Langchain et Mistral, votre ChatGPT souverain et adapté à vos données

Pierre CARTIER
Ingénieur IA et transition numérique
Resp. transition numérique à Le Mans Université & co-founder de Plaiades
72000 Le Mans

Gaétan GOTTIS
Ingénieur IA et milieu industriel
Dirigeant de Leveraize, co-founder de Plaiades & enseignant Le Mans Université
72000 Le Mans

Résumé

La nécessité d'accélérer la transformation digitale, combinée à la prolifération des solutions d'IA générative disponibles pour les utilisateurs, oblige les DSI à aborder ces enjeux de front, notamment en ce qui concerne les défis posés par le shadow IT (utilisation de technologies, d'applications ou de solutions informatiques au sein d'une entreprise sans l'approbation ou le contrôle de la direction des systèmes d'information).

L'utilisation de ces solutions impliquant le partage d'un volume important d'informations internes et/ou souvent confidentielles soulève des préoccupations en matière de souveraineté et de contrôle des données.

De nombreuses solutions disponibles sur le marché permettent aux DSI de les intégrer dans leur catalogue de services. Cependant, la majorité de ces solutions, telles que ChatGPT et ses concurrents, sont hébergées sur des plateformes cloud externes ce qui ne permet pas de répondre entièrement à la problématique.

Une alternative est le passage par des solutions open source, souvent moins performantes que les solutions propriétaires en matière d'IA générative, elles nécessitent une certaine expertise pour rivaliser avec ces dernières. L'expertise requise se retrouve à plusieurs niveaux : la gestion des modèles et leur orchestration puis mise à disposition, ou encore la gouvernance des données et leur ingestion dans des bases de connaissances à disposition des modèles dans la réalisation de cas d'usage métiers spécifiques.

Il est alors possible d'améliorer les services de support usagers par un traitement plus rapide. On peut mettre en place des chatbots spécialisés sur les données tout en garantissant une intégrité et une sécurité des données internes.

Mots-clefs

Intelligence Artificielle, IA, Large Language Model, LLM, ChatGPT, souveraineté numérique, chatbot, digitalisation, mistral, gemini, vectorstore, RAG, retrieval

1 Introduction

1.1 Contexte et enjeux des LLM

Les Large Language Models (LLM) sont une branche de l'intelligence artificielle, plus précisément du domaine du traitement du langage naturel (TAL/NLP). Ayant connu une avancée majeure en 2017 grâce à la publication des « Transformers » [1], une architecture capable de traiter un grand nombre d'information, ces technologies n'ont eu de cesse de s'améliorer en corrélation avec l'augmentation de leur taille. Tout cela est rendu possible par la production en masse d'informations à travers le web (BigData) et à l'accroissement de la puissance de calcul GPU (taillé pour le calcul matriciel).

L'utilisation de LLM par le grand public et par les entreprises a été fortement démocratisée dès novembre 2022 par la société OpenAI et son célèbre ChatGPT. De nombreux géants du web ou startups ont rapidement suivi le pas : Bard, Gemini, Claude, Mistral, Llama (pour ne citer qu'eux). Principalement sous forme de tchat, il est possible d'interagir avec ces modèles grâce à une suite d'instruction appelée « prompt ».

Des centaines de cas d'usage ont émergé : rédaction de contenu, recherche documentaire, traduction, génération de code informatique, macros Excel, synthèse de document, vulgarisation, détection de sentiments et même génération d'image (lorsque couplé à un modèle de génération d'image). Ces nouveaux cas d'usage ne sont qu'une infime partie de ce qu'il est et sera possible avec les LLM.

Ainsi, aidés de ces nouveaux outils, les travailleurs voient leur capacité d'analyse et de production accélérée et améliorée. Au-delà des questions d'éthique et de transformation de la société [annexe 1], l'usage de ces IA par les entreprises est conditionné par deux facteurs majeurs : la souveraineté[1] et la précision des prédictions sur le secteur d'activité.

