Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

# Construire la confiance des ChatBots à base de LLM

# Building confidence of LLM based ChatBots

**Nicolas Rémy**
*Direction Ingénierie et Performance des Systèmes*
*LGM*
*Vélizy, France*
nicolas.remy@lgm.fr

**Frédéric Deschamps**
*Direction Technique et Innovation*
*LGM*
*Toulouse, France*
frederic.deschamps@lgm.fr

**Stéphane Kreckelbergh**
*Direction Digital & Software*
*LGM*
*Tarnos, France*
stephane.kreckelbergh@lgm.fr

***Résumé*** *— Les LLM révolutionnent l'innovation en Intelligence Artificielle, en mettant à disposition des composants génériques de traitement du langage naturel utilisables facilement dans de multiples applications. Une des applications cible est l'utilisation d'un LLM pour interroger une base documentaire en dialoguant en langage naturel et ouvert, révolutionnant ainsi les solutions de ChatBots précédentes. Néanmoins, ces nouveaux composants comportent un certain nombre de limitations technologiques qui impactent directement la confiance et la robustesse de ces systèmes.*

*Cet article fait ainsi un tour d'horizon des architectures de ces nouveaux ChatBots, des limitations et difficultés rencontrées, et de quelques recommandations applicables, nécessaires pour maitriser le déploiement de ces projets, en mettant en exergue deux problématiques de ces systèmes : La structuration du corpus documentaire et la fonctionnalité dite de RAG, essentielles à la confiance et à la robustesse de ces systèmes pour pallier les insuffisances actuelles des LLM.*

***Mots-clefs — IA, LLM, RAG, ChatBot, Safety, Confiance***

***Abstract —*** *LLMs have revolutionized innovation in Artificial Intelligence, by providing generic natural language processing components that can be easily integrated in a wide range of applications. One of the target applications is the use of an LLM to interrogate a document database, thus dialoguing in natural and open language, revolutionizing previous ChatBot solutions. However, these new components have a number of technological limitations having a direct impact on the confidence and robustness of these systems.*

*This article provides an overview of the architectures of these new ChatBots, the limitations and difficulties encountered, and some applicable best practices needed to manage the deployment of these projects, highlighting two issues for these systems: the structuring of the document database and the so-called RAG functionality, which are essential to the trust and robustness of these systems in order to overcome the current shortcomings of LLMs.*

***Keywords —*** ***AI, LLM, RAG, ChatBot, Safety, Trust***

## I. INTRODUCTION

L'objectif de cette communication est de décrire des stratégies d'évaluation et d'atténuation des risques fonctionnels des logiciels à base d'Intelligence Artificielle dite « Générative », et en particulier ceux basés sur l'utilisation de LLM (Large Language Model), dont le développement exceptionnel actuel au niveau mondial est sans précédent, et tend à se développer pour de multiples applications grand public et professionnelles.

En effet, ces nouveaux algorithmes de NLP/TAL (NLP – Natural Language Processing/TAL – Traitement du Langage Naturel) basés sur des modèles de Machine Learning de taille massive (plusieurs dizaines de milliards de paramètres), apportent des capacités puissantes de compréhension et de génération du langage, mais par ailleurs posent des difficultés de confiance et de robustesse des réponses fournies, avec de surcroît une très bonne qualité grammaticale et orthographique qui peut fausser l'appréciation de l'utilisateur sur la véracité et la précision de la réponse.

Le contexte de cette communication se concentrera plus particulièrement sur les systèmes de type ChatBot, où le LLM est complété par des bases documentaires spécifiques à un domaine métier, permettant au LLM de s'appuyer sur un corpus dédié lors de son dialogue. Nous n'aborderons pas les autres types d'applications possibles des LLM ou de l'IA dite « Générative », comme par exemple l'ensemble des applications de génération ou de production de synthèse, code, images, ….

De plus nous nous limiterons aux applications professionnelles industrielles ou critiques, pour lesquelles la véracité, la précision et la pertinence des réponses proposées par l'IA sont primordiales, à l'opposé des applications grand public où d'autres problématiques complémentaires peuvent se poser, comme notamment des exigences éthiques particulières (RGPD, AI ACT), qui ne seront donc pas abordées ici.

Dans ce contexte précis, nous décrirons les principales difficultés d'évaluation et de validation de ces systèmes, et les recommandations et bonnes pratiques applicables, issues de l'état de l'art et du retour d'expérience du déploiement de ce type de systèmes dans des environnements industriels.

Cette communication abordera exclusivement les aspects relatifs à la sécurité fonctionnelle, ou « Safety », sans traiter de la sécurité logicielle ou cybersécurité, propres ou non à ces composants de type LLM.

Par ailleurs, dans ce domaine de l'IA extrêmement innovant et dynamique, il faut être conscient que l'état de l'art actuel est susceptible d'évoluer très rapidement, et que les conclusions ou argumentations de cette communication peuvent être remis en cause à court terme, suivant les nombreuses évolutions technologiques en cours.

## II. INTRODUCTION GENERALE

Afin de présenter la problématique de la confiance et de la robustesse de ces systèmes, il est nécessaire de présenter les architectures des solutions de type ChatBot, précédemment développées sans l'utilisation de LLM, et depuis peu, basées sur ces nouveaux composants linguistiques.

### *A. Rappel sur les technologies des ChatBots*

Le besoin d'apporter une aide ou des informations en communiquant en langage naturel avec des utilisateurs est à l'origine du développement de logiciels de ChatBot (ou agent conversationnel), qui ont évolué au gré des progrès en informatique et notamment des avancées en traitement du langage naturel.

Initialement basés sur de simples FAQ, les ChatBots se sont complexifiés pour devenir des arbres de décisions, ou des moteurs de règles, permettant d'identifier la réponse attendue dans une base de connaissance, puis depuis l'avènement des LLM, des environnements de dialogues complets et avancés exploitant des corpus documentaires étendus.

Si l'architecture des ChatBots est assez générique (cf Figure 1), les technologies utilisées pour leurs implémentations ont beaucoup évolué.

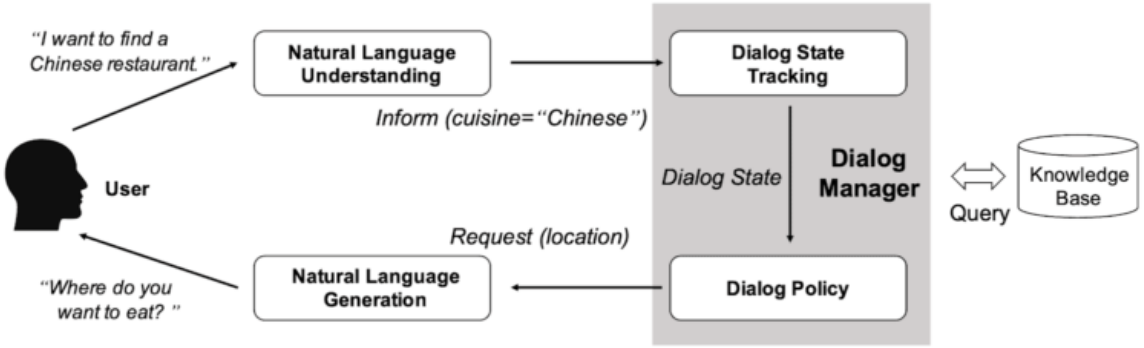


*Figure 1: Architecture générique d'un ChatBot selon [10]*

Les technologies NLP dites de NLU (Natural Language Understanding) sont nécessaires pour la compréhension des demandes de l'utilisateur, et celles de NLG (Natural Language Generation) pour la synthèse de réponses adaptées au contexte.

