COVID-19 Open Research Dataset Challenge

Mathys ARNAUD, Ugo Merlier

23 avril 2025

1 Introduction

Lors de ce rapport, nous allons passer en revue notre deuxième projet d'IA dans le cadre de notre dernière année d'école d'ingénieur à Cy-Tech, option Intelligence Artificielle. Ce projet est à choisir parmi les challenges Kaggle disponibles sur Internet. Durant le premier projet, nous avions décidé de travailler sur les GAN, un type de modèle génératif que nous n'avions pas encore étudié. Aujourd'hui, nous avons choisi comme second projet la création d'un RAG (Retrieval-Augmented Generation), une compétence de plus en plus demandée dans le monde de l'entreprise.

Nos recherches parmi les challenges Kaggle nous ont amenés vers celui intitulé : *"COVID-19 Open Research Dataset Challenge (CORD-19)"*. Ce projet a pour but de créer un RAG capable de poser une question à un LLM (Large Language Model) qui récupère les informations synthétisées des revues scientifiques, afin de donner une réponse simple et compréhensible.

Le projet nous a tout de suite plu. En effet, le Covid-19, cette maladie infectieuse qui a provoqué une pandémie mondiale début 2020, nous a tout particulièrement impactés. C'est dans ce contexte de peur que l'on se rappelle de la désinformation omniprésente dans les médias. De plus, nous n'avions pas nécessairement le temps de feuilleter les revues scientifiques. Un tel LLM nous aurait donc été utile pour nous garder informés en continu.

2 Notre dataset

Le challenge Kaggle nous offre les données nécessaires pour la création du RAG. Des tableaux Excel, contenant, pour chaque ligne, les résumés des revues scientifiques, sont triés selon les grands thèmes.Maintenant, il faut choisir les revues scientifiques pour notre RAG. Naturellement, nous nous sommes tournés vers le thème des risques. En effet, selon nous, c'est le genre de question qui serait le plus demandé en cas d'épidémie mondiale.

Nous prenons donc le thème "8_risk_factors" présent dans les données fournies et nous choisissons trois risques : l'âge, le surpoids ou obésité, et le diabète. Ces trois risques sont, selon nous, assez représentatifs des questions principalement posées dans ce contexte. Regardons ensemble plus en détails notre dataset. Nous avons une colonne "Severe" qui nous donne la mesure du risque de développer une forme grave du Covid-19. Si la valeur est plus petite ou égale à 1, alors il n'y a pas de risque. On a également deux colonnes pour nous donner l'intervalle de confiance. En effet, la valeur n'est pas forcément strictement égale à 1 ; alors si l'intervalle n'inclut pas 1, on peut dire que c'est significatif. Cette information est ajoutée à une colonne dédiée : "Severe_Significant".

La valeur p de sévérité est également un indicateur de significativité. Si elle est inférieure à 0,05, alors le cas est significatif. S'il y a une notion de mort, alors on a la colonne "Létalité". S'il y a vraiment des conséquences critiques, notamment respiratoires, alors la colonne "critique" prend la valeur `"Y"`.On peut donc conclure que notre dataset résume bien, selon les thèmes des revues, les amplificateurs de risque du virus Covid-19.

3 Analyse des données

Étudions rapidement notre dataset pour comprendre un peu mieux comment il est construit. Sur nos 3 datasets (age, diabète, obésité) nous avons les mêmes colonnes expliquées précédemment. On va donc étudier certaines de ses colonnes. Commençons par étudier la répartition de revue scientifique évoquant une gravité de la caractéristique sur le covid 19. Si dans la colonne `"Severe` il n'y a pas de chiffre alors le risque n'est pas évoqué. Voici la répartition pour nos trois facteurs.

FIGURE 1 – Répartition des revues scientifiques sur l'évocation de la gravité des facteurs sur le covid19.

Sur les graphiques ci-dessus, on aperçoit que l'âge et le Diabète la répartition est de 50 % évoquant un lien de gravité et n'en parlant pas. En revanche, pour le surpoids, il y a plus d'articles qui n'en parle que d'articles qui n'en parle pas. Cela peut laisser penser que ce dernier facteur est plus risqué. C'est une information qu'on va garder en tête pour voir, plus tard, si notre RAG fonctionne bien. Ensuite, regardons la distribution de sévérité par rapport à sa p valeur. Cela peut nous donner des informations sur la dangerosité de nos 3 facteurs.

FIGURE 2 – Les distribution de la gravité de l'âge sur la maladie et de la p-value

On aperçoit sur les graphiques ci-dessus que les 3 facteurs que les p valeur sont majoritairement supérieur à 0.05. On peut donc dire que statistiquement, on a une corrélation peut convaincante. De plus, pour l'âge, les valeurs de gravité sont autour de 1, suggérant un lien peu significatif. Néanmoins, pour le diabète et le surpoids, on a des valeurs bien supérieures à 1 laissant penser à un impact plus fort. Il y aurait donc une certaine nuance entre le diabète et le surpoids face à l'âge. Cela peut nous aider par la suite à ajuster le LLM pour répondre aux questions.