1.2 Souveraineté et RAG

Le principal frein à l'usage des LLM en entreprise est que ces outils sont mis à disposition sur des plateformes tierces, essentiellement détenues par des GAFAM, stipulant que les données transmises peuvent être réemployées pour entraîner les futurs modèles. Ainsi, pour ne pas faire fuiter des données internes à l'entreprise (fichiers clients, échanges de mails, tickets d'assistance, base de données produits, etc.), ces dernières ont rapidement interdit l'usage des LLM tiers pour leurs salariés [3], ce qui n'empêche pas leur utilisation en Shadow IT pour les avantages qu'ils procurent.

Les entreprises ayant les moyens ou une équipe dédiée ont rapidement mis en place leur propre service basé sur des modèles propriétaires. Pour les autres, il a fallu attendre l'arrivée des modèles open source courant 2023 tels que Mistral, Gemma ou encore Llama. Ainsi, l'hébergement du service dans le système d'information de l'entreprise permet de garantir la souveraineté financière, technologique et une sécurité dans le traitement des données.

Cependant, l'hébergement de LLM nécessite une forte puissance de calcul allant des dernières cartes de calcul à la mise en place d'un datacenter. Plus les modèles sont grands, plus ils nécessitent

1 Souveraineté : terme désignant la capacité à être autonome et non dépendant d'un tiers. On parle de souveraineté des données lors qu'on maîtrise leur usage de bout en bout. Cette notion est de plus en plus forte au sein de l'Union Européenne pour contrer l'hégémonie américaine ou chinoise.

de puissance de calcul et meilleurs sont les résultats de leurs prédictions. De nouvelles techniques ont néanmoins permis l'utilisation de modèles beaucoup plus petits, que l'on vient spécialiser sur certaines données, des données de l'entreprise ou du métier concerné. Appelées Retrieval Augmented Generation (RAG), elles permettent d'injecter des données spécifiques dans la construction des réponses que le service procure aux utilisateurs. Avec le RAG, le but est de construire une base de connaissance et de pouvoir s'appuyer dessus grâce à des calculs de similarité et sélectionner les bonnes sources d'informations à transmettre au LLM. Le modèle peut donc s'appuyer sur des éléments techniques dans son contexte afin d'améliorer la précision de sa prédiction.

2 Exploration de l'espace vectoriel

2.1 Fonctionnement d'un LLM

On peut résumer le fonctionnement de l'intelligence artificielle à des estimations statistiques.

C'est mouillé dehors → Il a plu.

C'est statistiquement vrai, mais comme vous pouvez l'imaginer, la cause peut être autre.

Le but est de prédire un résultat le plus probable en fonction des données d'observation ou de connaissance en entrée. Plus les données sont nombreuses et diversifiées, plus le modèle va être capable de s'adapter à une situation déjà rencontrée par son entraînement. Ainsi, nous sommes passés du « correcteur T9 » capable de corriger un mot ou de prédire le mot suivant, aux LLM, capables de générer une énorme quantité de texte.

Un programme informatique fonctionne sur des signaux électriques représentants des valeurs numériques. Il faut donc transformer les lettres, mots et phrases en nombres, en vecteurs plus précisément. Tout repose donc sur la façon dont on représente les mots, les phrases et les paragraphes dans l'espace vectoriel, aussi appelé « embeddings ».

Cet espace vectoriel peut faire des dizaines, des centaines de dimensions, mais dans l'exemple ci-après, on peut le représenter en deux dimensions.