Un logiciel de ChatBot doit ensuite être capable de suivre le contexte ou l'état du dialogue avec l'utilisateur, afin de pouvoir répondre de façon pertinente (cf Figure 1). La conception du suivi du dialogue par le ChatBot va être très liée à chaque technologie, en mixant des approches à base de règles, de workflow sous la forme de graphes ou d'arborescence, ou par des approches purement linguistiques par l'exploitation du contexte (LLM).

Ainsi la description du dialogue des ChatBots a longtemps été formalisée sous forme de graphes ou workflow, par exemple au travers des solutions comme Botpress, RASA, DialogFlow (cf Figure 2), mais tend dorénavant à être remplacée par des approches linguistiques (LLM) plus ouvertes.

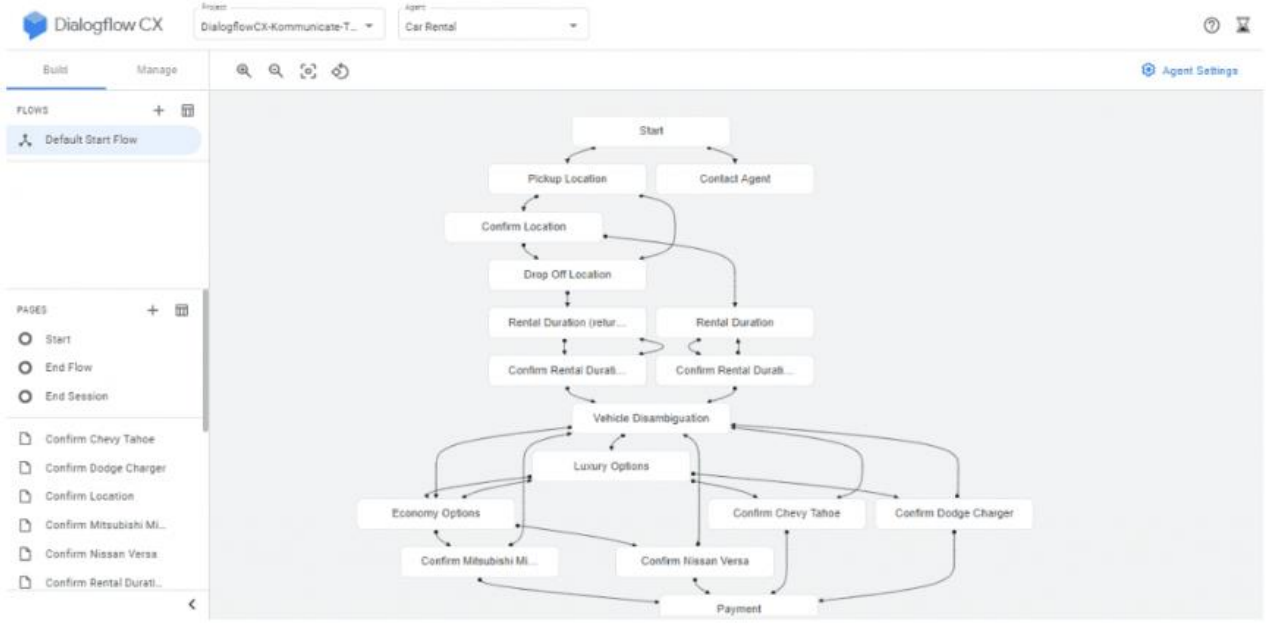


*Figure 2: Exemple de graphe de dialogue dans le ChatBot DialogFlow de Google décrivant les différents états du dialogue utilisateur*

Les avantages de ces solutions étaient assez nombreux :

- Une maîtrise complète du dialogue avec l'utilisateur et du contenu de la base de connaissance, rendant impossible la production de réponses fausses, mais pas forcément pertinentes,
- Un développement logiciel classique, assisté par de nombreux éditeurs visuels/ergonomiques, permettant la réalisation assez rapide de ce type de solution.

Néanmoins ces solutions avaient plusieurs inconvénients que nous avons tous directement constaté lors de nos propres utilisations de ChatBots :

- Un manque de souplesse dans le dialogue, et de compréhension de la véritable intention de l'utilisateur,
- Une base de connaissance limitée, du fait que celle-ci devait explicitement être mise à jour par les administrateurs du ChatBot, ce qui représentait une charge importante difficilement maintenable.

Face à ces solutions robustes et fiables mais limitées et figées, de nombreux chercheurs et industriels ont vu dans les progrès de traitement du langage naturel et en particulier avec les nouvelles capacités des LLM, la possibilité de concevoir des solutions de ChatBots beaucoup plus ouvertes et puissantes (voir [21] pour un historique des approches de ChatBot).

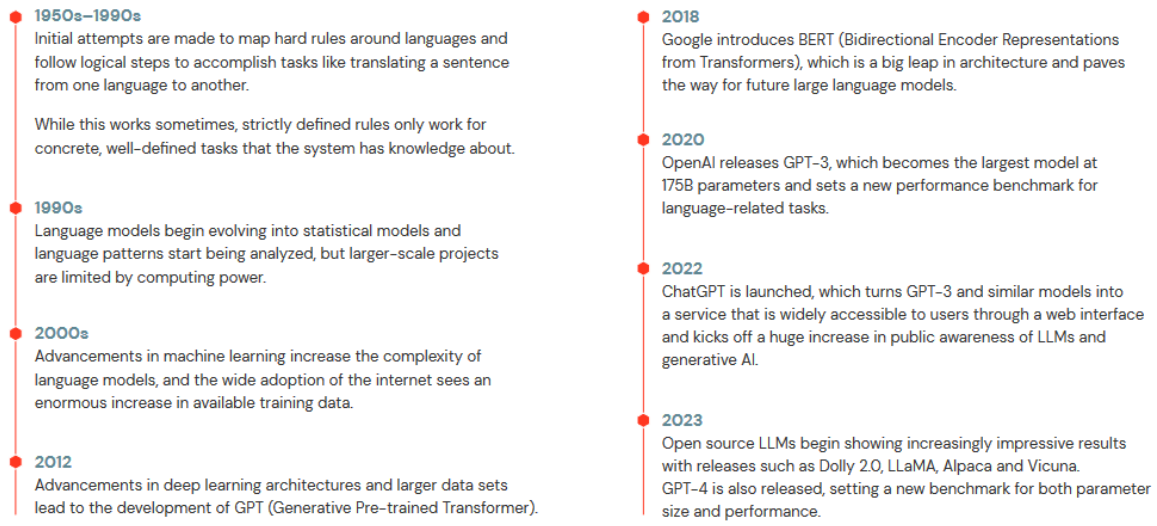


*Figure 3: Frise chronologique des technologies de TAL [19]*

## B. *Architecture de Chatbots à base de LLM et RAG*

Les LLM (voir [19] pour une vue détaillée de l'état de l'art LLM, mis à jour régulièrement) utilisés dans les Chatbots sont actuellement basés sur un fonctionnement commun, c'est à dire celui de générer des phrases mot à mot, en tenant compte de l'historique du dialogue, appelé « contexte » dans cette terminologie, et dont la taille est exprimée en « tokens », un « token » correspondant plus ou moins à la taille un mot.

