041
3	0.282051	0.294643	0.288210	0.116279	0.103093	0.109290	0.230769	0.241071	0.235808
4	0.198020	0.181818	0.189573	0.084507	0.060914	0.070796	0.188119	0.172727	0.180095
5	0.258065	0.253165	0.255591	0.069124	0.056180	0.061983	0.238710	0.234177	0.236422

TABLE 5.2: Résultats des scores ROUGE pour chaque question testée

La Figure 5.16 illustre la distribution des scores ROUGE-1_f pour les cinq questions, permettant de visualiser la variabilité des performances du modèle sur ces différentes requêtes.

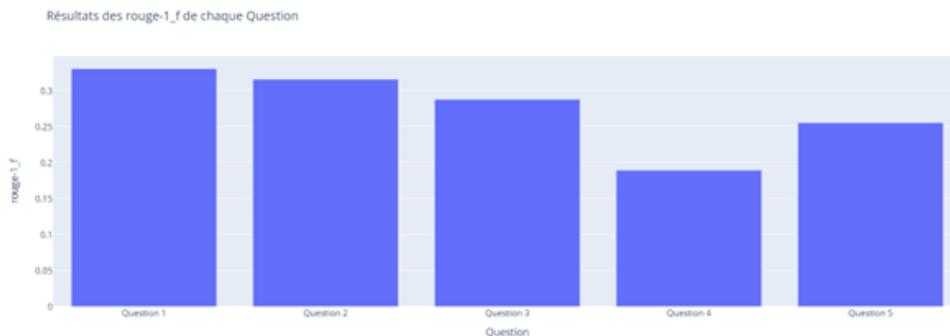


FIGURE 5.16: Distribution des scores ROUGE-1_f pour chaque question

Interprétation des Résultats :

- Les résultats pour les questions issues de la base de données (BDD) affichent des scores ROUGE élevés et stables (0.33, 0.31, et 0.28), démontrant que le modèle a efficacement appris à générer des

réponses pertinentes.

- Malgré son origine hors BDD, la Question 5 obtient un score de 0.25, proche de ceux obtenus pour les questions de la BDD. Cela souligne une capacité remarquable du modèle à généraliser ses apprentissages.
- La Question 4, en revanche, montre le score F1 le plus bas, suggérant une baisse de performance qui pourrait être attribuée à sa complexité ou à son caractère hors BDD, potentiellement au-delà des capacités actuelles du modèle.
- Le score moyen de l'hallucination des réponses est de 0.86244184, indiquant que la grande majorité des réponses générées par le modèle sont basées sur des données fiables et pertinentes.

En résumé, les scores ROUGE obtenus par le modèle sont significatifs et compétitifs par rapport à ceux issus de techniques de Few-shot Learning. Ils illustrent la capacité du modèle à ajuster ses paramètres et à formuler des réponses pertinentes en s'appuyant sur des exemples, sans dépendre de prompts explicites comme cela est souvent nécessaire dans le Few-shot Learning. Le score élevé de non-hallucination renforce la confiance dans la précision et la pertinence des réponses générées. Néanmoins, il est essentiel d'élargir le corpus d'entraînement et de peaufiner les capacités de compréhension du modèle pour améliorer sa capacité de généralisation et la précision de ses réponses dans des scénarios diversifiés.

5.3.4 Retrieval Augmented Generation RAG

5.3.4.1 Pourquoi opter pour la méthode RAG ?

Nous avons opté pour la méthode Retrieval-Augmented Generation (RAG) pour plusieurs raisons :

- Réduction des coûts : La méthode RAG ne nécessite pas d'entraînement ou de modification du modèle de base, ce qui réduit les coûts d'implémentation par rapport à d'autres méthodes comme le *fine-tuning* ou la DPO, qui demandent énormément de ressources.
- Solutions adaptées à des cas réels (avec des données internes) : Les modèles habituels sont généralement entraînés sur des données *open source* issues d'Internet ou de bases de données connues. La méthode RAG, en revanche, utilise les données internes de l'entreprise, effectue des requêtes sur ces données et fournit des réponses cohérentes grâce à la capacité de génération du modèle de langage choisi, sans nécessiter un entraînement spécifique avec ces données.
- Réponses actualisées : Chaque fois que de nouvelles données sont disponibles, elles peuvent être ajoutées directement à la base de données vectorielle et mises à jour, contrairement à l'entraînement qui nécessite une nouvelle session d'entraînement.
- Simplicité de l'implémentation : La méthode RAG est simple à implémenter.

5.3.4.2 Schématisation de la solution avec RAG

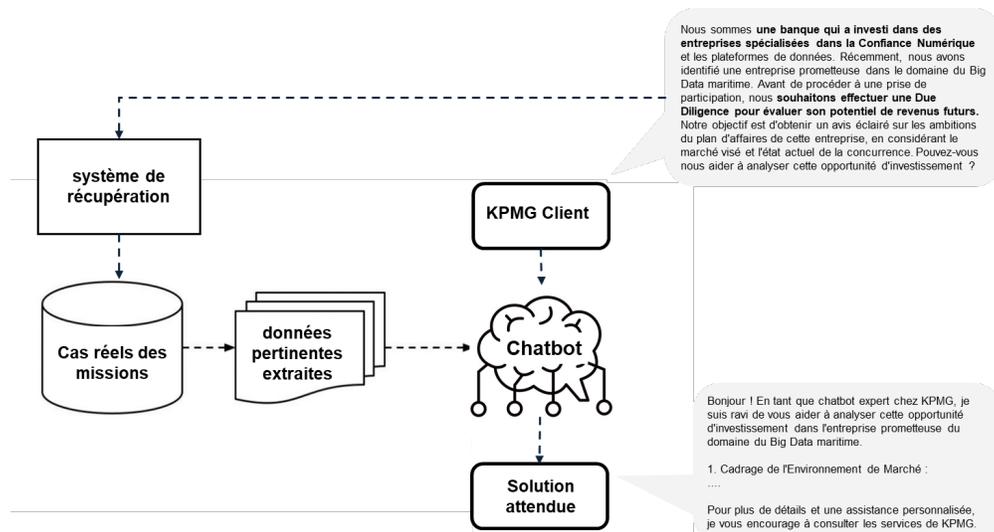


FIGURE 5.17: Schématisation de la solution avec RAG

La Figure 5.17 démontre la schématisation de la mise en œuvre de la solution utilisant la méthode RAG (Retrieval Augmented Generation). Ce schéma explique comment les requêtes initiales d'un client sont traitées pour extraire des données pertinentes à partir de cas réels de missions, lesquelles sont ensuite utilisées par un LLM qui représente le chatbot afin de générer des solutions attendues.

5.3.4.3 Implémentation des RAGs

Après avoir choisi les modèles à tester, nous avons préparé notre base de données pour l'implémentation de la méthode, ainsi que l'environnement de développement.

Suite à la phase de *prompt engineering*, les modèles ayant montré les meilleurs résultats en utilisant la méthode *few-shot learning* sont les suivants :

- Mixtral 7x8B avec un score ROUGE-1 F1 maximale de **0.34**.
- LLaMA 8B avec un score ROUGE-1 F1 maximale de **0.3**.
- Llama3 70B avec un score ROUGE-1 F1 maximale de **0.35**.
- Mistral 7b Instruct avec un score ROUGE-1 F1 maximale de **0.27**.

Suite à ces résultats, nous avons décidé de tester les quatre modèles cités ci-dessus en utilisant la méthode RAG.

1. Installation des bibliothèques nécessaires : Nous avons commencé par installer les bibliothèques nécessaires à l'implémentation des RAG, à savoir :
 - *Langchain* : une bibliothèque contenant tous les packages nécessaires à l'implémentation des RAG.
 - *Transformers* : une bibliothèque de Hugging Face permettant d'importer les modèles.
 - *langchain-groq* : une bibliothèque qui permet l'utilisation des modèles de GroqCloud avec la bibliothèque *langchain*.
2. Téléchargement de la base de données : Nous avons téléchargé la base de données contenant les questions et leurs réponses en format CSV.
3. Création des embeddings : Nous avons créé une instance *HuggingFaceEmbeddings* en utilisant le modèle d'embedding *sentence-transformers/all-MiniLM-l6-v2* qui permet de faire l'encodage de la base de données.
4. Conception de la base de données vectorielle : Nous avons utilisé le module FAISS pour encoder les données, créer des vecteurs d'*embedding* en utilisant l'instance d'embedding déjà créée, et les charger

dans la base de données vectorielle.

5. Téléchargement des modèles : Nous avons téléchargé les modèles soit via Hugging Face pour Falcon 7B, soit via GroqCloud pour les autres modèles.
6. Préparation du *prompt* : Nous avons préparé des exemple de questions pouvant être posées par un client de KPMG à notre modèle de langage.

— *Prompt* avec template : Un *prompt* avec une structure précise, par exemple :

```
prompt_template = """
```

```
### [INST]
```

```
Instruction: You are an expert chatbot for KPMG.
```

```
You excel at providing responses to client issues through a series of steps.
```

```
Help as much as you can to solve our problems:
```

```
{context}
```

```
### QUESTION:
```

```
{question}
```

```
[/INST]
```

```
"""
```

7. Création d'un *retriever* : Nous avons créé une instance en appliquant la méthode `as_retriever` à la base de données vectorielle pour récupérer les données.
8. Utilisation de la classe *retrievalQA* : Nous avons utilisé la classe *retrievalQA* de la librairie *langchain.chains* pour passer le *prompt* au modèle de langage avec le *retriever* déjà créé et le modèle télécharger. Le résultat obtenu représente la réponse du modèle de langage en utilisant la méthode RAG.

Nous avons effectué plusieurs tests sur les modèles cités précédemment en utilisant divers questions de la base de données.

5.3.4.3 Évaluation des résultats

a) Métrique Utilisée

Les métriques utilisées pour évaluer les résultats des modèles sont le *ROUGE Score*, ainsi que l'utilisation d'un modèle qui calcule un score d'hallucination des réponses générées par les modèles.

b) Modèles Utilisés

Les modèles suivants ont été évalués :

- Mixtral 7x8b
- Llama3 70b
- Llama3 8b
- Mistral 7b Instruct

c) Méthodologie d'Évaluation

Pour comparer les performances des modèles, deux approches ont été utilisées :

1. Évaluation des Réponses du Modèle sur Plusieurs Questions
 - Nous avons sélectionné huit questions aléatoires de la base de données.

- Les réponses de chaque modèle à ces huit questions ont été évaluées.
 - Les résultats obtenus sont détaillés ci-dessous.
2. Évaluation des Réponses du Modèle avec Variation de la Température
- La température de chaque modèle a été variée de 0 à 1 par incréments de 0,2.
 - Trois questions aléatoires de la base de données ont été sélectionnées pour cette évaluation.
 - Le *ROUGE Score* a été calculé pour chaque variation de la température afin d’observer son impact sur les performances des modèles.

d) Résultats de l’Évaluation

Mixtral 8x7b :

- Avec variation des questions :

Les résultats obtenus de l’application de la méthode RAG avec Mixtral 7x8B sont démontré dans le tableau 5.18, et la figure 5.3.