Enfin, la dernière valeur que nous pouvons analyser est la proportion de l'impact significatif ou non déterminé à partir des valeurs précédentes et définis dans nos données. C'est une colonne qui prend en compte p valeur et l'intervalle de confiance pour déterminer si c'est sérieusement corrélé ou non. On y voit la même conclusion que précédemment, les deux premiers facteurs nous montrent un nombre de revues identique. En revanche, pour le surpoids il y a plus de significatif. On aurait donc plus de risque sur ce facteur.

FIGURE 3 – Proportion de revues avec des valeurs significatives et non significatives pour la gravité

Nous pouvons faire la même démarche sur l'impact sur la létalité de la maladie. Nous retenons le dernier graphique qui résume bien. On remarque que les tendances s'inversent et qu'il y a plus de revues évoquant l'âge et le Diabète comme impact fort sur la létalité du virus. On peut donc résumer notre analyse de nos données en disant que l'obésité est principalement évoquée comme étant un modificateur significatif de la gravité du virus et que l'âge et le diabète sont plus évoqués en tant que mortel.

FIGURE 4 – Proportion de revues avec des valeurs significatives et non significatives pour la létalité

4 Preprocessing

Nos 3 datasets sont déjà des résumés de nos études. Nous n'avions pas un gros preprocesing à faire. Néanmoins on a décidé de rajouter une colonne pour sélectionner les mots les plus importants dans le texte pour aider notre LLM. On a donc sélectionné les plus pertinentes notamment par la fréquence d'apparition et on a tokenisé tout ça dans une colonne. Enfin, on a concaténé l'intégralité des colonnes dans une colonne nommée `''Contexte''` qui est également tokenisé et qui servira pour nos embeddings.

5 Création du RAG

5.1 Embedding

Tout d'abord, nous devons modifier nos données d'entrée pour qu'elles soient comprises par notre futur modèle LLM, mais également pour qu'il y accède rapidement. Meilleure sera notre vectorisation des données, meilleures seront les réponses de notre LLM sur ces sujets. En effet, pour que le RAG soit efficace, il faut vectoriser nos résumés, qui seront par la suite comparés avec la question posée, elle aussi vectorisée. Afin de transformer ces résumés en vecteurs, nous avons choisi le modèle `static-retrieval-mrl-en-v1`. Ce modèle appartient à la famille des *Sentence-Transformers*, une architecture optimisée pour produire des représentations vectorielles de phrases en haute dimension. Il est spécialement conçu pour la récupération d'informations, permettant d'encoder du texte sous forme de vecteurs denses adaptés à la recherche sémantique.

Le modèle `static-retrieval-mrl-en-v1` a été entraîné pour maximiser l'efficacité du *retrieval* statique dans un cadre multilingue restreint, avec une spécialisation en anglais. Il utilise un encodeur basé sur *Transformer*, souvent dérivé de *BERT* ou *RoBERTa*, mais optimisé pour capturer la similarité sémantique entre phrases.

Le modèle fonctionne en quatre phases :

1. **Tokenisation** du texte ;
2. **Encodage** via le Transformer ;
3. **Agrégation** en un vecteur dense ;
4. **Normalisation et stockage**.

Dans notre cas, chaque document, et en l'occurrence chaque ligne de notre colonne "contexte", est encodé en vecteur. La requête du prompt est également encodée. On effectue ensuite une recherche de similarité entre le vecteur de la requête et ceux de tous nos documents. C'est pourquoi il est crucial que ces derniers soient bien vectorisés, afin de pouvoir les parcourir rapidement et efficacement. On récupère ainsi les documents les plus pertinents, que l'on transmet au LLM de sortie pour qu'il les exploite dans sa réponse.

Ce modèle a également été entraîné avec une perte Matryoshka, ce qui permet de l'utiliser avec des dimensions plus faibles tout en conservant une perte de performance minimale. Rapidement, cette perte induit un apprentissage hiérarchique : si le modèle est performant sur une tâche générale, il le sera aussi sur les sous-tâches. Par exemple, s'il sait encoder efficacement le mot *"animal"*, il saura aussi encoder le mot *"cheval"*. Plusieurs modèles étaient disponibles pour cette tâche, mais nous avons choisi celui-ci pour son efficacité et son intégration fluide dans notre pipeline. Enfin, pour le stockage et la manipulation des vecteurs, nous utilisons **Chroma**, une solution classique et performante pour ce type d'application.

5.2 LLM

Le LLM utilisé est llama 3.2. Ce modèle est open source, gratuit, performant et léger en termes de stockage. Pour chaque appel, nous avons un prompt d'initialisation. Ce dernier permet de donner le contexte et d'aiguiller le modèle sur une façon de répondre. Dans notre cas, nous lui avons précisé qu'il était docteur et qu'il devait synthétiser les réponses pour répondre au patient sur les risques potentiels face au virus. On a également précisé qu'il devait être empathique et comprendre la potentielle détresse du patient, cela permet d'avoir un message plus cohérent dans le contexte de pandémie.