Les LLM sont entrainés à partir de corpus documentaire publics dont le contenu exact peut-être partiellement indiqué par le fournisseur, et avec comme objectif de produire le mot le plus « probable » ou « plausible » dans le cadre de ce corpus, ce qui évidemment ne garantit pas la véracité de l'information générée. Cette phase d'apprentissage initial du LLM (cf pré-training Figure 4) est ensuite complétée par une spécialisation du LLM pour du dialogue, puis avec des rétrofits à base d'évaluations humaines.

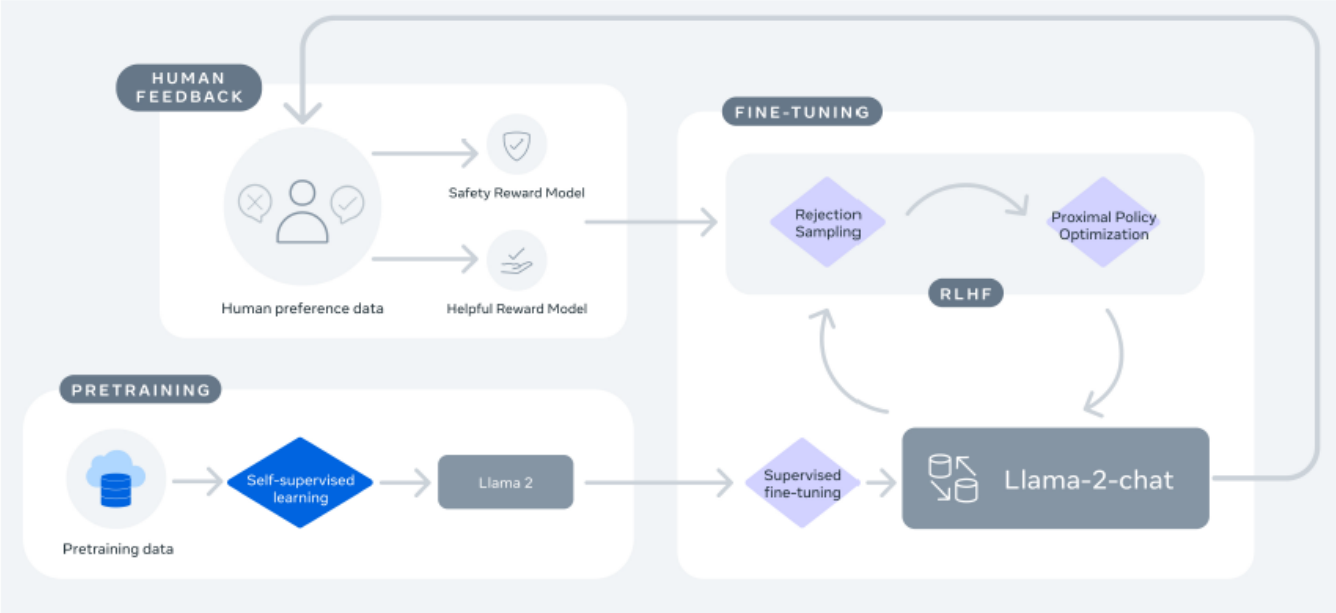


*Figure 4: Etapes de conception d'un LLM pour un ChatBot (cf LLama2 [6])*

Les LLM ont ainsi été entrainés sur des corpus documentaires certes « encyclopédiques » mais figés à un moment donné, et ne peuvent donc prendre en compte des mises à jour de l'information, ni bien entendu des corpus documentaires privés ou spécifiques. De plus, l'information utilisée a été entièrement « diluée » dans le LLM, sans conserver les références vers les sources initiales.

Pour pallier cette limitation dans le cadre d'une application de ChatBot, étant entendu qu'il n'est pas réaliste de réaliser un réapprentissage du LLM pour chaque mise à jour de la base documentaire, les chercheurs ont déployé un mécanisme dit de RAG (Retreival Augmented Generation)[4], [13], dont le principe général est de rechercher dans la base documentaire les extraits les plus pertinents ou similaires par rapport à la question de l'utilisateur, puis dans un deuxième temps ces extraits seront fournis au LLM, afin que celui-ci synthétise une réponse en tenant compte explicitement de ces extraits (cf Figure 5).

Afin de procéder à cette extraction de façon efficace, les documents originaux sont découpés en extraits (« chunks ») puis vectorisés sous forme de « tokens » qui sont ensuite stockés dans une base vectorielle.

Le schéma suivant (Figure 5) décrit ce mécanisme de RAG : la question de l'utilisateur, après reformulation, est comparée aux extraits vectorisés du corpus (« chunks »), puis le LLM effectue une synthèse de la réponse, en se basant sur cette sélection.

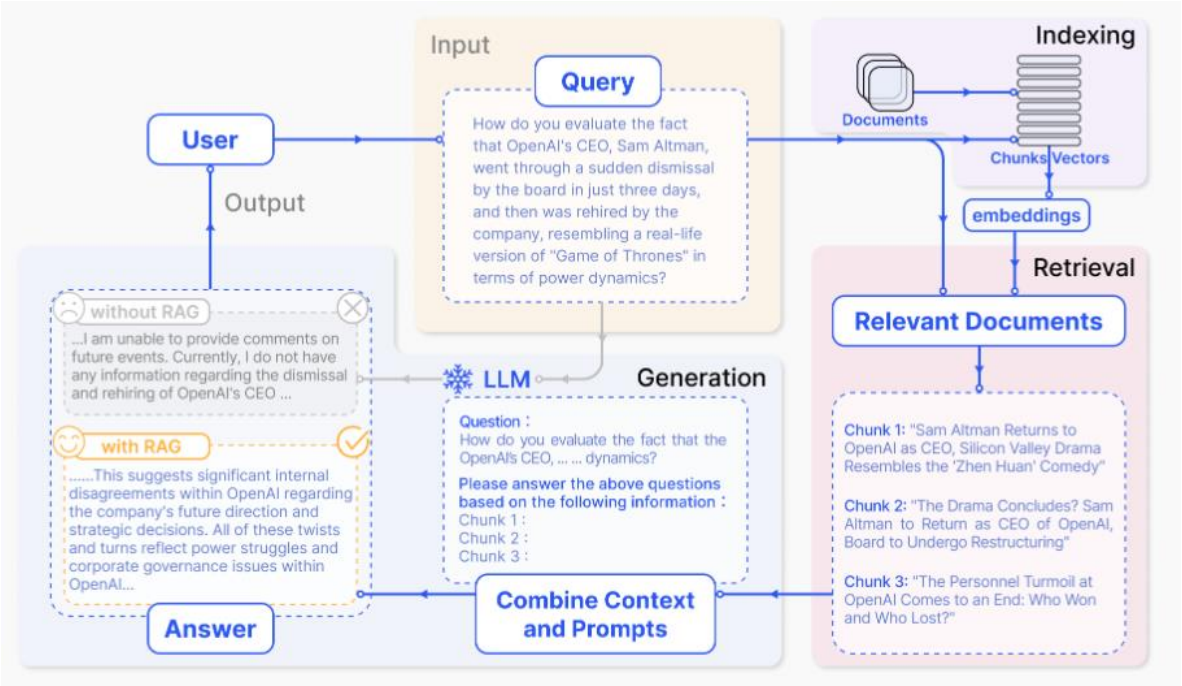


*Figure 5: Fonctionnement générique d'un ChatBot basé sur un LLM couplé avec une fonction de RAG [10]*