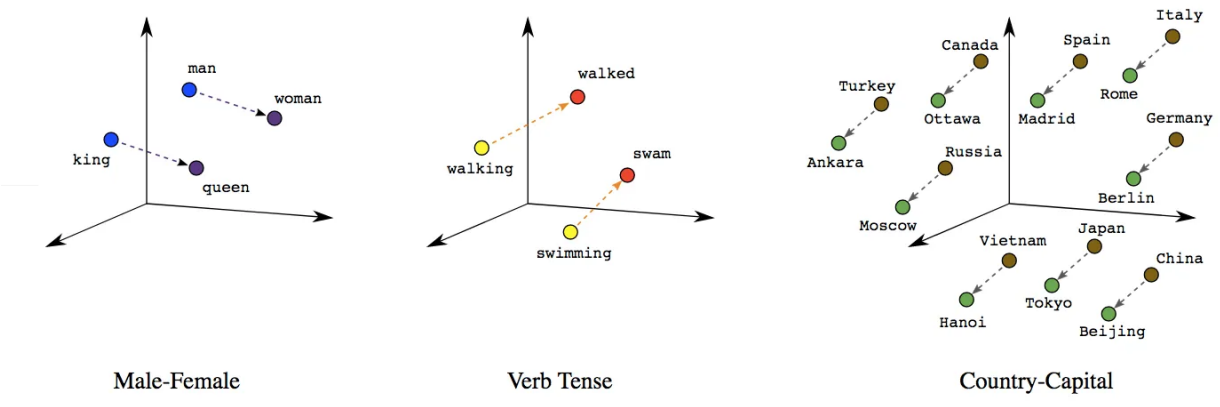

Figure 1: Représentation des embeddings

On entraîne donc un modèle à placer des mots dans l'espace de façon à donner du sens à la position et à un déplacement dans cet espace. Ainsi, des concepts proches ou similaires vont se retrouver dans une même région et un concept de « féminisation » va être déplacement parallèle sur un plan.

Roi → Reine // Homme → Femme

C'est cette représentation vectorielle qui permet au LLM de « comprendre » (*je ne le répète jamais assez, ce ne sont que des statistiques*) le sens d'un mot, d'une phrase ou d'un texte. Plus cet espace est grand (nombre de dimensions), plus on peut porter de concepts différents et donner de la profondeur à la compréhension d'un texte.

2.2 Just read the instructions

Pour interagir avec un LLM, on utilise bien souvent une invite de texte, appelé « prompt ». C'est en quelque sorte le chemin à suivre pour diriger la réponse dans l'espace vectoriel. Plus les informations données en contexte sont nombreuses et précises, plus le LLM va pouvoir prédire une réponse qui se rapproche de la réponse attendue.

« *Un Large Language Model hallucine tout le temps. Il peut arriver que cette hallucination soit juste* »

En suivant les instructions du contexte, il est possible d'emmener le LLM vers un point de l'espace vectoriel qui existe mathématiquement, mais qui est absurde au niveau sémantique, par exemple : les œufs de vache, des tomates-cerises poussant dans les arbres, etc.

2.3 Couplage du RAG

En résumé, pour prédire un résultat, un LLM s'appuie sur ses données d'entraînement et sur des instructions données en contexte.

Par exemple pour entraîner un modèle à générer des réponses sur des tickets support informatiques, il nous faudrait des millions d'exemples de questions utilisateur et des réponses du service d'assistance technique formatées de la même manière.

On comprend alors qu'il est difficile pour l'écrasante majorité des entreprises d'avoir suffisamment de données pour entraîner leur propre LLM, il reste donc la possibilité de jouer sur les informations contenant des éléments de réponse au problème dans la question.

En injectant ces éléments supplémentaires dans le contexte, on permet au LLM de générer une réponse plus pertinente et au format attendu. A travers différents outils d'indexation, de recherche et de configuration de prompt, on peut utiliser des données de l'entreprise pour alimenter le LLM à la volée. Il n'y a pas d'entraînement supplémentaire (fine-tuning), donc aucune donnée n'est apprise par le LLM, perdant ainsi en vitesse mais gagnant grandement en facilité d'implémentation.