Question	rouge-1_r	rouge-1_p	rouge-1_f	rouge-2_r	rouge-2_p	rouge-2_f	rouge-l_r	rouge-l_p	rouge-l_f
Question 1	0.780	0.786	0.783	0.605	0.617	0.611	0.771	0.778	0.774
Question 2	0.847	0.800	0.823	0.729	0.682	0.704	0.847	0.800	0.823
Question 3	0.784	0.752	0.768	0.568	0.578	0.573	0.767	0.736	0.751
Question 4	0.670	0.570	0.616	0.552	0.468	0.506	0.650	0.554	0.598
Question 5	0.838	0.827	0.832	0.720	0.683	0.701	0.818	0.807	0.812
Question 6	0.690	0.731	0.710	0.474	0.506	0.489	0.676	0.716	0.696
Question 7	0.747	0.770	0.758	0.568	0.601	0.584	0.741	0.764	0.752
Question 8	0.758	0.711	0.734	0.556	0.514	0.534	0.742	0.695	0.718

TABLE 5.3: Résultats du ROUGE Score pour le modèle Mixtral 8x7b sur huit questions de la BDD

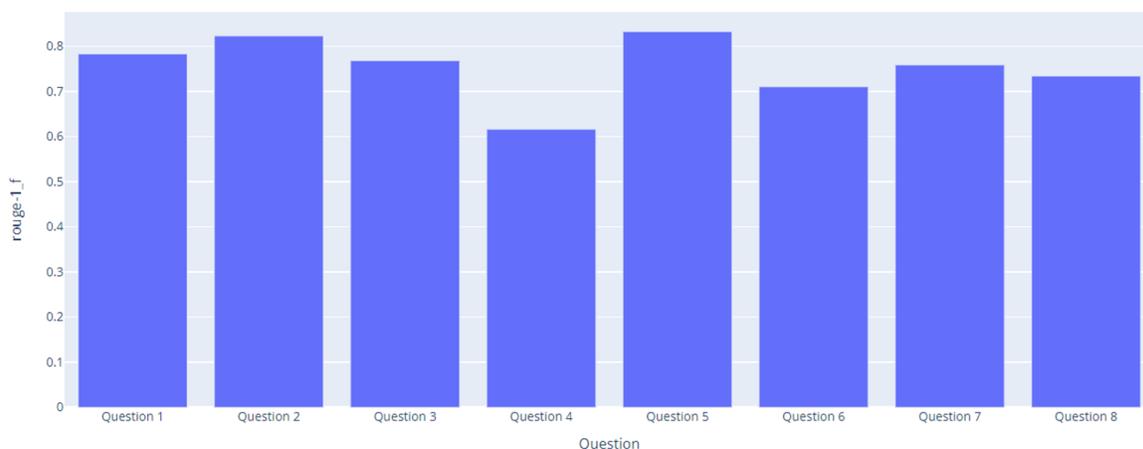


FIGURE 5.18: Les résultats du *rouge – 1_f* du modèle Mixtral 8x7b sur huit questions de la BDD

En interprétant le graphe on remarque les points suivants :

- En few-shot learning, Mixtral 8x7 b avait un score ROUGE maximal de 0.34.
- Avec les RAGs, le score ROUGE a augmenté de manière significative jusqu’à 0.83.
- Le score ROUGE-1.f de la Question 4 est plus bas (environ 0.6) comparé aux autres questions (entre 0.75 et 0.85).

- Cette disparité peut être due à la complexité du contexte, la qualité des données, la variabilité des réponses, ou les limitations spécifiques du modèle.

— **Avec variation de la température**

Le graphique de la figure 5.19 montre les résultats obtenus lors de la variation de la température sur trois questions de la base de données avec le modèle Mixtral 8x7 B :

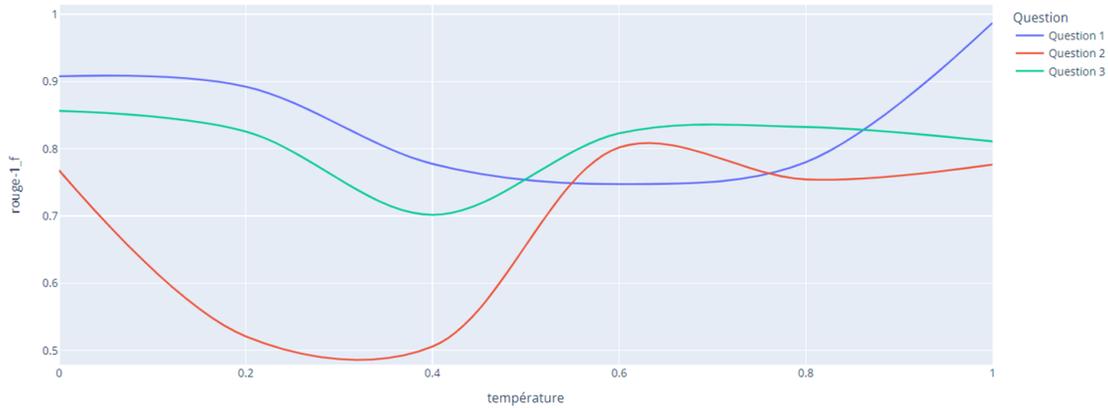


FIGURE 5.19: Les résultats du $rouge - 1_f$ du modèle Mixtral 8x7b avec variation de la température sur trois questions de la BDD

En interprétant le graphe on remarque les points suivants :

- Les trois questions montrent des comportements différents face à la variation de la température.
- Question 1 et Question 2 ont des tendances similaires avec des diminutions initiales suivies d'une augmentation à des températures plus élevées.
- Question 3 montre une diminution plus modérée et une légère remontée.
- Le modèle performe mieux pour les trois questions à des températures extrêmes (soit très basses, soit très élevées).
- À la température de 0.85, les performances du modèle semblent plus optimales pour toutes les questions.

Llama3 8b :

— **Avec variation des questions**

Les résultats obtenus de l'application de la méthode RAG avec Llama3 8B sont démontrés dans le tableau 5.20, et la figure 5.4.

Question	rouge-1_r	rouge-1_p	rouge-1_f	rouge-2_r	rouge-2_p	rouge-2_f	rouge-L_r	rouge-L_p	rouge-L_f
Question 1	0.843	0.770	0.805	0.775	0.663	0.715	0.843	0.770	0.805
Question 2	0.914	0.688	0.785	0.874	0.640	0.739	0.914	0.688	0.785
Question 3	0.885	0.898	0.891	0.857	0.876	0.866	0.885	0.898	0.891
Question 4	0.810	0.712	0.758	0.742	0.632	0.682	0.802	0.705	0.750
Question 5	0.816	0.679	0.741	0.749	0.601	0.667	0.816	0.679	0.741
Question 6	0.748	0.695	0.720	0.650	0.571	0.608	0.742	0.689	0.714
Question 7	0.794	0.688	0.737	0.708	0.617	0.659	0.789	0.683	0.732
Question 8	0.876	0.727	0.795	0.729	0.599	0.658	0.876	0.727	0.795

TABLE 5.4: Résultats du ROUGE Score pour le modèle Llama3 8b sur huit questions de la BDD.

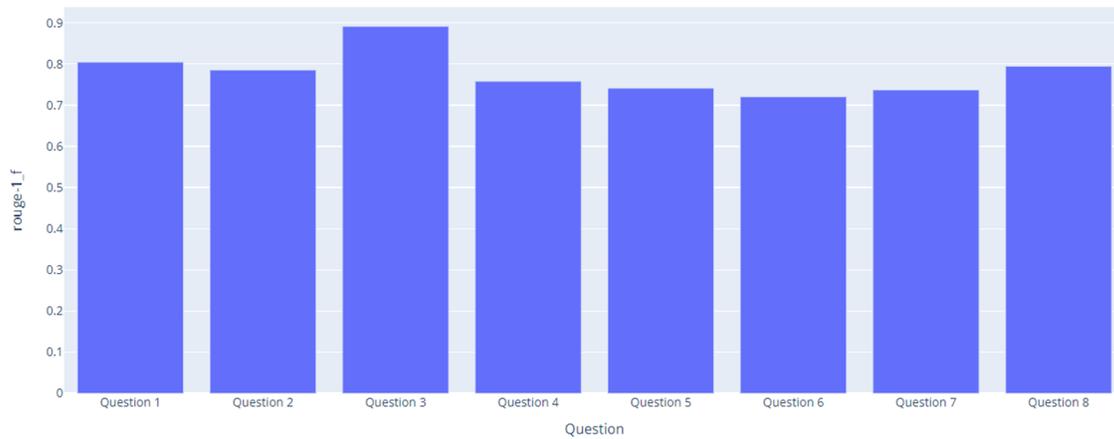


FIGURE 5.20: Les résultats du *rouge* – 1_f du modèle Llama3 8b sur huit questions de la BDD

En interprétant le graphe on remarque les points suivants :

- La valeur maximale obtenue pour Llama3 8b en few-shot learning était un score ROUGE de 0.3. Cela indique une performance limitée où les réponses fournies par le modèle avaient une correspondance modeste avec le texte de référence.
- Avec l'intégration des RAGs (Retrieval-Augmented Generation), le score ROUGE a considérablement augmenté, atteignant un maximum de 0.89. Cette augmentation significative montre que les RAGs ont permis d'améliorer la qualité des réponses générées par le modèle.
- Les scores varient légèrement entre les questions, mais toutes montrent des performances élevées, allant de 0.74 à 0.89.

— Avec variation de la température

Le graphique de la figure 5.21 montre les résultats obtenues lors de la variation de la température sur trois questions de la base de données avec le modèle Llama3 8B :

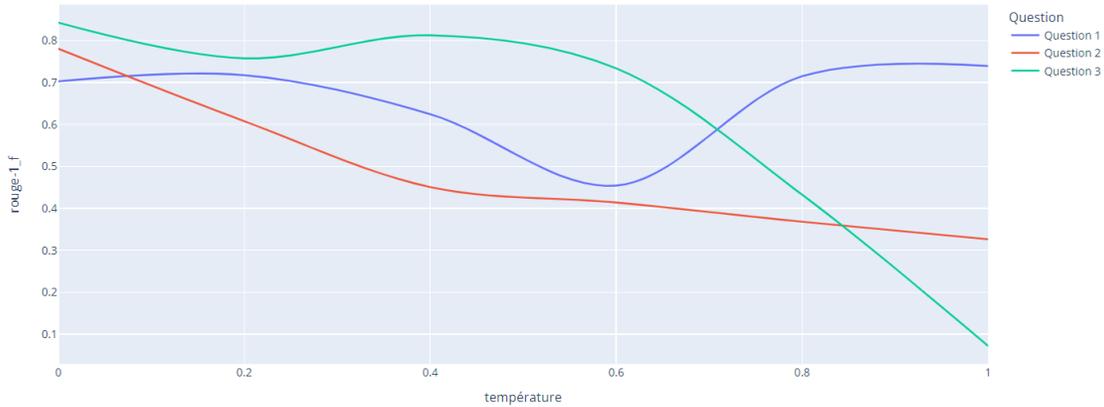


FIGURE 5.21: Les résultats du $rouge - 1_f$ du modèle Llama3 8b avec variation de la température sur trois questions de la BDD

En interprétant le graphe on remarque les points suivants :

- Les trois questions montrent des comportements différents face à la variation de la température.
- Le modèle performe mieux pour les trois questions à des températures basses.
- À des températures très basses (0.0 - 0.2), le modèle montre des performances élevées et stables pour toutes les questions.
- À des températures élevées (0.8 - 1.0), les performances diminuent pour toutes les questions, avec la Question 3 montrant la plus grande baisse.

Llama3 70B

— Avec variation des questions :

Les résultats obtenus de l'application de la méthode RAG avec Llama3 70B sont démontré dans le tableau 5.22, et la figure 5.5.

Question	rouge-1_r	rouge-1_p	rouge-1_f	rouge-2_r	rouge-2_p	rouge-2_f	rouge-l_r	rouge-l_p	rouge-l_f
Question 1	0.827	0.625	0.712	0.719	0.481	0.577	0.827	0.625	0.712
Question 2	0.895	0.673	0.768	0.874	0.617	0.723	0.895	0.673	0.768
Question 3	0.915	0.730	0.812	0.847	0.623	0.717	0.915	0.730	0.812
Question 4	0.534	0.416	0.468	0.400	0.258	0.314	0.534	0.416	0.468
Question 5	0.899	0.751	0.818	0.823	0.639	0.719	0.899	0.751	0.818
Question 6	0.910	0.825	0.865	0.862	0.751	0.803	0.910	0.825	0.865
Question 7	0.917	0.838	0.875	0.852	0.766	0.807	0.917	0.838	0.875
Question 8	0.934	0.771	0.845	0.788	0.663	0.720	0.934	0.771	0.845

TABLE 5.5: Résultats du ROUGE Score pour le modèle Llama3 70b sur huit questions de la BDD.