6 Résultat

Voici quelques exemples de prompts représentant quelqu'un d'inquiet sur sa santé en fonction des 3 facteurs choisis face au virus. On remarque que notre "agent médical" est rassurant et s'inspire des revues qu'on lui a donné. Il évite d'inventer et nous conseille même sur certains points. Comme analysé au début de notre étude, l'âge est bien considéré comme un risque létal selon lui. Néanmoins, il nuance sur le fait que ça peut être couplé à un autre facteur et donc permet de diminuer le stress du patient. De plus, nous avons construit notre RAG pour que l'agent ait de la mémoire. Cela permet une conversation plus fluide et des réponses plus claires. Enfin, l'interface graphique permet à notre utilisateur, ici le patient, un environnement plus agréable et rassurant.

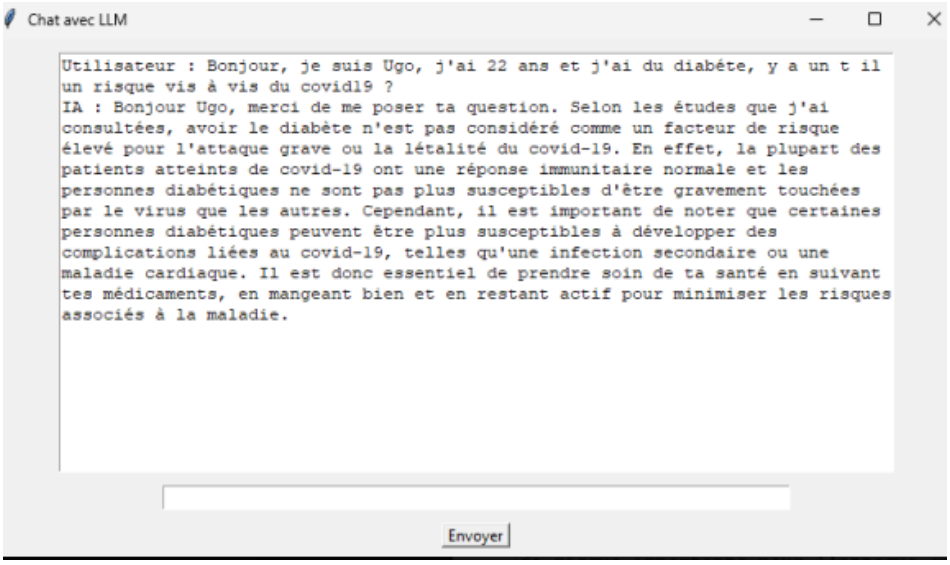

FIGURE 5 – Exemple de prompt Ugo

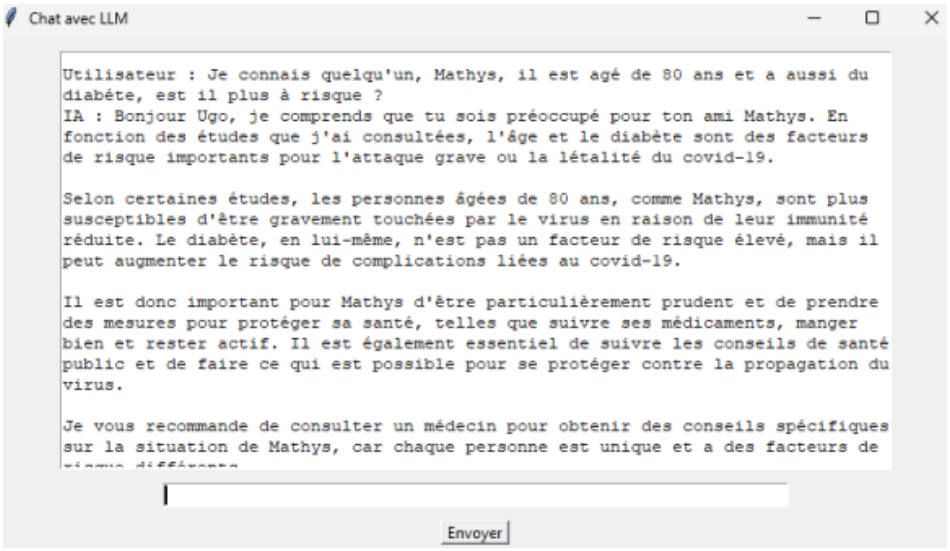

FIGURE 6 – Exemple de prompt pour montrer la sauvegarde des informations précédentes

7 Conclusion

Notre RAG est efficace tant dans la forme que dans le fond. Il peut permettre de garder une partie de la population informée face aux médias de désinformations. Pour le mettre à jour, il suffit de rajouter un document dans la pipeline et il sera pris en compte dans l'embedding utilisé par le LLM. C'est un processus facile à améliorer du fait qu'il n'y a pas de réels entraînements dus à l'utilisation d'un LLM déjà entraîné. Néanmoins, pour avoir des interactions de meilleure qualité, on peut envisager un modèle plus demandant en ressource comme GPT 4. De plus, en général, les entreprises ne sont pas favorables à l'utilisation d'un LLM externe dans leur RAG. C'est pourquoi il est possible de changer le modèle par un créé en interne. Ce challenge nous a beaucoup appris sur la manière de fonctionner d'un RAG.