

Contexte

Depuis l'arrivée de l'envahisseur vert, les nouvelles espèces pathogènes ont explosé. Les équipes médicales, pour ce qui en reste, sont débordées et ne suffisent pas à répondre à la demande. Pour réduire la charge de travail qui repose sur leurs épaules, vous devrez créer un chatbot simple visant à identifier des problèmes de santé en fonction de symptômes. Ce triage automatisé permettra de rediriger les malades vers les bonnes ressources et ainsi éviter d'engorger un système déjà mis à mal.

Objectif

Votre défi est de créer un chatbot simple qui permet de faire un triage préliminaire de potentiels futurs patients en fonction de leurs symptômes. Celui-ci doit récolter les informations en lien avec l'affection de l'utilisateur, les utiliser pour questionner un modèle AI et lui transmettre un diagnostic potentiel de même que lui recommander des actions à prendre au besoin.

Vous serez évalué sur la capacité de votre outil à guider l'utilisateur, à l'interroger de façon pertinente, à poser un potentiel diagnostic et à le lui communiquer.

Scénario

L'utilisateur est aux prises avec des symptômes qu'il croit être dus à une infection respiratoire. Incertain de la gravité de la situation, il utilise l'outil pour déterminer la sévérité de ses symptômes et les actions à prendre.

Paramètres

Interface

L'interface du chat est réalisée en HTML/JavaScript.

Des fichiers de départ sont fournis;

- Index.html contient le DOM du chat
- Style.css définit le UI du chat
- Index.js contient les fonctionnalités de base du chat

Le visuel et les fonctionnalités de base du chat sont comprises dans les fichiers de départ.



Référez-vous à [MDN Web Docs](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/JSON) si vous n'êtes pas familier avec ces technologies.

Modèle

Le modèle utilisé pour le défi est [medalpaca-7b](#).

Le modèle est pré-entraîné sur des données médicales.

Le modèle comprend le langage naturel.

Le modèle est hébergé dans un cloud et n'a pas à être installé localement.

Des snippets de code pour effectuer les appels à l'API du modèle sont fournis dans le fichier `query.md`.

Référez-vous à la [documentation](#) pour choisir et configurer vos paramètres.

Chatbot

Le bot est en charge d'échanger en langage naturel avec l'utilisateur via le chat.

Le bot est en charge d'échanger avec le modèle via l'API.

La logique du bot peut être faite dans le langage de votre choix, nous recommandons toutefois JavaScript ou Python.

Restrictions

L'utilisation d'un autre modèle est interdite.

L'utilisation d'outils de génération (chatGPT, Copilot, etc) est interdite.

Seules les pages web listées dans le présent document peuvent être accédées pendant la tenue du défi.

Attentes

Le bot devrait ...

- Fournir le contexte de l'outil
- Guider l'utilisateur dans l'utilisation
- Récolter les informations nécessaires
- Questionner le modèle AI

L'utilisateur devrait comprendre que ...

- L'échange se fait avec un bot et non un humain
- Le contexte est médical
- Les informations obtenues visent à effectuer un triage
- Le diagnostic posé est à titre indicatif seulement (non officiel)

Le participant doit ...

- Compléter le code du chat (`index.js`) pour permettre l'échange entre le bot et l'utilisateur
- Programmer la logique du bot pour récolter les informations de l'utilisateur
- Interroger le modèle pour poser un diagnostic et effectuer le triage
- Expliquer sa démarche pour établir les informations à récolter

Stratégie

Vous devrez faire du *prompt engineering* afin d'ajuster les échanges du bot avec l'utilisateur dans le but d'obtenir les meilleurs résultats possibles.

Tous les échanges devraient être faits en langage naturel.

Le [*prompt engineering*](#) est le processus par lequel vous guidez une solution d'intelligence artificielle générative (*generative AI*) pour générer les résultats souhaités.

Le traitement automatique du langage naturel (TALN), en anglais *natural language processing* (NLP) est la capacité d'un programme informatique à comprendre le langage humain tel qu'il est parlé et écrit.

Considérations

Exemples d'informations à recueillir :

- Liste des symptômes
- Depuis quand les symptômes sont présents
- Présence de forte fièvre

Exemples d'actions à prendre :

- Utiliser de la médication sans prescription pour aider les symptômes
- Consulter un professionnel si les symptômes s'aggravent
- Se présenter aux urgences

Exemples d'expérimentations :

- Questionner le modèle de différentes façons sur un même sujet
- Questionner le modèle sur un sujet pour en obtenir les détails
- Tester différentes configurations des paramètres

L'usage de l'anglais risque de produire de meilleurs résultats lors du questionnement du modèle.

Remise

Vous devez soumettre avant la fin du défi:

- Le code de votre projet
 - Créer un repo Git public et nous faire parvenir le lien
- Un README détaillé expliquant comment exécuter le projet
 - Le projet doit pouvoir être lancé
- Un rapport détaillant votre processus de *prompt engineering*
 - Le/les stratégies utilisées
 - Les paramètres configurés
 - Le processus d'expérimentation
 - Tout autres tests ou réflexions menant à raffiner l'échange
- Deux scénarios différents à tester
 - Décrire l'échange à effectuer
 - Spécifier le résultat attendu

- Un scénario devrait mener à consulter un professionnel et l'autre non

Pointage

L'expérience utilisateur, le compte rendu de même que la logique du bot seront évalués.

Voici les différents critères d'évaluation :

- Fonctionnalités et expérience (UX) : 300 points
- Démarche et expérimentations (prompt engineering): 400 points
- Pertinence, capacité et justesse (code): 700 points
- Score maximum possible: 1400 points