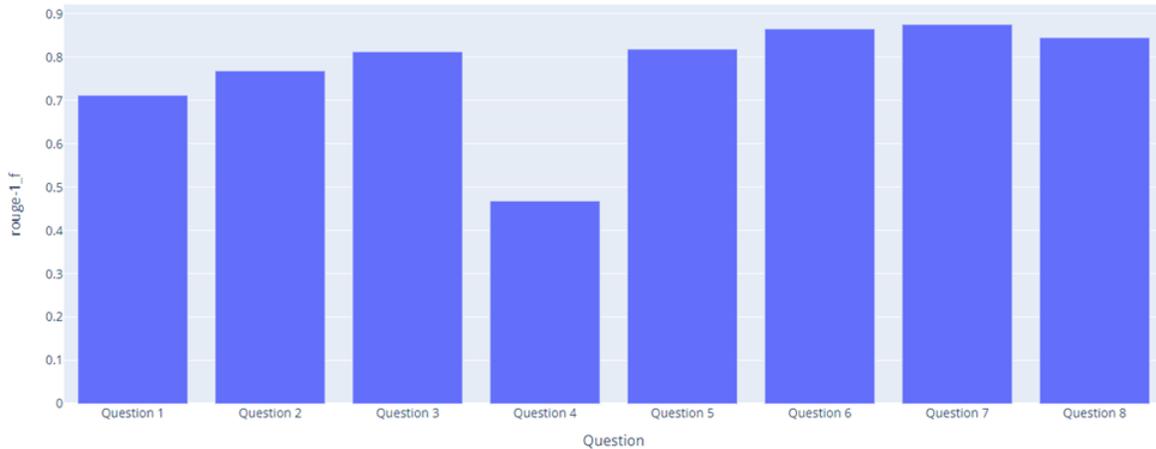


FIGURE 5.22: Les résultats du $rouge - 1_f$ du modèle Llama3 70b sur huit questions de la BDD

En interprétant le graphe on remarque les points suivants :

- La valeur maximale obtenue pour Llama3 70b en few-shot learning était un score ROUGE de 0.35. Cela indique une performance limitée où les réponses fournies par le modèle avaient une correspondance modeste avec le texte de référence.
- Avec l'intégration des RAGs, le score ROUGE a considérablement augmenté, atteignant un maximum de 0.87. Cette augmentation significative montre que les RAGs ont permis d'améliorer la qualité des réponses générées par le modèle.
- Le graphique montre que les scores ROUGE-1_f varient entre les questions, allant de 0.47 à 0.88.
- La Question 4 affiche le score le plus bas (0.47), ce qui peut indiquer une complexité particulière ou des difficultés spécifiques pour cette question.
- Les variations de scores entre les questions montrent que le modèle est capable de générer des réponses de haute qualité pour la plupart des questions, mais peut rencontrer des difficultés spécifiques pour certaines, comme la question 4.

— Avec variation de la température

Le graphique de la figure 5.23 montre les résultats obtenues lors de la variation de la température sur trois questions de la base de données avec le modèle Llama3 70B :

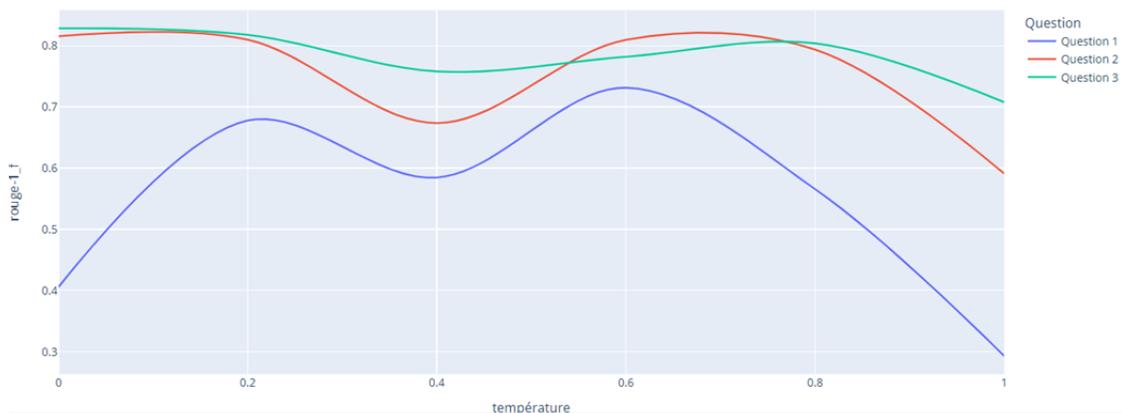


FIGURE 5.23: Les résultats du $rouge - 1_f$ du modèle Llama3 70b avec variation de la température sur trois questions de la BDD

En interprétant le graphe on remarque les points suivants :

- À une température de 0.6, le modèle a atteint son pic de performance pour les trois questions. Cela suggère que cette température est optimale pour obtenir les meilleures réponses du modèle.
- Les variations de performances montrent que le modèle réagit différemment à la température pour chaque question, avec des comportements distincts pour les trois questions.
- À des températures très basses (0.0 à 0.2) et très élevées (0.8 à 1.0), les performances du modèle sont généralement moins stables, avec une diminution notable des scores pour la Question 1 et des baisses modérées pour les Questions 2 et 3. En revanche, à des températures moyennes (0.5 à 0.7), le modèle atteint sa performance optimale.

Mistral 7b Instruct

— Avec variation des questions :

Les résultats obtenus de l'application de la méthode RAG avec Mistral 7B Instruct sont démontré dans le tableau 5.6, et la figure 5.24.

Question	rouge-1_r	rouge-1_p	rouge-1_f	rouge-2_r	rouge-2_p	rouge-2_f	rouge-l_r	rouge-l_p	rouge-l_f
Question 1	0.568	0.523	0.545	0.287	0.281	0.284	0.551	0.508	0.528
Question 2	0.839	0.767	0.801	0.665	0.625	0.644	0.839	0.767	0.801
Question 3	0.672	0.578	0.622	0.343	0.333	0.338	0.647	0.556	0.598
Question 4	0.689	0.628	0.657	0.462	0.456	0.459	0.689	0.628	0.657
Question 5	0.642	0.487	0.554	0.329	0.278	0.301	0.595	0.451	0.513
Question 6	0.613	0.521	0.563	0.268	0.245	0.256	0.556	0.473	0.511
Question 7	0.602	0.505	0.549	0.199	0.194	0.196	0.536	0.449	0.489
Question 8	0.752	0.710	0.730	0.667	0.632	0.649	0.745	0.703	0.723

TABLE 5.6: Résultats du ROUGE Score pour le modèle Mistral 7b Instruct sur huit questions de la BDD.

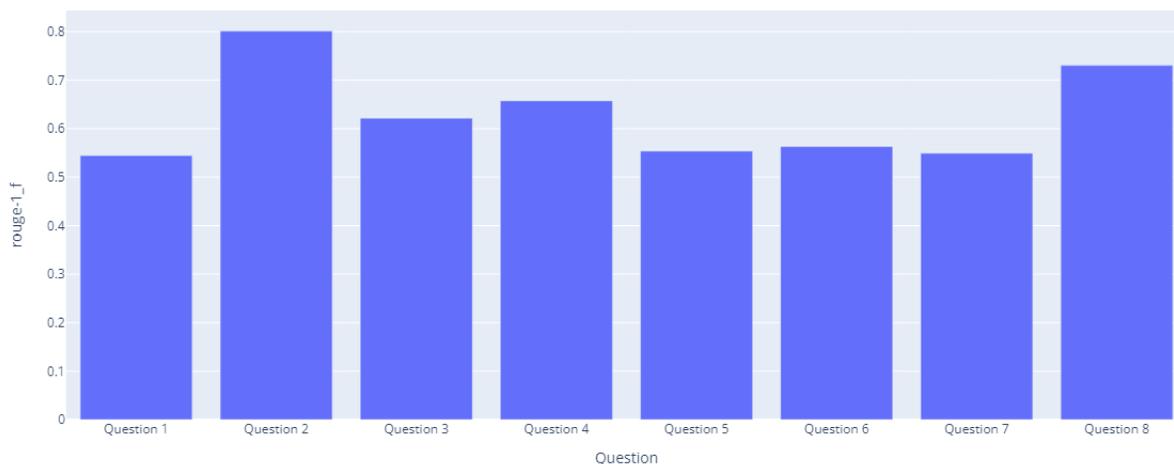


FIGURE 5.24: Les résultats du *rouge* – 1_f du modèle Mistral 7b Instruct sur huit questions de la BDD

En interprétant le graphe on remarque les points suivants :

- Le modèle Mistral 7B a initialement obtenu un ROUGE-1_F score de 0.27 lors de la phase de few-shot learning. Après l'application de la méthode Retrieval-Augmented Generation (RAG), on observe une amélioration significative du score pour toutes les questions. Cette augmentation démontre l'efficacité de l'approche RAG pour affiner les capacités du modèle Mistral 7B, en le rendant plus performant et pertinent dans ses réponses.

- Le graphique montre que le modèle Mistral 7B atteint un ROUGE-1_F score supérieur à 0.5 pour toutes les questions, indiquant une performance consistante à travers différentes entrées.

— **Avec variation de la température**

Le graphique de la figure 5.25 montre les résultats obtenues lors de la variation de la température sur trois questions de la base de données avec le modèle Mistral 7b Instruct :

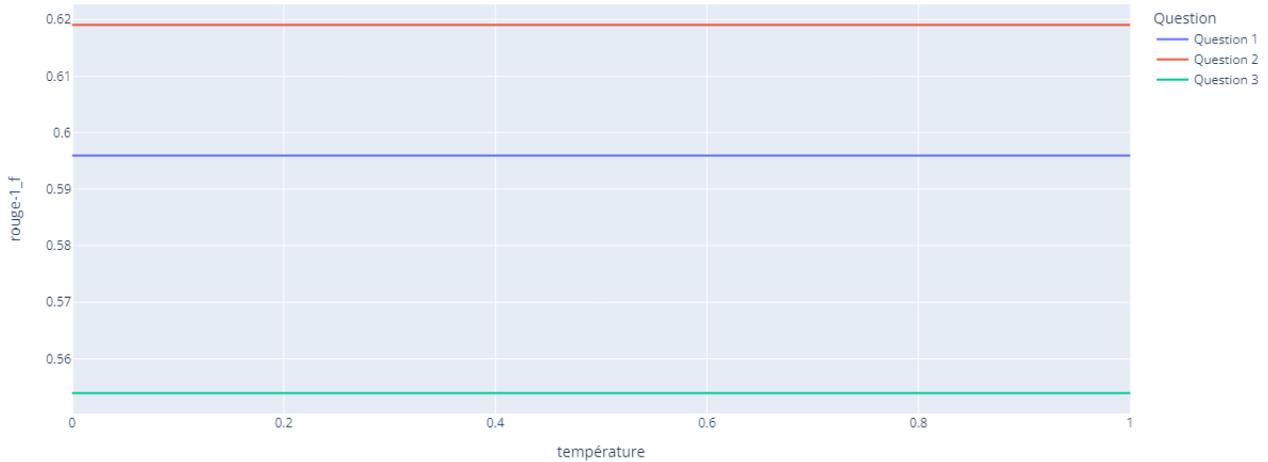


FIGURE 5.25: Les résultats du $rouge - 1_f$ du modèle Mistral 7b Instruct avec variation de la température sur trois questions de la BDD

En interprétant le graphe on remarque les points suivants :

- Le graphique montre que le ROUGE-1_F score du modèle Mistral 7B reste constant quelle que soit la température, variant de 0 à 1. Cela indique que la performance du modèle ne dépend pas de la température lorsqu'il répond aux questions choisies.
- Malgré la stabilité des scores par rapport à la température, il est évident que la performance du modèle varie en fonction des questions. Par exemple, la question 2 obtient un ROUGE-1_F score de 0.62, qui est supérieur aux scores obtenus pour les questions 1 et 3, qui sont de 0.60 et 0.56 respectivement.

e) Analyse comparative entre les quatre modèles

Pour comparer les performances des différents modèles utilisés avec l'approche Retrieval-Augmented Generation (RAG), nous avons évalué deux métriques principales : le ROUGE score moyen et l'allucination moyenne des réponses générées par les modèles sur huit questions. Les modèles comparés sont les suivants :

- Mixtral 8x7b
- Llama3 70b
- Llama3 8b
- Mistral 7b Instruct

On a comparé les quatre modèles testés avec les métriques d'évaluations suivantes :

- **ROUGE Score** : Il s'agit d'une métrique d'évaluation des réponses générées par les modèles en comparant les réponses de référence avec celles générées. Nous avons calculé les scores ROUGE-1, ROUGE-2 et ROUGE-L (rappel, précision, F-score) pour chaque modèle.
- **Allucination** : Nous avons évalué l'allucination des réponses générées à l'aide du modèle *vectara/*

hallucination_evaluation_model de Hugging Face, conçu pour évaluer l'allucination d'un texte généré par un modèle.