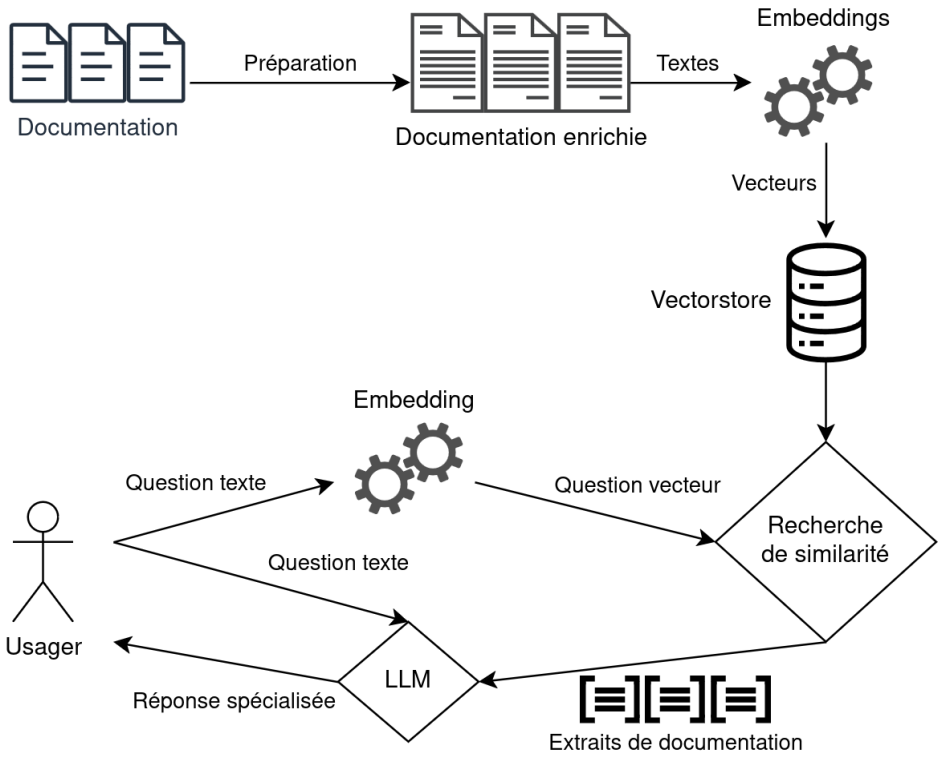

Figure 2: Fonctionnement du RAG

La figure 2 montre le fonctionnement du RAG. Au préalable, la documentation est préparée et enrichie, puis transformée en vecteurs de texte (embeddings) qui sont stockés dans une base de données dédiée (vectorstore). Par la suite, lorsqu'un usager pose une question, celle-ci est convertie en vecteur et comparée aux vecteurs de la documentation dans un processus de recherche par similarité. Les extraits les plus pertinents sont sélectionnés et envoyés à un modèle de langage (LLM), qui génère une réponse contextualisée à partir des informations fournies.

3 Cas d'usage : chatbot d'assistance aux usagers d'une DSI

3.1 Explication du cas d'usage et contraintes

Pour la suite nous allons prendre un cas d'usage bien connu des services d'assistance informatique aux usagers, à travers la gestion de tickets. Nombre de tickets peuvent être résolus facilement en lisant la documentation ou la FAQ fournie aux usagers. Pour autant, l'usager pour diverses raisons, ne va pas voir cette documentation. On souhaite donc brancher une IA qui va lire le ticket avec la question de l'usager et prédire une réponse pour que l'agent d'assistance puisse avoir une suggestion de réponse automatique.

La contrainte principale c'est la puissance de calcul. Peu de DSI peuvent se targuer d'avoir une ferme de GPU dernière génération pour faire tourner des modèles puissants. Nous allons donc par la suite prototyper un programme qui puisse fonctionner dans des configurations communes : du CPU entre 24 et 48 cœurs, sur un serveur. La RAM ou le stockage sont négligeables. Comme nous n'allons pas entraîner de modèle, mais utiliser seulement l'inférence (la prédiction), l'usage du CPU n'est pas extrêmement pénalisant (de l'ordre de 5 % à 20 % de la vitesse en GPU). Sur le cas d'usage sélectionné, même si le modèle met quelques minutes à s'exécuter, l'agent d'assistance ne

met généralement pas moins de 5 à 10 minutes à voir et répondre au ticket, laissant largement le temps au programme de tourner en tâche de fond.

Bien sûr, pour des cas d'usage nécessitants du quasi temps réel, il faudra donc acquérir des cartes de calcul GPU.