Il existe bien sûr une multitude d'approches techniques pour implémenter et optimiser cette fonctionnalité d'extraction des extraits les plus pertinents [10], y compris de nombreuses approches non-déterministes et/ou utilisant partiellement des LLM, et de nombreux travaux de recherche sont encore en cours pour améliorer les performances, mais le principe général de la fonctionnalité reste le même.

Au final les avantages de ces architectures de Chabots à base de LLM/RAG sont nombreux :

- Compréhension plus souple et robuste du dialogue, et de la demande de l'utilisateur,
- Synthèse d'une réponse plus fine et adaptée à la question,
- Corpus documentaire facile à mettre à jour, par ingestion de nouveaux documents,
- Multi-langue et traduction implicite des documents.

Il faut cependant prendre en compte les inconvénients suivants :

- Nécessité de ressources matérielles très significatives (GPU pour Graphic Processing Unit), suivant la taille du LLM et du corpus,
- Hallucinations possibles du LLM lors de la compréhension de la question, ou bien lors de la synthèse de la réponse,
- Un comportement non-déterministe du système, y compris dans des contextes strictement identiques.

On notera cependant que cette approche de RAG permet d'obtenir des réponses précises basées sur des extraits documentaires, **mais ne permet pas d'obtenir des synthèses de document, ou des réponses transverses à l'ensemble de la base documentaire**. Ces applications demandent des développements et des stratégies d'implémentation différentes.

## III. PROBLEMATIQUE GENERALE

Les solutions de ChatBot conçues sur un couple LLM/RAG sont ainsi basés sur 4 éléments principaux (cf Figure 6):

- Le corpus documentaire, dont on verra que la spécification, la qualité et la structuration sont des éléments essentiels,
- Le LLM, qui pour la plupart des solutions sera vu comme un COTS intégré et exploité dans ce contexte particulier, et dans certaines situations sera adapté localement, au travers d'un processus dit de « Fine-Tuning »,
- La fonctionnalité de RAG, essentielle à la performance du LLM, dont l'objectif est d'identifier les extraits les plus pertinents du corpus documentaire,
- L'interface Homme-Machine qui va présenter les résultats à l'utilisateur et gérer son dialogue.

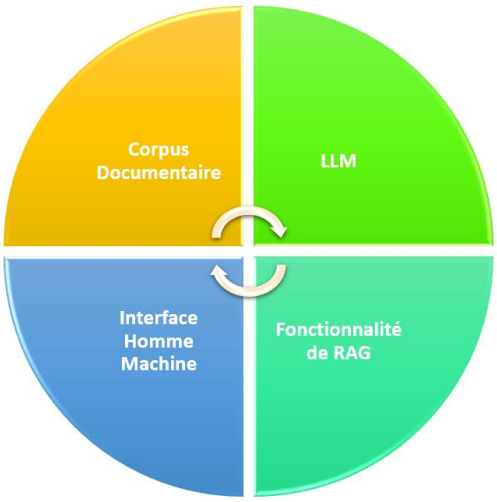


*Figure 6: Eléments essentiels d'un ChatBot basé sur un LLM/RAG*

Afin d'analyser les problématiques de validation et de robustesse de ces ChatBots, il est nécessaire de lister les principales limitations techniques des différents composants, et les difficultés méthodologiques de leur intégration ou évaluation.

### *A. Limitations techniques et difficultés méthodologiques relatives à ces solutions*

#### *1) Les limitations d'exploitation des extraits par les LLM*

Comme indiqué précédemment, les extraits pertinents (appelés « chunks ») sont intégrés dans le contexte du LLM, afin que celui-ci élabore une réponse (cf Figure 5 – phase « Generation »).

Dans une approche naïve, il semblerait pertinent d'intégrer le maximum de documents plausibles dans le contexte du LLM, en espérant que celui-ci soit capable d'en extraire l'information utile pour répondre à la question de l'utilisateur. Cependant, mis à part le problème de la limitation technique de la taille du contexte des LLM (de l'ordre de 30/40k de mots) qui impacterait les performances de temps de réponse, les expérimentations montrent que l'augmentation de la taille du contexte n'améliore pas la qualité de la réponse, et au contraire finit par la dégrader [14]. Par ailleurs, l'augmentation des extraits augmenterait aussi la difficulté de produire une synthèse adaptée et cohérente.

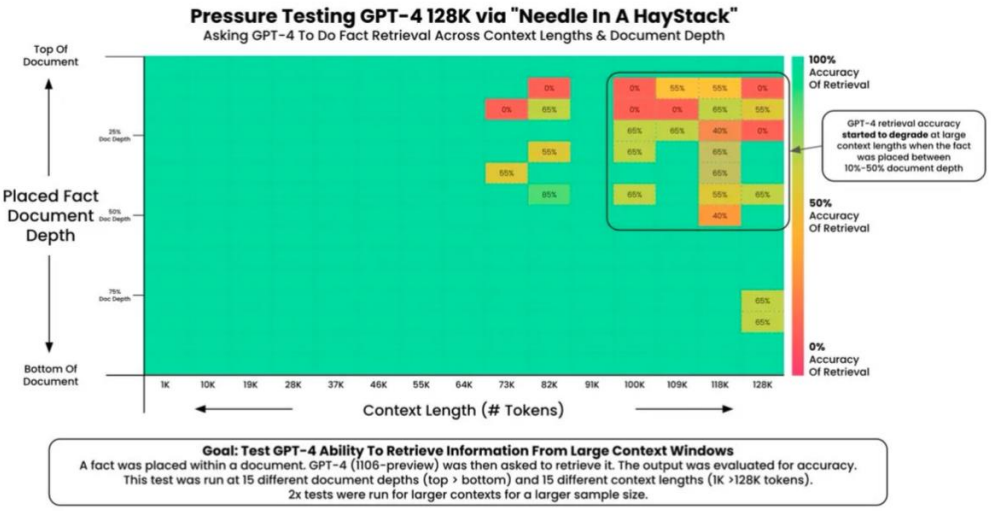


*Figure 7: Evaluation des performances d'un LLM en fonction de la taille du contexte [14]*

**La conséquence de cette limitation des LLM est que la fonctionnalité de RAG doit extraire un nombre très réduit d'extraits pertinents (de l'ordre de 4 à 6 en pratique), ce qui impose un filtrage extrêmement performant, en liaison directe avec la taille du corpus.**

Ainsi, on comprendra très facilement que la performance requise par la fonctionnalité de RAG ne sera pas la même si le corpus documentaire est constitué de centaines ou bien de dizaines de milliers de documents, et que de nombreux ajustements pourront être nécessaires pour optimiser cette fonction du RAG (taille des extraits, fct de recherche.), y compris pour chaque cas d'usage.