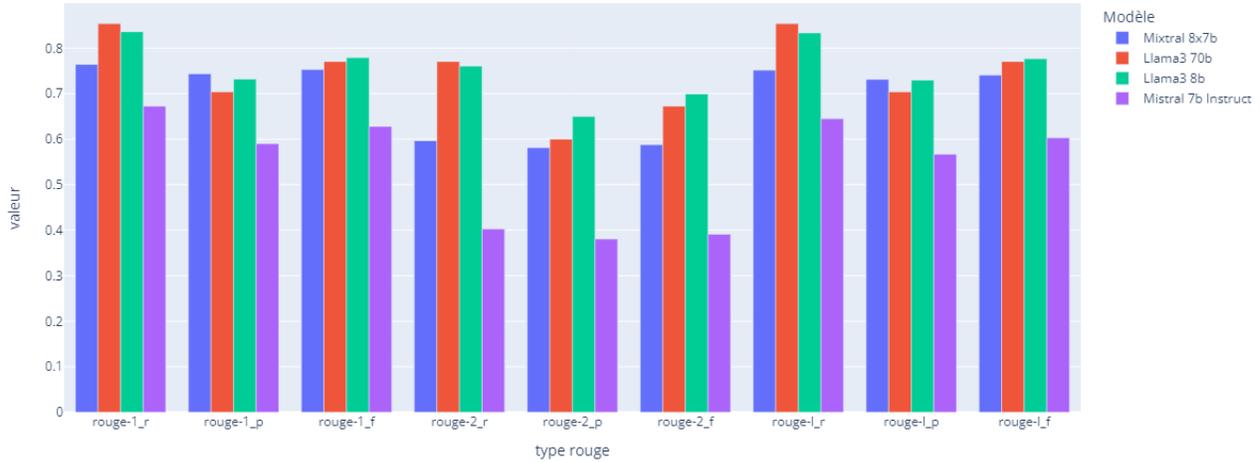


FIGURE 5.26: Les résultats du ROUGE Score moyen sur huit questions des modèles testés

Modèle	Score d'Hallucination
Mixtral 8x7b	0.9407
Llama3 70b	0.8933
Llama3 8b	0.8471
Mistral 7b Instruct	0.8360

TABLE 5.7: Scores d'hallucination pour les différents modèles

Le graphique de la figure 5.26 montre un comparaisn des Rouge scores moyens sur huit questions de la base de données des différents modèles utilisées avec l'approche RAG, en interprétant ces résultats, on remarque :

- **Llama3 70b (barres rouges)** : Ce modèle montre une performance légèrement supérieure dans la plupart des mesures ROUGE-1 F comparé aux autres modèles. Cela indique que Llama3 70b génère des réponses de meilleure qualité et plus fidèles au texte de référence.
- **Mixtral 8x7b (barres bleues) et Llama3 8b (barres vertes)** : Ces modèles ont des performances similaires dans les mesures ROUGE-1 F, bien que Llama3 8b montre une légère supériorité par rapport à Mixtral 8x7b dans certaines mesures.
- **Mistral 7b Instruct (barres violettes)** : Ce modèle affiche des scores légèrement inférieurs aux autres modèles dans la plupart des mesures ROUGE. Toutefois, il reste compétitif et montre une certaine stabilité dans ses performances.
- **Stabilité des Performances** : Les scores des modèles Mixtral 8x7b et Llama3 8b montrent moins de variation entre les différentes mesures ROUGE (rappel, précision, F-score) comparé à Llama3 70b. Cela pourrait suggérer que Mixtral 8x7b et Llama3 8b ont des performances plus stables, bien que légèrement inférieures à Llama3 70b.
- **Allucination Moyenne** :
 - Les scores d'hallucination (affichés sur le tableau 5.7) des modèles révèlent que Mixtral 8x7b a le score le plus élevé (0.9407), indiquant une très faible tendance à l'hallucination. Llama3 70b suit avec un score de 0.8933, puis Llama3 8b avec 0.8471, et enfin Mistral 7b Instruct avec

0.8360. Tous ces scores sont élevés, ce qui montre que les modèles ont une faible tendance à l'hallucination, bien que Mistral 7b Instruct montre une légère tendance plus élevée par rapport aux autres modèles.

5.3.5 Comparaison entre les deux méthodes utilisées :

- **Méthode de la DPO (Direct Preference Optimization) :** Cette technique requiert un entraînement du modèle pour lui permettre d'apprendre à générer la structure de réponse souhaitée de manière autonome. Cet entraînement consomme une quantité considérable de ressources computationnelles et nécessite un vaste ensemble de données pour être efficace. Elle requiert des ressources computationnelles substantielles ainsi qu'une grande quantité de données. Dans notre contexte, bien que ces conditions ne soient pas idéales, le modèle a réussi à ajuster ses poids de manière à s'adapter efficacement à la structure de réponse souhaitée et aux informations contenues dans la base de données. Néanmoins, la complexité et le temps requis pour l'entraînement complet peuvent limiter son application dans des environnements à ressources limitées ou sous contraintes de délai.
- **Méthode des RAGs (Retrieval-Augmented Generation) :** Utilisée pour affiner les capacités de génération du modèle à l'aide de données supplémentaires précises, cette méthode ne nécessite pas d'entraînement préalable du modèle complet, ce qui contribue à réduire les coûts et diminuer significativement le temps de mise en œuvre. De plus, les RAGs facilitent une meilleure généralisation en évitant le surapprentissage potentiel associé à des entraînements intensifs. Dans notre cas, cette approche s'est révélée être plus efficace que la DPO, le modèle parvenant à extraire correctement les informations et à reformuler les réponses conformément à la structure désirée.
- **Comparaison des performances :** Les tests ont révélé que les réponses générées par les RAGs étaient souvent plus pertinentes et précises, surtout lorsqu'il s'agissait de questions nécessitant une extraction spécifique d'informations.
- **Impact sur la scalabilité :** Les RAGs, grâce à leur moindre dépendance à des volumes massifs de données d'entraînement et à leur intégration flexible de sources d'informations externes, offrent une meilleure scalabilité par rapport à la DPO. Cette adaptabilité les rend appropriées pour l'expansion à de nouveaux domaines ou pour des applications nécessitant une mise à jour régulière des connaissances sans reformation complète du modèle.

5.4 Déploiement de la solution

Le déploiement représente une phase essentielle dans la réalisation d'un projet de développement d'un chatbot. Cette étape inclut l'intégration du modèle utilisant l'approche RAG, assurant ainsi une réponse efficace et pertinente aux clients de KPMG.

Dans le cadre spécifique de notre projet, nous avons orchestré le déploiement de nos modèles avec pour objectif de fournir une vue détaillée et approfondie de notre processus. Cette démarche met l'accent sur les aspects techniques tout en soulignant l'importance cruciale de l'expérience utilisateur.

5.4.1 Interface du chatbot

Notre solution est développée en utilisant la bibliothèque Streamlit, une bibliothèque Python qui permet de créer des interfaces utilisateur sophistiquées pour faciliter l'interaction entre l'utilisateur et nos modèles de génération de texte. Elle est conçue pour gérer efficacement les requêtes des utilisateurs, les transmettre à nos modèles en appliquant l'approche RAG, et renvoyer les réponses générées de manière fluide et efficace.

Cette application repose principalement sur trois composantes :

1. **L'interface du chatbot :** C'est le point de contact entre l'utilisateur et notre système. Conçue pour être intuitive et facile à utiliser, elle permet aux utilisateurs d'envoyer des prompts et de recevoir



FIGURE 5.27: Aperçu d'ensemble sur l'interface utilisateur

des réponses de manière simple et directe. L'interface est également responsable de présenter les résultats générés par le modèle de manière claire et lisible.

2. **Le choix du modèle** : Les utilisateurs ont la possibilité de sélectionner le modèle avec lequel ils souhaitent générer les réponses. Nous avons intégré quatre choix de modèles, chacun offrant des paramètres spécifiques comme le nombre maximum de tokens qu'il peut générer, le score d'hallucination, et le score ROUGE moyen. De plus, les utilisateurs peuvent ajuster la température du modèle pour contrôler le degré de créativité des réponses produites.
3. **L'utilisation de l'approche RAG** : Notre solution repose sur l'approche RAG, où le prompt est envoyé à la chaîne RAG pour effectuer une recherche dans notre base de données vectorielle et générer des réponses pertinentes avec le modèle LLM sélectionné.

5.4.2 Tests de l'interface

Notre plateforme se distingue par sa richesse en fonctionnalités et sa facilité d'utilisation. Elle interagit avec une base de données préalablement définie, contenant des questions-réponses basées sur les données internes de l'entreprise, offrant ainsi aux clients un accès instantané à une mine d'informations et de connaissances. La figure 5.27 donne un aperçu d'ensemble sur notre solution.

Lorsque le client choisit le modèle pour générer les réponses, l'interface affiche les caractéristiques de chaque modèle, telles que le nombre maximum de tokens qu'il peut générer, le score d'hallucination, ainsi que le score ROUGE moyen (figure 5.28). De plus, une fois le modèle sélectionné, le client peut ajuster la température du modèle pour contrôler le degré de créativité des réponses générées (figure 5.29).

Une fois ces paramètres choisis, le client peut poser une question au chatbot. Celui-ci répond en utilisant les paramètres sélectionnés et l'approche RAG pour fournir une réponse pertinente. Les figures 5.30, et 5.31 montrent des exemples de question posée par le client et la réponse proposée par notre chatbot.



FIGURE 5.28: Choix du modèle de génération



FIGURE 5.29: Ajustement de la température



FIGURE 5.30: Exemple de question d'un client de KPMG



FIGURE 5.31: Exemple de réponse générée par le chatbot

En cas de besoin de changer de discussion, le client a la possibilité d'effacer l'historique du chat en utilisant le bouton dédié "Effacer le chat", comme illustré dans la figure 5.32.



FIGURE 5.32: Possibilité d’effacer l’historique du chat

5.5 Conclusion

Ce chapitre a permis de concrétiser la solution proposée en mettant en œuvre les différentes étapes nécessaires à sa réalisation. La construction du dataset a impliqué la collecte de données issues de missions réelles dans le domaine du conseil bancaire, structurées sous forme de questions-réponses. Pour la modélisation, des modèles appropriés ont été sélectionnés en utilisant la méthode du prompt engineering. Ensuite, l’implémentation s’est déroulée selon deux approches : le reinforcement learning with human feedback (RLHF) et le retrieval augmented generation (RAG). Des tests ont été réalisés pour évaluer les performances des modèles sous les deux approches, suivis d’une analyse critique des résultats. Le déploiement de la solution a été effectué en concevant une architecture adaptée avec l’utilisation de la bibliothèque Streamlit, offrant ainsi une interface efficace et facile à utiliser. La réalisation de cette solution marque une étape significative dans l’atteinte des objectifs fixés. Les résultats obtenus attestent de l’efficacité des méthodologies adoptées et ouvrent de nouvelles perspectives pour l’application de cette solution dans des contextes réels.