Nous allons utiliser un modèle relativement léger afin de gagner en performances sur le CPU, mais la qualité des réponses s'en retrouve dégradée. Pour améliorer les résultats et spécialiser la réponse sur notre documentation interne, nous allons coupler un modèle LLM avec des techniques de RAG afin de lui donner des éléments supplémentaires.

La question utilisateur servant de test est « comment configurer mon Thunderbird pour utiliser mon adresse email universitaire ? ».

3.2 Installation d'un LLM open source

Nous allons utiliser le langage Python pour les exemples ci-après, c'est de loin le langage le plus utilisé pour l'IA, mais il est très perfectible en terme de vitesse. Des implémentations dans d'autres langages sont disponibles.

La plupart des modèles open source sont hébergés sur HuggingFace, un hub façon github mais dédié à l'IA. Une librairie nommée Ollama, disponible en bash ou python, permet de faciliter l'installation d'un modèle de langage. Nous allons utiliser un modèle léger, Gemma2:2B, petit frère de Gemini de Google. Selon les ressources, un modèle comme Mistral:7B peut améliorer les prédictions, et garantir encore plus la souveraineté, Mistral étant une startup française.

```
curl -fsSL https://ollama.com/install.sh > install_ollama.sh
sh install_ollama.sh
ollama pull gemma2:2b
ollama run gemma2:2b
```

Au bout de ces quelques lignes, il est déjà possible d'utiliser un tchat depuis le terminal !

Selon le modèle choisi, les réponses peuvent être plus ou moins précises. Pour le moment le modèle n'est donc pas spécialisé sur nos données internes.

3.3 Pré-traitement des données de spécialisation

Après avoir extrait la documentation interne, au format texte, il faut le normaliser. Dans notre cas, la documentation est extraite en HTML, il faut retirer toutes les balises grâce à « unescape » en python, puis sauvegarder le dataset dans un répertoire de travail, en multiples fichiers .txt.

Une stratégie consiste à passer un à un les fichiers dans un LLM afin de générer automatiquement des éléments clés du texte dans le but de rendre son contenu plus accessible et compréhensible lors d'une recherche par similarité. Par exemple : reformer chaque fichier sous la forme d'une question.

Ces données vont servir par la suite dans un vectorstore défini ci-après. Il est donc nécessaire d'enrichir au maximum les données initiales. Le temps de calcul est négligeable puisqu'il suffit de sauvegarder les données enrichies une fois cette étape terminée par le LLM.

Il est également possible de faire des résumés ou des reformulations à ce stade du traitement des données.

3.4 Vectorstore

Pour fonctionner correctement, le LLM a une certaine taille maximale en entrée, il est donc impossible de lui passer l'intégralité de la base de connaissances dans le contexte. Nous allons donc indexer le contenu avec un vectorstore, qui va nous permettre de calculer des similarités sur des vecteurs et pour pouvoir extraire très rapidement des informations. Le calcul de la similarité permet de sélectionner les documents les plus pertinents à injecter dans le contexte. Ainsi, on limite grandement l'explosion de la taille du contexte.

Il est cependant recommandé de découper nos documents en bouts de textes afin de rendre le calcul de similarité plus précis. On peut jouer avec les paramètres de taille et de recouvrement pour obtenir de meilleurs résultats en fonction de son dataset.

Comme ce calcul de similarité se fait entre la question de l'usager et les extraits de documentation enrichie précédemment, le fait de générer des questions types auxquelles répondent les documentations initiales, il est plus probable d'obtenir le bon document répondant à la question de l'usager.

Le fait de travailler dans l'espace vectoriel a pour avantage de s'exécuter très rapidement, mais a pour inconvénient de passer à côté de la richesse des textes en eux-mêmes, puisqu'ils sont réduits à leur état vectoriel.

On peut donc coupler cette recherche documentaire avec d'autres outils. Tout d'abord, par le LLM lui-même, en lui demandant pour chaque document, est-ce que ce document répond à la question. On notera un allongement des temps de calcul, surtout dans notre cas en CPU. Il est donc préférable d'utiliser cette méthode en complément de la première. On récupère « n » documents qui sont proches dans l'espace vectoriel, puis on les repasse dans le LLM pour faire un re-ranking, pour les reclasser en allant dans la richesse du texte.