Cette problématique est à l'origine de l'effet « Waow » lors des démonstrations de ces solutions lorsque le corpus est extrêmement réduit, mais malheureusement dont les performances se dégradent très nettement au fur et à mesure de la constitution du corpus réel et de la mise en production.

*2) Les limitations de prise en compte du contenu des documents par les LLM/RAG*

Dans l'état de l'art actuel, la plupart des LLM ne sont actuellement capables que de manipuler du texte, en entrée comme en sortie.

L'exploitation de schémas ou d'images dans les documents du corpus documentaire va devoir passer par des pré-traitements de reconnaissance d'image, capables de produire des descriptions textuelles qui pourront ensuite être intégrées dans les extraits documentaires exploitables. Cependant la performance de ce type de traitement de reconnaissance est toujours extrêmement variable.

**La conséquence de cette limitation est que les documents ayant de fortes parties graphiques/schémas ne pourront pas être correctement intégrés dans le corpus documentaire, et donc interprétés par le LLM.**

On notera le cas intéressant de l'intégration de vidéos, dont l'exploitation des sous-titres issus de la bande-son peut-être particulièrement utile et fiable.

*3) La difficulté de l'évaluation objective de la qualité des réponses fournies*

Une des grandes difficultés de l'évaluation de ces solutions, est l'évaluation objective de la qualité de la réponse fournie par le LLM. S'il est possible de définir un ensemble de critères qualitatifs pour estimer la qualité de la réponse (Véracité, Pertinence, Exhaustivité et Concision de la synthèse, …), aucun de ces critères ne peut être évalué quantitativement et objectivement de façon automatique.

De fait, ces évaluations seront donc exclusivement humaines et de plus, dépendantes des cas d'usage et des utilisateurs concernés (expertise et contexte d'emploi).

Certaines approches proposent l'utilisation d'autres LLM afin de vérifier le comportement du Chabot, mais ce type d'approche ne peut être proposée que comme une solution d'évaluation préliminaire.

**La conséquence de cette limitation est que l'évaluation des réponses des ChatBots ne peut être effectuée que par des humains, ce qui implicitement limite la quantité et la couverture des tests réalisables.**

*4) Le non-déterminisme des LLM*

Le principe des LLM est de produire les mots successivement, en tenant compte de la probabilité d'apparition, suite à l'apprentissage préalable. Ces logiciels sont donc intrinsèquement non-déterministes, même s'il est possible en théorie de limiter la « température » du ChatBot afin de sélectionner systématiquement le choix le plus probable, en se rapprochant ainsi d'un comportement plus figé.

Cependant en pratique il est même utile de permettre une certaine souplesse dans la réponse, ceci afin de permettre au ChatBot de proposer plusieurs réponses à l'utilisateur, et aussi de garder un certain naturel et spontanéité dans le dialogue.

**La conséquence de cette limitation est que le résultat produit par le ChatBot sera nécessairement non-reproductible, ce qui en pratique ne permet pas de concevoir des tests d'évaluation simples.**

On notera par ailleurs, que l'exécution d'un même LLM sur différentes implémentations Hardware (GPU), peut aussi aboutir à des résultats légèrement différents, en liaison avec la cascade de calculs mathématiques effectués [22].

*5) Les hallucinations de LLM*

Dans ce contexte les LLM, le terme « hallucination » correspond de façon générale à toute sorte d'erreur que peut produire celui-ci dans sa réponse.

Compte tenu de l'intégration du LLM au travers de la fonction de RAG, les hallucinations pourront principalement se produire lors de la synthèse finale de la réponse à partir des extraits utilisés. Si quelques techniques de « Prompting » peuvent limiter ces « hallucinations », il n'existe pas, dans l'état de l'art actuel, de méthode définitive pour les supprimer.

**La conséquence de cette limitation est que le résultat produit par le ChatBot sera toujours soumis à de potentielles hallucinations du LLM.**

*6) La taille des LLM et le corpus d'apprentissage initial*

Les LLM sont aujourd'hui de taille très conséquente (~170 milliards de paramètres pour GPT4), ce qui nécessite implicitement un entrainement sur des contenus documentaires gigantesques dont les utilisateurs et/ou intégrateurs finaux ne peuvent que rarement connaitre, et encore moins investiguer précisément. De plus, le LLM, une fois construit, ne conserve pas les références d'origine aux documents, et ne peut les indiquer.

Par ailleurs, quelque soit les phases de validation effectuées par le concepteur du LLM à proprement parlé, seule une infime partie du LLM a été exploitée ou évaluée avant sa diffusion.

**La conséquence de cette limitation est que l'intégration d'un LLM dans un système consiste à importer un composant de type boite noire (ou grise au mieux) dans son environnement, similaire à l'intégration d'un COTS complexe.**

*7) Quelques points positifs liés au contexte d'emploi particulier*

Il existe cependant quelques éléments positifs pour permettre d'augmenter la confiance de ces systèmes !

Le premier est le fait que ces systèmes sont (actuellement) à destination d'utilisateurs humains, qui pourront donc toujours reconsidérer ou rejeter les propositions des ChatBots. Ce point est néanmoins à prendre avec circonspection, puisque cette « barrière de protection » sera très dépendante de l'expertise et de la vigilance des utilisateurs, et qu'il est paradoxal de demander aux utilisateurs de se méfier des réponses d'un système censé leur apprendre des informations ou leur fournir des recommandations.

Le deuxième est le fait que l'environnement d'utilisation de ces systèmes est très facilement maitrisable, puisque les données d'entrée sont limitées au corpus documentaire et aux échanges avec les utilisateurs, sans l'intégration de données environnementales externes, potentiellement beaucoup plus difficilement maitrisables.

**L'avantage de ce contexte d'entrée/sortie limité est qu'il sera toujours facile de tracer et analyser à postériori le comportement de ces ChatBots.**

### *B. Les événements indésirables des ChatBots*

L'évènement indésirable évident de ce type de solution est bien évidemment que le système produise une réponse inappropriée vers l'utilisateur, que l'on peut plus précisément lister selon les catégories suivantes :

- Réponse fausse, contenant une contre-vérité, une erreur factuelle d'interprétation, de calcul, de chronologie,
- Réponse incomplète, imprécise ou mal-structurée, ne prenant pas en compte une partie significative de l'information, ou n'exprimant pas les nuances nécessaires, ou ne présentant pas l'information dans un ordre approprié,
- Réponse non-pertinente, c'est à dire, ne répondant pas au sujet abordé même si la réponse peut être correcte,
- Réponse trop détaillée ou trop courte, par rapport à l'attendu de précision/concision de l'utilisateur,
- Absence de réponse, alors que le système dispose de l'information.