Conclusion générale

Ce mémoire explore les dynamiques complexes du domaine émergent de l'IA générative, en mettant particulièrement l'accent sur son application dans le secteur spécifique du conseil bancaire. Notre objectif était de développer un chatbot capable de répondre efficacement aux questions des clients de KPMG, en utilisant une LLM pré-entraînée sélectionnée via la méthode du prompt engineering. Nous avons ensuite amélioré cette LLM choisie en comparant deux approches : le reinforcement learning with human feedback (RLHF) et le retrieval augmented generation (RAG). La solution finale a été déployée en utilisant l'approche RAG, qui s'est avérée particulièrement efficace pour générer des réponses pertinentes en exploitant des données internes de l'entreprise.

Dans la revue de littérature, nous avons exploré en détail les fondements du traitement du langage naturel, en introduisant les concepts clés de l'IA générative et des chatbots, ainsi que les principaux modèles de langage. Nous avons discuté des défis associés à ces modèles et des méthodes pour améliorer leurs performances, notamment à travers des techniques comme le prompt engineering et les méthodes d'entraînement avancées.

Nous avons également présenté KPMG en tant qu'organisme d'accueil, détaillant ses différents départements et services, ainsi que les processus actuels pour répondre aux appels d'offres des clients. Notre solution a été positionnée comme un ajout de valeur potentiel au système existant de KPMG, en permettant une réponse plus rapide et plus précise aux demandes des clients grâce à l'automatisation intelligente des réponses.

Ce travail représente une première étape vers l'intégration réussie de l'IA générative dans le secteur du conseil, spécifiquement dans le conseil bancaire. Pour optimiser davantage ce système, des efforts futurs seront nécessaires pour augmenter le volume de données disponibles, en particulier en généralisant l'application à d'autres secteurs du conseil et en incorporant des données spécifiques à ces domaines. L'utilisation de modèles de dernière génération tels que GPT-4 est également envisagée pour améliorer encore l'efficacité des réponses générées.

Malgré les progrès significatifs réalisés, plusieurs défis persistent, notamment la gestion des ressources nécessaires pour l'entraînement efficace de modèles de grande taille. Néanmoins, les résultats obtenus jusqu'à présent sont prometteurs et ouvrent la voie à des applications futures et à des innovations continues dans le domaine de l'IA générative appliquée au conseil.

Perspectives

Bien que notre solution ait démontré son efficacité à travers les résultats obtenus, il reste des améliorations à entreprendre pour en accroître l'efficacité et l'étendue. Notamment, une augmentation du volume de données d'entraînement avec l'approche DPO offrirait de meilleurs résultats en termes d'amélioration du modèle. De plus, générer des réponses rejetées par les experts permettrait de mieux distinguer une réponse incorrecte d'une réponse attendue. De même, l'utilisation de modèles de dernière génération produirait des résultats plus efficaces grâce à la qualité des réponses générées par le modèle, même avant l'entraînement.

Pour l'approche RAG, il serait bénéfique d'inclure plusieurs sources de données, notamment en utilisant des agents pour extraire des informations à partir du web lorsque celles-ci ne sont pas disponibles dans la base de données de l'entreprise. Ainsi, l'inclusion d'avantage de données offrirait une source d'informations plus large au LLM, résultant en des réponses plus précises.

À long terme, notre solution pourrait également s'étendre à d'autres secteurs du conseil en incluant des données spécifiques à ces secteurs, augmentant ainsi son éventail et son impact sur l'entreprise.

Références

Bibliographie

- [1] Humza N., et al. *A Comprehensive Overview of Large Language Models* . 09.04.2024. <https://arxiv.org/pdf/2307.06435>
- [2] Chin-Yew Lin. *ROUGE : A Package for Automatic Evaluation of Summaries*. Information Sciences Institute. <https://aclanthology.org/W04-1013.pdf>
- [3] Luca Ardito. *Design, Implementation and Evaluation of a Chatbot for Accounting Firm : A Fine-Tuning Approach with Two Novel Dataset*. École polytechnique de Turin, 04/2024.
- [4] Iulian, V. S., Chinnadhurai, S., Mathieu, G., Saizheng, Z., Zhouhan, L., Sandeep, S., Taesup, S., Pieper, M., Sarath, C., Nan, R., Ke, Sai, R., Alexandre, B., Jose, M., R., S., Dendi S., Vincent, M., Alexandre, M., Joelle, P1,2., Yoshua, B. *A Deep Reinforcement Learning Chatbot (Short Version)*. Montreal Institute for Learning Algorithms, Montreal, Quebec, Canada, 20.01.2018. <https://arxiv.org/pdf/1801.06700>
- [5] Siddharth, V., Mengjiao, Y., Justin, F., Sergey, L. *CHAI : A CHatbot AI for Task-Oriented Dialogue with Offline Reinforcement Learning*. 18.04.2022. <https://arxiv.org/pdf/2204.08426>
- [6] Redha T. *Essai de Pr evision des Indices du March  Boursier Tunisien : Approche par le R seau de Neurones*. https://www.researchgate.net/figure/Architecture-dun-reseau-multicouches_fig1_290394554
- [7] Zhilin, W., Yi, D., Jiaqi, Z., Virginia, A., Makesh, N. S., Daniel, E., Olivier, D., Jane, P. S., Neel, K., Aidan, S., Oleksii, K. *HELPSTEER : Multi-attribute Helpfulness Dataset for STEERLM*. NVIDIA, 16.11.2023. <https://arxiv.org/pdf/2311.09528>
- [8] Md, A., Biplob D. *LeanContext : Cost-efficient domain-specific question answering using LLMs*. Natural Language Processing Journal, 2024.
- [9] Jean K., Joshua H., Maximilian M., Herbie B., Roberta R., Robert D., M. *Challenges and Applications of Large Language Models*. University College London, University of Cambridge, 19.07.2023. <https://arxiv.org/pdf/2307.10169>
- [10] Sinan Ozdemir. *Quick Start Guide to Large Language Models : Strategies and Best Practices for using ChatGPT and Other LLMs*. Addison-Wesley, 20.09.2023.
- [11] Alec, R., Jeffrey, W., David, L., Dario, A., Ilya, S., Rewon, C. *Language Models are Unsupervised Multitask Learners*. https://d4mucfpksyvw.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
- [12] Jacob D., Ming-Wei C., Kenton L., Kristina T. *BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding*. <https://aclanthology.org/N19-1423.pdf>
- [13] Hugo T., Thibaut L., Gautier I., Xavier M., Marie-Anne L., Timothee L., Baptiste R., Naman G., Eric H., Faisal A., Aurelien R., Armand J., Edouard G., Guillaume L. *LLaMA : Open and Efficient Foundation Language Models*. <https://arxiv.org/pdf/2302.13971>
- [14] Jingfeng Y., Hongye J., Ruixiang T., Xiaotian H., Qizhang F., Haoming J., Bing Y., Xia H. *Harnessing the Power of LLMs in Practice : A Survey on ChatGPT and Beyond*. 27.04.2023. <https://arxiv.org/pdf/2304.13712>

- [15] Aleksandra P., Fabio P., Vladimir K., Naman G., Heinrich K., Mike L., Wen-tau Y., Tim R., Sebastian R., Douwe K. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Facebook AI Research, University College London, New York University, 19.07.2023. <https://arxiv.org/pdf/2307.10169>
- [16] Jia-Yu Y., Kun-Peng N., Zhen-Hui L., Mu-Nan N., Li Y. *LLM LIES : HALLUCINATIONS ARE NOT BUGS, BUT FEATURES AS ADVERSARIAL EXAMPLES*. Peking University, 04.10.2023. <https://arxiv.org/pdf/2310.01469>
- [17] Philip F., James R. F., Shimei P., ASRC F. *Trapping LLM “Hallucinations” Using Tagged Context Prompts*. University of Maryland, Baltimore County, 12.06.2023. <https://arxiv.org/pdf/2306.06085>
- [18] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, Pascale Fung. *Towards Mitigating Hallucination in Large Language Models via Self-Reflection*. Center for Artificial Intelligence Research (CAiRE), Hong Kong University of Science and Technology. <https://aclanthology.org/2023.findings-emnlp.123.pdf>
- [19] Zongxi L., Xianming L., Yuzhang L., Haoran X., Jing L., Fu-lee W., Qing L., Xiaoqin Z. *LABEL SUPERVISED LLAMA FINETUNING*. University of Maryland, Baltimore County, 02.10.2023. <https://arxiv.org/pdf/2310.01208>
- [20] Zheng Z., Chen Z., Da T., Ke S., Yukun M., Yingtong B., Xun Z., Liang Z. *Balancing Specialized and General Skills in LLMs : The Impact of Modern Tuning and Data Strategy*. Emory University, 07.10.2023. <https://arxiv.org/pdf/2310.04945>
- [21] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, Tat-Seng Chua. *Data-efficient Fine-tuning for LLM-based Recommendation*. National University of Singapore, Monash University, 30.01.2024. <https://arxiv.org/pdf/2401.17197>
- [22] Peter et al. *AUTOMATING RESEARCH SYNTHESIS WITH DOMAIN-SPECIFIC LARGE LANGUAGE MODEL FINE-TUNING*. National University of Singapore, Monash University, 30.01.2024. <https://arxiv.org/pdf/2401.17197>
- [23] Peter et al. *AUTOMATING RESEARCH SYNTHESIS WITH DOMAIN-SPECIFIC LARGE LANGUAGE MODEL FINE-TUNING*. National University of Singapore, Monash University, 30.01.2024. <https://arxiv.org/pdf/2401.17197>
- [24] Josef K., Samuel H. *Comparison Between RLHF and RLAIIF in Fine-Tuning a Large Language Model*. School of Electrical Engineering and Computer Science (EECS), 15.06.2023. <https://www.diva-portal.org/smash/get/diva2:1782683/FULLTEXT01.pdf>
- [25] Jagirdar S., USVSN Sai P., Aneech M., Vuppula A. *Raithubot : An RLHF Fine tuned Telugu chatbot for Farmers*. School of Electrical Engineering and Computer Science (EECS). https://books.google.dz/books?hl=fr&lr=&id=aWT8EAAAQBAJ&oi=fnd&pg=PA393&dq=Chatbot+using+RLHF&ots=XPBKRNh_Iz&sig=qXUXmr9-nPh-apobN97nS-zhs9k&redir_esc=y#v=onepage&q=Chatbot%20using%20RLHF&f=false
- [26] Avinash A., et al. *MM-PhyRLHF : Reinforcement Learning Framework for Multimodal Physics Question-Answering*. 19.04.2024. <https://arxiv.org/pdf/2404.12926>
- [27] John S., Filip W., Prafulla D., Alec R., Oleg K. *Proximal Policy Optimization Algorithms*. Bytedance Inc., 26.08.2017. <https://arxiv.org/pdf/1707.06347>
- [28] Rafael R., Archit S., Stefano E., Christopher D. M., Chelsea F. *Direct Preference Optimization : Your Language Model is Secretly a Reward Model*. Stanford University, 26.08.2017. https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf
- [29] Zhiyuan Z., Bin W., Linke O., Xiaoyi D., Jiaqi W., Conghui H. *Beyond Hallucinations : Enhancing LVLMS through Hallucination-Aware Direct Preference Optimization*. Shanghai AI Laboratory, 06.02.2024. <https://arxiv.org/pdf/2311.16839>