On peut également voir dans la littérature l'usage d'ElasticSearch pour cette phase, mais je n'ai pas pu la mettre en place à date de cet article.

3.5 Définition du prompt template

Nous allons donc maintenant préparer l'agent IA afin de le guider à répondre correctement à la demande de l'usager. Nous allons pour cela définir un prompt template comme l'exemple ci-après :

```
Tu es un assistant en informatique. Tu réponds en français sans emojis ni
smileys.
Tu fais partie de <entreprise> et tes réponses sont en lien avec le domaine
univ.fr.
Tu dois répondre à la question suivante :
{input}
```

Cet agent simple permet de répondre à la question de l'usager selon des paramètres souhaités : en français, sans emojis, etc.

Une fois que vous avez récupéré les documents susceptibles de répondre à la question comme vu précédemment, on peut les injecter dans le template :

```
Tu es un assistant en informatique. Tu réponds en français sans emojis ni
smileys.
Tu fais partie de <entreprise> et tes réponses sont en lien avec le domaine
univ.fr.
Tu dois t'appuyer sur les éléments supplémentaires fournis en contexte.
{context}
Tu dois répondre à la question suivante :
{input}
```

Selon les tailles de document, soit il est possible de tous les injecter ensemble, soit il faudra faire plusieurs passes et choisir la meilleure proposition de solution.

Enfin, il est possible de continuer l'interaction avec le modèle en conservant un historique de conversation, pour affiner la demande si la réponse est incomplète.

```
Tu es un assistant en informatique. Tu réponds en français sans emojis ni
smileys.
Tu fais partie de <entreprise> et tes réponses sont en lien avec le domaine
univ.fr.
Tu dois t'appuyer sur les éléments supplémentaires fournis en contexte.
{context}
Voici l'historique de la conversation :
{chat_history}
Tu dois répondre à la question suivante :
{input}
```

4 Résultats

En repartant de la question « Comment configurer mon Thunderbird pour utiliser mon adresse email universitaire ? », le modèle seul va simplement sortir ce genre de résultat :

```
Pour configurer votre adresse email dans Thunderbird, suivez ces étapes :


1. **Ouvrez Thunderbird:** Lancez l'application sur votre ordinateur.

2. **Accédez aux paramètres:**  Cliquez sur "File" puis "Account Settings".

3. **Ajoutez un compte:** Sélectionnez "Add Account" et suivez les instructions
à l'écran pour entrer vos informations d'accès (nom d'utilisateur, mot de
passe, etc.).

4. **Sélectionnez votre serveur:** Choisissez le type de serveur (POP3 ou IMAP)
selon la configuration de votre fournisseur d'email.

5. **Configurez les paramètres:**  Indiquez les paramètres spécifiques à votre
compte (ex: nom de domaine, port, etc.) et cliquez sur "Next".


**Note:** Si vous rencontrez des difficultés, consultez la documentation de
Thunderbird ou contactez le support technique de votre fournisseur d'email.
```

Au travers du processus d'enrichissement du RAG, on va pouvoir injecter des éléments de documentation comme la configuration imap.univ.fr ou pop.univ.fr, le port à utiliser ou divers éléments permettant d'obtenir un résultat comme :

```
Pour configurer Thunderbird, suivez ces étapes:


**IMAP (Configuration des mails entrants):**


1. Allez dans le menu en haut à droite > Paramètres des comptes.
2. Sélectionnez ensuite "Paramètres serveur" pour les mails qui rentrent.
3. Renseignez les informations suivantes :
    * **Nom du serveur:** imap.univ.fr
    * **Port:** 143
    * **Nom d'utilisateur:** votre login université
```

```
**SMTP (Configuration des envois de mails):**


1. Allez dans le menu gauche > serveur sortant (SMTP) pour l'envoi.
2. Renseignez les informations suivantes :
    * **Nom du serveur:** smtp.univ.fr
    * **Port:** 587
    * **Sécurité de la connexion:** STARTTLS
    * **Nom d'utilisateur:** votre login université


**Important:**  Chaque étape vous demandera votre mot de passe, celui de votre compte universitaire.
```

C’est typiquement le genre de réponses que peuvent fournir les agents d’assistance technique !