La criticité de cet événement indésirable va dépendre de la nature de l'information, ou plus précisément des conséquences que l'utilisateur pourrait en déduire, et donc indirectement de la criticité des documents concernés.

Par ailleurs, dans l'architecture d'un tel système, les causes de défaillance de ces solutions sont les suivantes :

- Absence de l'information dans le corpus documentaire ou via l'apprentissage initial du LLM,
- Information erronée (fausse, incomplète, imprécise) dans le corpus documentaire,
- Mauvaise compréhension de la question par le LLM dans le contexte du dialogue,
- Mauvaise sélection des extraits documentaires pertinents par le RAG,
- Mauvaise synthèse de la réponse par le LLM, produisant typiquement une « hallucination ».

Le chapitre suivant propose un certain nombre de recommandations de nature technique ou méthodologique, permettant de réduire le risque relatif à ces systèmes, et ainsi augmenter la confiance dans les réponses apportées.

## IV. Recommandations pour la maitrise de ces solutions

Afin permettre le déploiement de ces systèmes dans des environnements professionnels et éventuellement critiques, il est nécessaire de prendre en compte les nombreuses limitations précédentes, et les difficultés méthodologiques de la validation de ces systèmes.

Les recommandations suivantes ont été élaborées en se basant sur l'état de l'art technologique actuel et sur le retour d'expérience issu du déploiement de ce type de solutions au sein de l'Administration. Ces recommandations sont évidemment incomplètes ou sommaires, mais donneront un cadre pour maitriser le déploiement de ce type de systèmes.

Ces recommandations sont principalement d'ordre méthodologique, car il n'existe pas, à notre connaissance, d'approche systématique ou technique permettant de définir puis évaluer quantitativement la confiance ou la robustesse de ce type de systèmes.

### *1) Clarifier le besoin des utilisateurs et le domaine d'application concerné*

Si la phase de clarification du besoin avec les utilisateurs finaux est une étape évidente de tout projet (d'IA ou non), le point adressé ici est plus particulièrement celui du domaine applicatif ou opérationnel et de la couverture par le corpus documentaire associé. A l'instar de l'ODD (Operational Design Domain) utilisé dans le cadre des applications d'IA et servant à définir le domaine d'utilisation opérationnelle de l'IA suite à son apprentissage, il est important de déterminer avec précision le domaine opérationnel cible du ChatBot, en particulier par les étapes suivantes :

- Clarifier le niveau d'expertise des utilisateurs et lister des exemples de questions/réponses attendues,
- Définir les scénarios de dialogue et la capacité des utilisateurs à préciser leurs attendus (Prompting),
- Recenser le corpus documentaire initial et évaluer la faisabilité technique de l'ingestion des documents,
    - Langues, lexique, format particulier, prise en compte des images,
    - Taille du corpus et performances attendues du filtrage du RAG.
- Reformuler/rédiger les documents contenant des informations critiques, dont l'information pourraient être ambiguë, ou trop dépendante de schémas/images non exploités par le LLM,
- Préciser les processus de mise à jour du corpus documentaire.

Mais plus particulièrement dans une optique de « Safety », il est important de :

- Déterminer les sujets/questions critiques du domaine, ce qui permettra d'implémenter des traitements ou des IHM spécifiques, lors du dialogue et lors de l'ajout de documents ultérieurs par les Administrateurs du système,

- Identifier des sujets/questions hors-domaine, dont on souhaite explicitement qu'ils ne soient pas traités par le système, par exemple parce que l'on considère que les informations disponibles ne seraient pas assez fiables ou souhaitables.

Il faudra ensuite implémenter des verrous et protections possibles au niveau des différentes étapes des traitements du système (Prompting LLM, RAG, IHM) en tenant compte de ces informations.

*2) Structurer le corpus documentaire avec des meta-datas*

Il serait particulièrement contre-productif et inefficace d'imaginer un ChatBot où le corpus documentaire serait constitué d'un gigantesque « datalake » non structuré. En effet, cela reporterait sur le couple LLM/RAG de trop fortes contraintes de précisions pour la compréhension puis le filtrage documentaire. Il est donc important de disposer d'un corpus documentaire structuré avec des méta-datas, ce qui permettra à l'utilisateur de préciser très rapidement et sans ambiguïté le contexte de son dialogue (sujet, temporalité, localisation, ...) sans reporter cette sélection basique sur le LLM.

Idéalement, les méta-data des documents devront :

- Prendre en compte la confiance de la source de chaque document, ceci afin de favoriser les documents les plus fiables, et éventuellement d'apporter certains avertissements aux utilisateurs,
- Identifier la criticité des documents selon le contenu des informations, ceci afin de pouvoir les mettre en évidence,
- Identifier la temporalité/validité des documents, afin de permettre un filtrage par date/période des documents, information rarement présente au sein même des documents, et difficilement exploitable par les LLM,
- Regrouper les documents par sous-corpus pour permettre de filtrer immédiatement les besoins par l'utilisateur.

*3) Mettre en place une procédure qualité d'intégration de nouveaux documents*

La qualité et la cohérence du corpus documentaire étant essentielle, il est nécessaire de définir une procédure (ou checklist) d'intégration des nouveaux documents, et idéalement d'utiliser cette procédure dés la constitution du corpus initial, ceci afin, à la fois de consolider cette procédure, mais aussi de garantir que le corpus initial aura bien été constitué selon les mêmes règles que les futures mises à jour. Il y aurait en effet un risque important de dégradation des performances par l'intégration de documents de mauvaise qualité.

Une des possibilités serait de demander à l'utilisateur, à minima pour les documents critiques, de fournir des exemples de questions associées à ce nouveau document. Ceci permettrait en effet de vérifier que :

- Le RAG fonctionne correctement sur ce document, et n'est pas perturbé par d'autres documents existants du corpus,
- Réciproquement, que l'intégration de ce nouveau document ne perturbe le fonctionnement du RAG sur d'autres documents critiques du corpus,
- Le document fourni correspond bien au domaine opérationnel attendu (ODD).

*4) Sélectionner et intégrer le LLM au même titre qu'un COTS externe*

Le LLM qui va être intégré dans le ChatBot est typiquement une boite noire ou « Grise », compte tenu de sa taille et de la masse de documents utilisée pour son apprentissage.
Dans ce cadre, la seule approche d'ingénierie imaginable pour l'intégrateur du LLM, est de considérer ce composant comme un COTS complexe (un compilateur par exemple), et donc d'appliquer au mieux les recommandations et bonnes pratiques définies dans de nombreux standards de développement logiciel (plan d'acquisition du COTS, exigences, gestion des versions, tests de non-régression, modes de défaillance, documentation, …), malgré d'évidentes limitations (identification de code mort).