- [30] Matthias U., Sigurd S., Sudarshan K. B. *PHOENIX : OPEN-SOURCE LANGUAGE ADAPTION FOR DIRECT PREFERENCE OPTIMIZATION*. Shanghai AI Laboratory, 19.01.2023. <https://arxiv.org/pdf/2401.10580>
- [31] Lewis T., Edward B., Nathan L., Nazneen R., Kashif R., Younes B., Shengyi H., Leandro V. W., Clementine F., Nathan H., Nathan S., Omar S., Alexander M. R., T. W. *ZEPHYR : DIRECT DISTILLATION OF LM ALIGNMENT*. 25.10.2023. <https://arxiv.org/pdf/2310.16944>
- [32] Yunfan G., Yun X., Xinyu G., Kangxiang J., Jinliu P., Yuxi B., Yi D., Jiawei S., Meng W., Haofen W. *Retrieval-Augmented Generation for Large Language Models : A Survey*. 27.03.2024. <https://arxiv.org/pdf/2312.10997v5>
- [33] Penghao Z., Qinhan YX., Hailin Z., Penghao Z. *Retrieval-Augmented Generation for AI-Generated Content : A Survey*. 31.05.2024. <https://arxiv.org/pdf/2402.19473>
- [34] Patrick L., et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Facebook AI Research, University College London, New York University. https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf
- [35] *A RAG-based Question Answering System Proposal for Understanding Islam : MufassirQAS LLM*. <https://arxiv.org/pdf/2401.15378>
- [36] Huayang L., Yixuan S., Deng C., Yan W., Lemaol L. *A Survey on Retrieval-Augmented Text Generation*. 13.02.2022. <https://arxiv.org/pdf/2202.01110>
- [37] Jiawei C., Hongyu L., Xianpei H., Le S. *Benchmarking Large Language Models in Retrieval-Augmented Generation*. 20.12.2023. <https://arxiv.org/pdf/2309.01431>
- [38] Bhavith Chandra Challagundla, Gururanga Ramanujam B. *Financial Advisory LLM Model for Modernizing Financial Services and Innovative Solutions for Financial Literacy in India*. 03.05.2024. https://assets-eu.researchsquare.com/files/rs-4354348/v1_covered_341a77ab-9100-4c87-b9df-b5c2ee75aeb0.pdf?c=1714739682
- [39] Chloe C. H., Celestine M.D., Moritz H. *Revisiting Design Choices in Proximal Policy Optimization*. 23.09.2020. https://assets-eu.researchsquare.com/files/rs-4354348/v1_covered_341a77ab-9100-4c87-b9df-b5c2ee75aeb0.pdf?c=1714739682
- [40] Bram W., et al. *Diffusion Model Alignment Using Direct Preference Optimization*. Salesforce AI, Stanford University, 21.11.2023. <https://arxiv.org/pdf/2311.12908>
- [41] Ashish V., et al. *Attention Is All You Need*. 02.08.2023. <https://arxiv.org/pdf/1706.03762>
- [42] Kolla B. P., et al. *Chatterbot implementation using Transfer Learning and LSTM Encoder-Decoder Architecture*. 05.05.2020. https://dlwqtxts1xzle7.cloudfront.net/63554444/ijeter35852020200607-129940-svie0d-libre.pdf?1591542320=&response-content-disposition=inline%3B+filename%3DChatterbot_implementation_using_Transfer.pdf&Expires=1718838735&Signature=QAGz1tYzw34b-rW-jgQiukGPXxYnHv99j9e76fkeHm0XbMQCTnZrQda09s-aXS74V9kQC5s1hYZAxS0hKBJTq50yaF~qj__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA
- [43] Tapas N., Hwee T. N. *Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction*. Department of Computer Science, National University of Singapore.
- [44] *Qu'est-ce que le traitement automatique du langage naturel (NLP) ?*. <https://www.ibm.com/fr-fr/topics/natural-language-processing>
- [45] *Natural Language Processing : Les Chatbots Expliqués*. <https://www.webotit.ai/knowledge/natural-language-processing-les-chatbots-expliques#:~:text=Le%20r%C3%B4le%20du%20NLP%20dans,r%C3%A9pondre%20aux%20requ%C3%AAtes%20des%20utilisateurs>.
- [46] *Prompt Engineering Guide*. <https://www.promptingguide.ai/fr>
- [47] Kyunghyun C., Bart V. M., Caglar G., Dzmitry B., Fethi B., Holger S., Yoshua B. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 03.09.2014. <https://arxiv.org/pdf/1406.1078>

- [48] Junnan L., Dongxu L., Silvio S., Steven H. *BLIP-2 : Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. 15.01.2023. <https://arxiv.org/pdf/2301.12597v3>
- [49] Shengjie S., Jin L., Yihe Y. *Multi-Layer Transformer Aggregation Encoder for Answer Generation*. 26.05.2020. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9091180>
- [50] Masato Fujitake. *DTrOCR : Decoder-only Transformer for Optical Character Recognition*. 30.08.2023. <https://arxiv.org/pdf/2308.15996>
- [51] Tom B. B., Benjamin M., Nick R., Melanie S., et al. *Language Models are Few-Shot Learners*. 22.07.2020. <https://arxiv.org/pdf/2005.14165>
- [52] Zihao F., Haoran Y., Anthony M. S., Wai L., Lidong B., Nigel C. *On the Effectiveness of Parameter-Efficient Fine-Tuning*. 28.11.2022. <https://arxiv.org/pdf/2211.15583>
- [53] Vladislav L., Vijeta D., Anna R. *Scaling Down to Scale Up : A Guide to Parameter-Efficient Fine-Tuning*. 28.03.2023. <https://arxiv.org/pdf/2303.15647>
- [54] *What is fine-tuning ?*. <https://www.ibm.com/topics/fine-tuning>
- [55] *Reinforcement Learning from Human Feedback (RLHF) - a simplified explanation*. <https://pub.towardsai.net/reinforcement-learning-from-human-feedback-rlhf-f88687d5402e>
- [56] *A Comprehensive Guide to Proximal Policy Optimization (PPO) in AI*. <https://medium.com/@oleglatypov/a-comprehensive-guide-to-proximal-policy-optimization-ppo-in-ai-82edab5db200>
- [57] *Kullback-Leibler Divergence*. <https://hanj.cs.illinois.edu/cs412/bk3/KL-divergence.pdf>
- [58] *Direct Preference Optimization (DPO)*. <https://medium.com/@joaolages/direct-preference-optimization-dpo-622fc1f18707>
- [59] *Creare un Chatbot in Python con LangChain e RAG*. <https://www.diariodiunanalista.it/posts/chatbot-python-langchain-rag/>
- [60] Jiawei C., Hongyu L., Xianpei H., Le S. *Benchmarking Large Language Models in Retrieval-Augmented Generation*. 20.12.2023 <https://arxiv.org/pdf/2309.01431>
- [61] *Understanding Encoder-Decoder Sequence to Sequence Model*. <https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346>
- [62] *The Illustrated Transformer*. <https://jalammar.github.io/illustrated-transformer/>
- [63] Yang L., Dan I., Yichong X., Shuohang W., Ruochen X., Chenguang Z. *G-EVAL : NLG Evaluation using GPT-4 with Better Human Alignment*. 23.05.2023. <https://arxiv.org/pdf/2303.16634>
- [64] Peiyi W. et al. *Large Language Models are not Fair Evaluators*. 30.08.2023. <https://arxiv.org/pdf/2305.17926>
- [65] *vectara/hallucination_evaluation_model*. https://huggingface.co/vectara/hallucination_evaluation_model
- [66] *meta-llama/Meta-Llama-3-8B*. <https://huggingface.co/meta-llama/Meta-Llama-3-8B>
- [67] *meta-llama/Meta-Llama-3-70*. <https://huggingface.co/meta-llama/Meta-Llama-3-70B>
- [68] *mistralai/Mistral-7B-Instruct-v0.3*. <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3>
- [69] *google/gemma-1.1-7b-i*. <https://huggingface.co/google/gemma-1.1-7b-it>
- [70] *tiiuae/falcon-7b*. <https://huggingface.co/tiiuae/falcon-7b>
- [71] *mistralai/Mixtral-8x7B-v0.1*. <https://huggingface.co/mistralai/Mixtral-8x7B-v0.1>

Annexes

Annexe A : Bibliothèques utilisées

A.1 Présentation de Python

Python est un langage de programmation à haut niveau, interprété et orienté objet, créé par Guido van Rossum et publié pour la première fois en 1991. Il est devenu l'un des langages de programmation les plus populaires et les plus utilisés, apprécié pour sa lisibilité et son code clair. Python est souvent décrit comme un langage proche de l'anglais simple, ce qui rend son code particulièrement facile à lire et à comprendre.

Un autre avantage majeur de Python est sa polyvalence. Il est utilisé dans divers domaines, y compris le développement web, le développement de logiciels, l'analyse de données, l'apprentissage automatique, l'intelligence artificielle, la science des données, la bio-informatique, le calcul scientifique, et bien d'autres. Son usage est également très répandu dans le domaine du Traitement Automatique du Langage Naturel (NLP), grâce à sa lisibilité, sa simplicité d'utilisation et la variété de ses bibliothèques dédiées. Les applications de Python en NLP vont de la génération de texte à la traduction automatique, en passant par l'analyse de sentiments et le résumé automatique de textes.

A.2 Bibliothèques Utilisées

Les bibliothèques en programmation sont des ensembles d'outils, de modules, de méthodes et de classes qui étendent les fonctionnalités du langage utilisé. La plupart des bibliothèques disponibles en Python sont gratuites et open source. Voici les bibliothèques utilisées dans ce projet :

A.2.1 Transformers

Transformers est une bibliothèque qui fournit des API et des outils pour télécharger et entraîner facilement des modèles pré-entraînés de pointe. Ces modèles supportent des tâches courantes dans différentes modalités, telles que :

- **Traitement du Langage Naturel (NLP)** : classification de texte, reconnaissance d'entités nommées, questions-réponses, modélisation de langage, résumé, traduction, choix multiples, et génération de texte.
- **Vision par Ordinateur** : classification d'images, détection d'objets, et segmentation.
- **Audio** : reconnaissance automatique de la parole et classification audio.
- **Multimodal** : questions-réponses sur table, reconnaissance optique de caractères, extraction d'informations à partir de documents scannés, classification vidéo, et questions-réponses visuelles.

Transformers supporte l'interopérabilité des frameworks entre PyTorch, TensorFlow et JAX, offrant la flexibilité d'utiliser différents frameworks à chaque étape de la vie d'un modèle.

A.2.2 LangChain

LangChain est un cadre pour développer des applications alimentées par de grands modèles de langage (LLMs). LangChain simplifie chaque étape du cycle de vie de l'application LLM, que ce soit pour :

- **Développement** : construire des applications en utilisant les blocs de construction et composants open-source de LangChain, intégrations tierces et modèles.
- **Mise en production** : utiliser LangSmith pour inspecter, surveiller et évaluer vos chaînes, optimisant et déployant en toute confiance.
- **Déploiement** : transformer toute chaîne en une API avec LangServe.

LangChain comporte plusieurs modules pour assurer le bon fonctionnement des composants nécessaires à la création d'applications NLP, tels que l'interaction avec les modèles, la connexion et récupération des données, les chaînes, les agents, et la mémoire.

Exemples d'Application de LangChain

Grâce à sa capacité à construire des séquences ordonnées d'actions avec les LLM au cœur du sujet, LangChain se révèle puissant dans de nombreux projets basés sur le NLP. Une des principales applications de LangChain concerne la récupération d'informations (*retrieval*), c'est-à-dire la récupération d'informations contenues dans diverses sources de données. Ce mécanisme est à la base du RAG (*Retrieval-Augmented Generation*), une méthode qui intègre un mécanisme de recherche d'informations pour permettre à un LLM de disposer d'une base de connaissance spécifique à un domaine ou pour une entreprise donnée, ce système est illustre dans la figure [A.1](#).

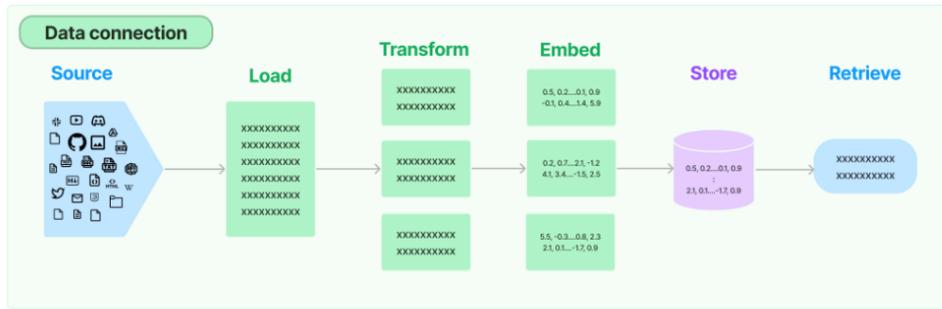


FIGURE A.1: Récupération d'informations avec la bibliothèque Langchain

A.2.3 Hugging Face Hub

La bibliothèque `huggingface_hub` permet d'interagir avec le Hugging Face Hub, une plateforme de machine learning pour les créateurs et les collaborateurs. Elle permet de découvrir des modèles pré-entraînés et des jeux de données pour vos projets, de jouer avec des applications de machine learning hébergées sur le Hub, et de créer et partager vos propres modèles et jeux de données avec la communauté.