Dans le cadre de cette expérimentation, une base de connaissances comprenant environ 70 articles de FAQ a été utilisée. En employant le modèle Gemini2:2b sur un serveur équipé de processeurs CPU, la phase de pré-traitement des données, incluant l'extraction des embeddings et la construction du vectorstore, a nécessité jusqu'à 2 heures. Cette durée s'explique principalement par les contraintes liées à l'infrastructure CPU, en l'absence de l'accélération GPU. Cependant, ce temps de pré-traitement reste raisonnable pour des mises à jour périodiques de la base de connaissances.

Lors de la phase d'inférence, où le système génère des suggestions de réponses pour les demandes issues de tickets, le temps de réponse a montré une variabilité importante, oscillant entre quelques secondes et 5 minutes. Ce délai dépend de plusieurs facteurs, tels que la complexité de la requête, la longueur des documents pertinents à analyser, et la charge de calcul du serveur au moment de l'interrogation. Malgré ces variations, le système a prouvé sa capacité à fournir des réponses de manière autonome et pertinente, en restant dans des délais globalement acceptables pour une utilisation opérationnelle.

5 Conclusion

Ces résultats démontrent la faisabilité d'un système RAG (Retrieval-Augmented Generation) sur des infrastructures à ressources limitées, comme des serveurs CPU. Bien que des améliorations soient envisageables pour optimiser les temps de traitement, notamment via l'utilisation de matériel dédié comme des GPU, l'approche actuelle offre déjà une solution viable pour la gestion de bases de connaissances de taille modérée. Elle permet de répondre efficacement aux besoins d'assistance en temps quasi réel, tout en maintenant une infrastructure matérielle relativement accessible.

Le prototype va prochainement être déployé en test, il reste à développer le connecteur de sortie vers l’outil de gestion de ticket GLPI afin de pouvoir intégrer la suggestion de réponse par IA en tant que pré-réponse à la demande de l’usager. On va également pouvoir utiliser les anciens tickets comme base de connaissance supplémentaire afin d’élargir les possibilités de réponses. Enfin, après une certaine durée de test, si la fiabilité est suffisante, on va pouvoir envoyer directement les réponses aux usagers sans nécessité d’action ou de validation humaine.

Annexe

Peurs liées à la transformation du travail

« L'IA ne va pas prendre votre emploi, ce sont plutôt les personnes capables de s'en servir pour être plus productives qui le feront », d'après Roberto Saracco [2].

Bien que le sujet de cet article ne traite pas directement la question néanmoins importe de l'éthique et de la transformation du monde du travail avec l'arrivée massive de l'IA, il est tout de même nécessaire de mentionner les inquiétudes et les débats en cours. Tout comme la révolution industrielle du XIXe siècle, de l'arrivée des lignes de production automatisées ou de l'essor d'internet, ces changements majeurs ont supprimé des emplois, mais de nouveaux s'en sont créés. S'il est encore incertain aujourd'hui de la capacité de la société à s'adapter à ce bouleversement rapide, il est déjà possible de savoir que l'IA offre un avantage concurrentiel à ceux qui vont s'en servir.

Bibliographie

[1] Attention Is All You Need (2017) : Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin ; https://arxiv.org/abs/1706.03762

[2] Roberto Saracco (2023) pour l'IEEE : https://intelligence-artificielle.developpez.com/actu/343163/-L-IA-ne-va-pas-prendre-votre-emploi-ce-sont-plutot-les-personnes-capables-de-s-en-servir-pour-etre-plus-productives-qui-le-feront-d-apres-Roberto-Saracco-membre-senior-de-l-IEEE/

[3] https://aiexplorer.io/actualites-ia/pourquoi-chatgpt-est-il-interdit-dans-certaines-entreprises/

Merci à Laurence Moindrot et Éric Jullien du comité d'organisation JRES pour la relecture.

Cet article a été rédigé par des humains assistés par l'IA (essentiellement de la reformulation).