Plus spécifiquement dans ce contexte des ChatBots, il sera particulièrement nécessaire :

- De privilégier les LLM conçus avec transparence (cf The Fondation Model Transparency Index [23]),
- D'identifier en particulier le corpus d'apprentissage du LLM [15],
- D'évaluer le LLM sur des tests adaptés au cas d'usage pour la connaissance implicite par rapport au domaine concerné (HELM/MMLU [5] et [7]),
- De prioriser un LLM favorisant la véracité et la précision des réponses, et en particulier, capable de dire qu'il ne peut pas répondre à une question.

D'un point de vue ingénierie, il faudra aussi tenir compte du remplacement rapide de ce LLM par des versions ultérieures ou d'autres LLM plus performants, compte tenu de l'innovation très rapide de ce domaine.

Par ailleurs, dans un certain nombre de cas d'usage, (lexique métier complexe), il pourra être nécessaire de réaliser un « Fine-Tuning » du LLM retenu sur le corpus dédié, ce qui complexifiera encore davantage la gestion du LLM en tant que « COTS adapté localement ».

*5) Permettre à l'utilisateur de vérifier ou appréhender la confiance du système dans sa réponse*

Au final, et quelles que soit les approches prises précédemment, il sera important de permettre aux utilisateurs de vérifier la réponse du système.
Plusieurs approches peuvent être envisagées, telles que :

- Présenter de façon ergonomique les extraits utilisés par le système pour construire sa réponse, ce qui permettra à l'utilisateur de vérifier la pertinence des extraits, et de compléter sa prise d'information (cf Figure 8),
- Donner accès aux documents originaux (sans régénération par le système), en particulier dans les cas où la question a été considérée comme critique, ou si les documents sont considérés comme tels.

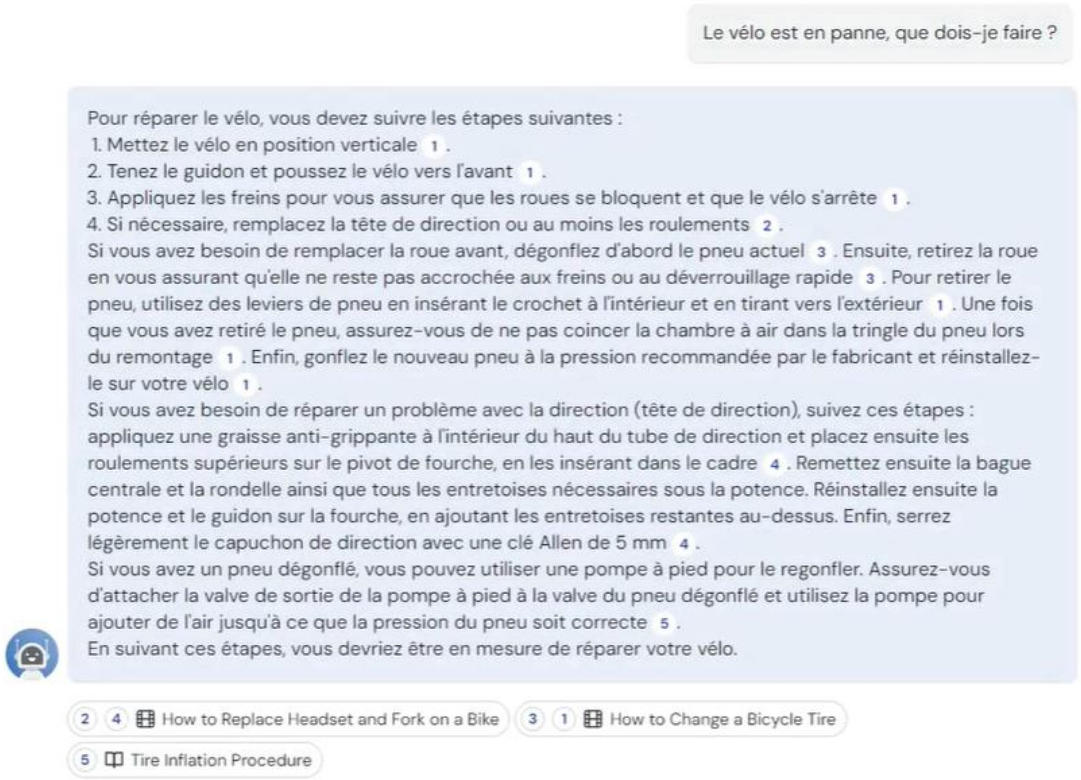


*Figure 8: Exemple de réponse de ChatBot présentant les références documentaires, dans l'assistant virtuel de la société Stellia*

- Si possible selon les capacités du LLM, utiliser une formulation conditionnelle dans la réponse, lorsque par exemple les sources ne sont pas de grande confiance, ou lorsque des ambiguïtés ou incohérences sont détectées entre les extraits, ceci afin d'encourager l'utilisateur à aller consulter les documents originaux,
- Indiquer clairement à l'utilisateur quand la réponse a été produite par le LLM seul, c'est à dire sans utiliser de documents issus du corpus, si le cas d'usage permet cette option,
- Enfin permettre à l'utilisateur de rejouer la requête pour obtenir un nouveau résultat du système, ce qui lui permettra de consolider sa prise d'information, et de confirmer/infirmer la première réponse du système.

*6) Monitorer les performances de la fonctionnalité de RAG*

Compte tenu de l'importance de la fonctionnalité du RAG sur la performance du système et la qualité des résultats, il est important d'évaluer spécifiquement cette fonction par des métriques [25] dédiées lors de tests exhaustifs par rapport au corpus initial, mais aussi de monitorer cette performance en cours d'exploitation, au fur et à mesure de l'évolution du corpus.

### *7) Définir un protocole d'évaluation de la solution*

Le processus d'évaluation et de validation externe de ce type de système avant leur mise en exploitation est complexe, et peu automatisable. Au contraire des approches classiques de validation de logiciels (ça marche ou ça ne marche pas), il faut ici déterminer des seuils d'acceptabilité du système sur différents scénarios et usages sous la forme d'un protocole d'expérimentation, et construire des cas de tests représentatifs des documents du corpus et des questions des utilisateurs.
Par ailleurs, assez rapidement sur des corpus métier large et complexe, il est nécessaire de faire appel aux utilisateurs finaux pour évaluer la qualité et la pertinence métier de la réponse.

Il est cependant possible de séparer deux critères de performance évaluables séparément :

- La pertinence des extraits sélectionnés (performance du RAG) :
    1. Les extraits présentés sont les plus pertinents du corpus par rapport à la question posée,
    2. Les extraits présentés sont pertinents, mais pas nécessairement les meilleurs,
    3. Les extraits présentés sont peu pertinents voire incorrects,
- Puis la qualité de la synthèse de la réponse par le LLM dans les cas précédents 1&2 :
    1. La synthèse du système est exacte, complète et concise,
    2. La synthèse du système est exacte, mais incomplète, imprécise ou mal structurée,
    3. La synthèse du système est incorrecte, non pertinente ou fausse.

Il sera ainsi nécessaire de définir des seuils d'acceptabilité pour ces différentes étapes du traitement.