A.2.4 Groq

Groq est une bibliothèque permettant l'utilisation de Groq Cloud avec LangChain, facilitant l'utilisation des différentes fonctionnalités de LangChain avec les modèles de Groq Cloud.

A.2.5 Rouge

Rouge est un package qui permet d'importer la métrique d'évaluation Rouge Score pour évaluer les textes générés par les LLMs en les comparant à des textes de référence. Le score ROUGE repose sur le rappel et la précision, généralement rapporté sous la forme d'un score F1, qui est la moyenne harmonique du rappel et de la précision.

A.2.6 PyTorch

PyTorch est une bibliothèque d'apprentissage en profondeur pour le langage Python, développée principalement par l'équipe de recherche en intelligence artificielle de Facebook. Elle est connue pour sa flexibilité et sa facilité d'utilisation, offrant un contrôle dynamique de la structure des réseaux de neurones et une intégration forte avec l'écosystème Python.

A.3 Outils Utilisés

A.3.1 Groq

Groq est une entreprise spécialisée dans l'inférence des grands modèles de langage (LLM). Au moment de la rédaction de ce mémoire, Groq revendique des performances d'inférence 18 fois plus rapides sur le tableau de bord LLMPerf d'Anyscale par rapport aux autres principaux fournisseurs basés sur le cloud. Groq propose actuellement des modèles tels que Llama 2 70B et Mixtral 8x7B de Meta AI, ainsi que LLama3 8b et LLama3 70B, via leurs API. Ces modèles sont alimentés par le moteur d'inférence Groq LPU™, conçu avec leur propre matériel personnalisé appelé unités de traitement de langage (LPUs). La figure A.2 ci-dessous compare le débit des jetons de sortie (jetons/s), qui correspond au nombre moyen de jetons de sortie renvoyés par seconde. Les nombres dans le graphique correspondent au débit moyen des jetons de sortie (basé sur 150 requêtes) des fournisseurs d'inférence LLM sur le modèle Llama 2 70B.

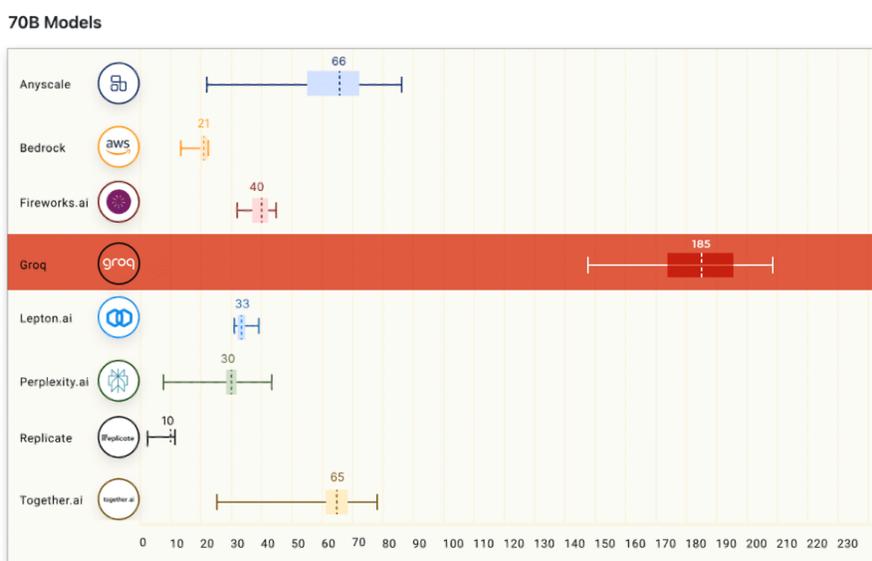


FIGURE A.2: Comparaison du débit moyen des jetons par seconde du modèle Llama2 70B sur 150 requêtes

A.3.2 Hugging Face

Hugging Face est une startup française créée en 2015 par Julien Chaumond et Clément Delangue, visant à rendre l'intelligence artificielle accessible à tous. Hugging Face propose une bibliothèque open-source pour le traitement du langage naturel (NLP), offrant à ses utilisateurs une API facilitant l'accès à des modèles pré-entraînés. Cela simplifie l'apprentissage et l'expérimentation, et inclut des outils pour la gestion des données et des modèles existants, ainsi que pour le développement et l'entraînement de nouveaux modèles.

Au fil des années, Hugging Face a développé une gamme de produits innovants basés sur l'IA :

- **Bibliothèque Transformers** : Transformers est une bibliothèque conçue pour entraîner et déployer des modèles NLP basés sur Python, permettant de réaliser diverses tâches comme la classification, la génération de texte, la détection d'entités nommées, l'extraction d'informations, et les questions-réponses.
- **Accelerate** : Accelerate est une API permettant aux développeurs et data scientists d'exécuter leurs scripts et de coder leurs propres boucles d'entraînement dans différents types de configurations.
- **Spaces** : Spaces est une plateforme communautaire où les membres peuvent partager leurs applications

de machine learning, créer directement leurs applications avec Hugging Face et collaborer plus facilement grâce au contrôle des versions.

- **Chatbots** Hugging Face s'est initialement fait connaître grâce à ses applications de chatbot, telles que Chatty>, Talking Dog, Talking Egg, et Boloss. Bien que ce ne soit plus leur cœur de métier, ces applications démontrent les capacités des modèles NLP développés par l'entreprise.

A.3.3 Google Colab

Google Colab, ou Google Colaboratory, est une plateforme gratuite offerte par Google permettant d'écrire et exécuter du code Python directement dans le navigateur. Elle facilite l'accès aux ressources de calcul et aux bibliothèques d'apprentissage automatique usuelles sans nécessiter d'installation locale. Voici quelques caractéristiques clés de Google Colab :

- **Accessibilité** Google Colab est hébergé sur le cloud, permettant de l'utiliser sans installer Python ou d'autres bibliothèques sur son ordinateur. Il suffit de se connecter à un compte Google pour y accéder. De plus, Google Colab est livré avec de nombreuses bibliothèques Python pré-installées, incluant des bibliothèques de data science et de visualisation telles que NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch, Matplotlib, Seaborn, et Plotly.
- **Accès aux ressources de calcul** Google Colab offre un accès gratuit à des processeurs graphiques (GPU) et à des unités de traitement de tenseur (TPU), ce qui est extrêmement utile pour les tâches gourmandes en calcul.
- **Facilité de collaboration** Les notebooks créés dans Google Colab sont sauvegardés automatiquement sur Google Drive, permettant un stockage et un partage faciles avec d'autres développeurs. Google Colab prend en charge la collaboration en temps réel, permettant à plusieurs utilisateurs de travailler simultanément sur le même notebook. Il permet également l'utilisation de GitHub pour le suivi des versions de code et la collaboration.
- **Flexibilité dans la programmation** Google Colab permet d'accéder facilement à des données externes et d'inclure des cellules de texte explicatif pour documenter le travail. Les notebooks offrent plusieurs modes d'exécution, incluant l'exécution d'une seule cellule, d'une sélection de cellules, ou de tout le notebook en une seule fois.

En résumé, Google Colab est un outil puissant pour le développement en Python, particulièrement dans le domaine de l'apprentissage automatique. Il offre un environnement de développement interactif, un accès aux GPU et TPU, un stockage sur Google Drive, une collaboration en temps réel, et de nombreuses autres fonctionnalités avancées.

A.3.4 Jupyter Notebook

Le Jupyter Notebook est une application web open-source permettant de créer et de partager des documents interactifs, autrefois appelés IPython Notebooks. Cet environnement de calcul interactif basé sur le web permet aux utilisateurs de créer des documents notebooks, qui sont des cellules de codes pouvant être documentées avec du texte ou des données. Voici quelques caractéristiques clés de Jupyter :

- **Support de plusieurs langages de programmation** Jupyter prend en charge plusieurs langages de programmation, notamment Python, R, Julia, et bien d'autres, permettant d'écrire et d'exécuter du code directement dans les cellules des notebooks.
- **Exécution interactive du code** Les notebooks Jupyter permettent d'exécuter du code de manière interactive, cellule par cellule, facilitant l'exploration des données et les analyses itératives. Les résultats des calculs et les visualisations sont affichés en temps réel.

- **Intégration de documentation** Les notebooks Jupyter permettent d'inclure des textes explicatifs, des titres, des sous-titres et des commentaires au sein des cellules de texte, fournissant des explications détaillées et documentant le code.
- **Visualisations interactives** Jupyter offre des bibliothèques et des outils de visualisation tels que Matplotlib, Seaborn, et Plotly, permettant de créer des graphiques et des tableaux de bord interactifs directement dans les notebooks.

En résumé, Jupyter est un environnement de développement interactif populaire qui facilite l'écriture, l'exécution et le partage de code, ainsi que la création de documents combinant du code, des visualisations et des explications. C'est un outil polyvalent utilisé par de nombreux développeurs, scientifiques et analystes de données pour explorer, analyser et communiquer des informations.

Annexe B : Notions avancés sur l'Apprentissage par Renforcement à partir de Feedback Humain

B.1 Apprentissage par Renforcement à partir de Feedback Humain (RLHF)

L'Apprentissage par Renforcement à partir de Feedback Humain (RLHF) représente une avancée significative dans le domaine de l'intelligence artificielle, exploitant les préférences et les orientations humaines pour former et améliorer les modèles d'apprentissage machine. RLHF est une extension de l'apprentissage par renforcement, une technique de formation basée sur la récompense et la punition pour les modèles d'IA, qui se distingue par l'introduction du feedback humain pour assurer que les modèles se comportent de manière sûre, éthique et souhaitable. Contrairement à l'apprentissage par renforcement traditionnel, où les modèles apprennent à partir de récompenses générées par des interactions avec un environnement, le RLHF utilise le feedback humain comme source précieuse de guidage. Ce feedback peut aider les systèmes d'IA à naviguer dans des espaces de décision complexes, à s'aligner sur les valeurs humaines et à effectuer des choix plus informés et éthiques.

B.1.1 Pourquoi utiliser l'Apprentissage par RLHF

L'Apprentissage par Renforcement à partir de Feedback Humain (RLHF) est devenu un concept significatif et influent dans le domaine de l'intelligence artificielle (IA), offrant plusieurs avantages clés :

1. **IA centrée sur l'humain** : Le RLHF aide à créer des systèmes d'IA qui respectent et comprennent les valeurs humaines, rendant les applications plus sûres et plus pertinentes.
2. **Amélioration des récompenses** : Il permet d'utiliser le feedback humain pour guider l'apprentissage de l'IA, offrant une alternative intuitive pour les tâches complexes où la spécification des récompenses est difficile.
3. **Expérience utilisateur améliorée** : Cette méthode adapte les systèmes d'IA aux préférences des utilisateurs, améliorant la personnalisation et la satisfaction, et évitant les problèmes d'hallucinations des LLMs.
4. **Applications variées** : Le RLHF est applicable dans plusieurs domaines comme la robotique et le traitement du langage naturel, enrichissant la polyvalence de l'IA.
5. **Sécurité et fiabilité** : Il contribue à la création de systèmes d'IA fiables et sécurisés, en minimisant les risques de comportements indésirables.

Ces aspects font du RLHF une approche précieuse pour le développement responsable et efficace des technologies d'intelligence artificielle.