### *8) Faire une formation minimale de la technologie des LLM aux futurs utilisateurs*

Comme toute nouvelle technologie et en particulier en IA, et même si celle-ci se développe très rapidement auprès du grand public, il est souhaitable de réaliser des formations minimales de cette technologie LLM auprès des futurs utilisateurs, notamment afin de leur permettre de démystifier ses capacités, mieux appréhender leurs limites (RAG, hallucinations), et pouvoir utiliser tout leur potentiel dans le processus d'interrogation (Prompting, contexte).
Un autre point important sera la prise en compte par les administrateurs du système, du maintien de la qualité du corpus documentaire.

## V. Conclusion

Si cet article a été écrit sans l'utilisation de LLM (!), il semble intéressant de tester leur capacité de synthèse, et leur demander de produire la conclusion de celui-ci :

*« Les systèmes de ChatBots basés sur des LLM (Large Language Model) et des mécanismes de RAG (Retreival Augmented Generation) présentent de nombreux avantages, notamment en termes de compréhension du dialogue, de synthèse de réponses adaptées et d'évolution du corpus documentaire. Toutefois, il est important de prendre en compte les inconvénients potentiels, tels que les hallucinations possibles du LLM, le comportement non-déterministe du système et la nécessité de ressources matérielles importantes. Les évaluations et les recommandations pour garantir la confiance et la robustesse de ces systèmes sont essentielles pour leur utilisation dans des environnements critiques et industriels ».*

## REMERCIEMENTS

Je tiens à remercier l'ensemble des partenaires et acteurs, dont la confidentialité est requise, qui nous ont permis, de par le déploiement et l'expérimentation de ces systèmes dans des environnements complexes et significatifs, de pouvoir appréhender et analyser la difficulté de prise en main de ces technologies, et d'en concevoir des stratégies de déploiement et une ébauche de méthodologie.

# ABREVIATIONS ET ACRONYMES

| Terme | Signification |
|---|---|
| AI | Artificial Intelligence |
| AI ACT | Règlement Européen sur l'Intelligence Artificielle |
| API | Application Programming Interface |
| BERT | Bidirectional Encoder Representations from Transformers |
| ChatBot | Agent conversationnel |
| Chunk | Sous-partie d'un texte pour son exploitation par le RAG. Le découpage peut se faire au niveau de la phrase, du paragraphe ou être de taille fixe, avec un chevauchement entre chunks succesifs |
| Contexte | Buffeur de discussion du LLM |
| CU | Cas d'Utilisation |
| Fine-tuning | Adaptation d'un modèle LLM pré-entraîné par un entraînement sur un corpus spécifique ou un retour d'expérience utilisateur |
| GPU | Graphic Processing Unit, composant HW nécessaire à l'entrainement et l'exécution des LLM |
| GPT | Generative Pre-trained Transformer |
| HELM | Holistic Evaluation of Language Models. Approche d'évaluation des LLM |
| IHM | Interface Homme Machine |
| IA | Intelligence Artificielle |
| LLM | Large Language Models |
| ML | Machine Learning – Apprentissage Machine |
| MMLU | Multi-task Language Understanding |
| NLG | Génération de la langue, abréviation de l'anglais Natural Language Generation |
| NLP | Natural Language Processing (TAL Traitement Automatique des Langues en Français) |
| NLU | Compréhension de la langue, abréviation de l'anglais Natural Language Understanding |
| ODD | Operational Design Domain |
| PoC | Proof of Concept |
| Prompt | Instruction sous forme de langage naturel à l'attention des LLM |
| RAG | Retrieval Augmented Generation |
| SLM | Small Language Models |
| TAL | Traitement Automatique des Langues, équivalent anglais de Natural Language Processing (NLP) |
| Token | Unité de base utilisée pour représenter et traiter le langage dans les IA génératives, de granularité variable : lettre, syllabe, d'un mot ou d'une partie de mot, |
| Zero-shot | Requête d'un LLM avec un contexte du dialogue vide, ou « à froid » |

## References


[1] BERT : Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[2] Language models are unsupervised multitask learners. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019).

[3] End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. Ham, D., Lee, J.-G., Jang, Y., and Kim, K.-E. (2020) In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 583–592, Online. Association for Computational Linguistics.

[4] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (34th Conference on Neural Information Processing Systems (NeurIPS 2020). https://proceedings.neurips.cc/paper/2020

[5] HELM: Holistic Evaluation of Language Models - enter for Research on Foundation Models (CRFM) - Stanford Institute for Human-Centered Artificial Intelligence (HAI) Stanford University (2023). https://arxiv.org/pdf/2211.09110.pdf

[6] Llama 2: Open Foundation and Fine-Tuned Chat Models (2023). https://ai.meta.com/resources/models-and-libraries/llama/

[7] MMLU: Measuring Massive Multitask Language Understanding (2021). https://arxiv.org/pdf/2009.03300.pdf

[8] Resources and best practices for responsible development for products powered by large language models – Meta AI – (2023)

[9] BLOOM Open-Access Multilingual Language Model (2023). https://arxiv.org/pdf/2211.05100.pdf

[10] How to Build a ChatBot: Components & Architecture in 2024. (2024). Cem Dilmegan. https://research.aimultiple.com/chatbot-architecture/

[11] Comparaison des solutions de NLU sur un corpus français pour un chatbot de support COVID-19 (2022). https://ci.mines-stetienne.fr/pfia2022/conferences/ic/Articles/S-10-Article-2.pdf

[12] Base de données d'incidents relatifs à l'Intelligence Artificielle https://incidentdatabase.ai/

[13] Retrieval-Augmented Generation for Large Language Models: A Survey (2024). https://arxiv.org/pdf/2312.10997.pdf

[14] The Needle in a Haystack Test, Evaluating the performance of RAG systems - Aparna Dhinakaran -Towards Data Science 2024 (2024). https://towardsdatascience.com/the-needle-in-a-haystack-test-a94974c1ad38

[15] The Claire French Dialogue Dataset (2023) - https://arxiv.org/abs/2311.16840

[16] A Comprehensive Overview of Large Language Models (2024). https://arxiv.org/abs/2307.06435

[17] The LLM Testing Guide: Comprehensive Strategies for Testing and Behavior Analysis (2023) – Mark Chen - Kolena

[18] LLM AI Security & Governance Checklist – Sandy Dunn – OWASP (2023)

[19] A Compact Guide to Large Language Models (2023)– Copyright Databricks

[20] AI ACT 2024 : https://artificialintelligenceact.eu/fr/ai-act-explorer/

[21] A Complete Survey on LLM-based AI Chatbots (2024). https://arxiv.org/pdf/2406.16937

[22] Changing the GPU is changing the behaviour of your LLM (2024) - https://medium.com/@anis.zakari/changing-the-gpu-is-changing-the-behaviour-of-your-llm-0e6dd8dfaaae

[23] The Foundation Model Transparency Index (2023) https://arxiv.org/pdf/2310.12941v1 https://crfm.stanford.edu/fmti/May-2024/index.html

[24] Evaluating the Efficacy of Open-Source LLMs in Enterprise-Specific RAG Systems: A Comparative Study of Performance and Scalability (2024) : https://arxiv.org/pdf/2406.11424

[25] Top Evaluation Metrics for RAG Failures. https://towardsdatascience.com/top-evaluation-metrics-for-rag-failures-acb27d2a5485