B.1.2 Processus d'implémentation du RLHF

L'Apprentissage par Renforcement à partir de Feedback Humain (RLHF) est particulièrement utile lorsque la création d'une fonction de perte adéquate est difficile, comme pour évaluer l'humour, ou lorsque l'étiquetage des données de production est complexe, tel que pour les données de ChatGPT. Ce processus d'implémentation se divise en trois étapes principales [55] :

1. le pré-entraînement d'un modèle de langage.
2. l'entraînement d'un modèle de récompense basé sur le feedback humain.
3. le fine-tuning du modèle de langage avec l'apprentissage par renforcement.

1. Le pré-entraînement d'un modèle de langage :

La première étape du processus RLHF consiste à choisir un modèle de langage, souvent un modèle pré-entraîné tel que BERT ou LLaMA, en raison de la complexité et des ressources nécessaires pour former un modèle à partir de zéro.

Une étape facultative mais utile qui suit est le Supervised Fine-Tuning (SFT), où le modèle est ajusté pour mieux répondre à des prompts spécifiques via des exemples annotés par des humains. Cette phase prépare le modèle à générer des réponses plus alignées avec les attentes des utilisateurs, établissant une fondation solide avant d'introduire des ajustements plus ciblés grâce au RLHF.

Cette première phase s'achève donc avec un modèle formé ou ajusté (Figure B.1), prêt à être encore amélioré via le RLHF pour répondre précisément aux exigences posées par les interactions humaines.

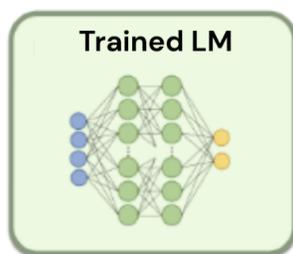


FIGURE B.1: Notre modèle de langage [55]

2. L'entraînement d'un modèle de récompense :

La deuxième étape du processus RLHF implique l'entraînement d'un modèle de récompense, qui est crucial pour l'optimisation des performances du modèle de langage principal. Voici un résumé des étapes clés de cette phase :

- Processus de collecte de données : Cette étape consiste à rassembler des données sous forme de triplets (texte d'entrée, texte de sortie, récompense). Pour ce faire, les données d'entrée sont utilisées pour générer des sorties via le modèle initial. Ces sorties sont ensuite évaluées par des humains

qui attribuent des récompenses selon la pertinence et la qualité des réponses. Les récompenses, attribuées par des annotateurs humains, peuvent varier de 0 à 5 ou être binaires (0/1), comme illustré dans la Figure B.2. Les annotateurs humains jouent un rôle crucial en générant le dataset d'entraînement et en classant les réponses selon leurs préférences

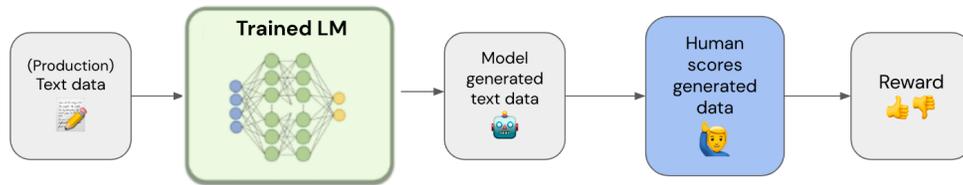


FIGURE B.2: Pipeline de collecte de données [55]

- Formation du modèle de récompense : Avec les données collectées, un nouveau modèle de langage est entraîné dont le but est d'associer des paires de texte (entrée, sortie) à une valeur scalaire de récompense. Ce modèle appelé modèle de récompense (Figure ??) agit comme un outil d'alignement, évaluant les sorties du modèle de base et fournissant un signal de récompense utilisé pour optimiser les paramètres du modèle de langage principal.

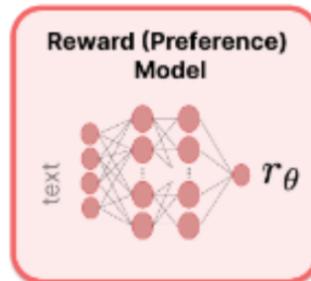


FIGURE B.3: Modèle de récompense [55]

En pratique, le modèle de récompense aide à surmonter les limitations de l'apprentissage par renforcement direct, telles que la lenteur, le coût, et la contre-productivité potentielle de l'entraînement avec de grands volumes d'échantillons d'entraînement. En alignant ses systèmes de récompenses avec les modèles de préférences humaines, le modèle de récompense permet une optimisation plus fine et contextuellement appropriée du modèle de langage principal.

3. Fine-tuning du LM avec l'apprentissage par renforcement :

L'étape finale du processus d'Apprentissage par Renforcement à partir de Feedback Humain (RLHF) implique le fine-tuning du modèle de langage en utilisant des techniques de renforcement. Voici les principales étapes de ce processus comme illustre la Figure B.4 :

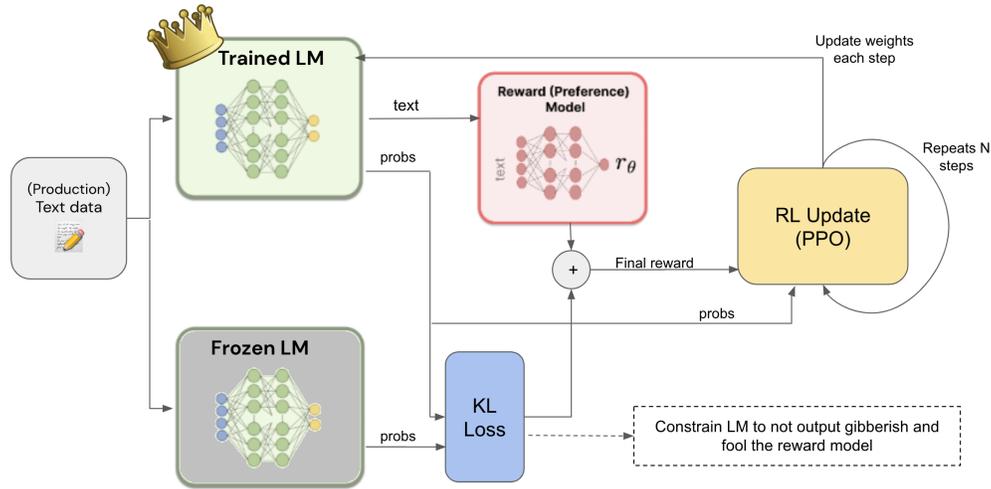


FIGURE B.4: Ajustement fin du modèle de langage principal en utilisant le modèle de récompense et le calcul de perte PPO. [55]

1. **Préparation du Modèle** : Une copie exacte du modèle de langage (LM) est créée, et ses poids entraînaables sont gelés. Cette mesure préventive aide à maintenir la stabilité du modèle durant l'apprentissage par renforcement.
2. **Calcul de la Divergence KL** : La divergence de Kullback-Leibler (KL) est calculée entre les probabilités de sortie de texte du LM gelé et celles du LM actif. Cette divergence mesure l'écart entre les comportements prévus et les comportements actuels du modèle [57]. La formule détaillée est présentée dans l'équation 1.

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \quad (1)$$

où P représente la distribution des probabilités du modèle original, et Q celle du modèle ajusté.

3. **Intégration de la Récompense** : La perte KL est combinée avec la récompense produite par le modèle de récompense. Cette intégration crée un signal complet qui guide l'ajustement du modèle.
4. **Application de l'Optimisation de Politique Proximale (PPO)** :

À cette étape, l'Optimisation de Politique Proximale (PPO) est appliquée pour ajuster finement les poids du modèle. Cette technique de renforcement équilibre stratégiquement l'exploration de nouvelles réponses et l'exploitation des réponses éprouvées pour optimiser les performances.

Nous explorant plus en détails l'algorithme du PPO dans la section précédente.

B.1.3 Un regard approfondi sur l'Optimisation de Politique Proximale (PPO)

L'Optimisation de Politique Proximale (PPO) est une technique d'apprentissage par renforcement qui optimise les politiques de manière à ce qu'elles ne s'éloignent pas trop de la politique précédente. Il apprend directement à partir des interactions de son agent avec l'environnement, Une fois qu'un lot d'expérience est utilisé pour une mise à jour du gradient, il est écarté, et la politique avance.

Fonction de perte de PPO

La fonction de perte de PPO peut être décomposée en les composantes suivantes comme le montre la Figure B.5 :

$$L^{PPO} = L^{POLICY} + c_1 L^{VF} + c_2 L^{ENT}$$

Policy loss Value loss Entropy loss

FIGURE B.5: Fonction de perte de PPO [56]

Perte de gradient de politique (Policy Gradient Loss) : Cette perte maximise la probabilité de prendre de bonnes actions. Elle est calculée en comparant la nouvelle politique (π) à l'ancienne politique (π_o) en utilisant l'échantillonnage d'importance et est limitée pour assurer que les mises à jour restent proches de la politique originale. La fonction de clip aide à contrôler la taille des pas de mise à jour, empêchant ainsi les mises à jour importantes qui pourraient nuire à la performance de l'agent, garantissant ainsi l'optimisation de politique "proximale" [27]. La formule détaillée est présentée dans l'équation 2.

$$\text{Perte de politique : } \mathbb{E} [\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)] \quad (2)$$

- où $r_t(\theta)$ est le rapport des probabilités selon les nouvelles et anciennes politiques.
- A_t est la récompense + la divergence KL précédemment calculée et passée à l'algorithme PPO.
- ϵ est un hyperparamètre qui contrôle le degré de région de confiance dans PPO.

La Figure B.6 schématise en détail le calcul de la perte de gradient de politique.

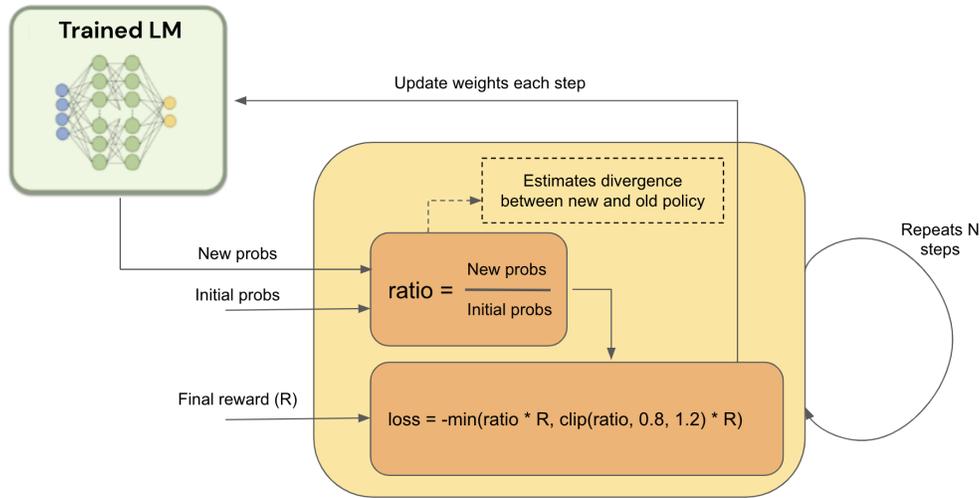


FIGURE B.6: Zoom sur le calcul du Policy loss [55]

Perte de fonction de valeur (Value Function Loss) : Cette perte minimise l'erreur dans l'estimation de valeur. La fonction de valeur prédit la récompense future à partir d'un état, conduisant à une meilleure prise de décision par l'agent. Elle est calculée comme l'erreur quadratique moyenne entre la valeur estimée et la valeur observée :

$$\text{Perte de valeur} : c_1 \mathbb{E}_t[(V_w(s_t) - V_t)^2] \quad (3)$$

où $V_w(s_t)$ est la valeur estimée de l'état actuel, V_t est le retour observé.

Bonus d'entropie (Entropy Bonus) : Ce terme encourage l'exploration en ajoutant un bonus pour des politiques plus aléatoires. L'entropie d'une politique est une mesure de sa randomisation, utilisée pour décourager la convergence prématurée vers une politique sous-optimale. La perte d'entropie est donnée par :

$$\text{Perte d'entropie} : c_2 \mathbb{E}_t[S\pi_\theta] \quad (4)$$

où $S\pi_\theta$ est le bonus d'entropie qui est une mesure de la randomisation de la politique.