Dakar Institute of Technology

Habilitation n°3386-21 FEV.2024

Domaine : Sciences et Technologies

Département : Informatique

Spécialité : Intelligence Artificielle

MÉMOIRE

Présenté par

Lauriane MBAGDJE DORENAN

Pour l'obtention du diplôme de

Master en Informatique (Option : Intelligence Artificielle)

SUJET :

MISE EN PLACE D'UN SYSTEME DE PRECONSULTATION ET D'ORIENTATION VERS LES STRUCTURES SANITAIRES

Soutenu à Dakar le 24/01/2025 devant le jury composé de :

Prénom & Nom	Grade / Titre	Structure de rattachement
Président : Cherif Bachir DEME	**Professeur**	**UADB**
Superviseur : Abdoul Wahab DIALLO	**Ingénieur**	**DIT**
Examinateur 1 : Abdoulaye BARRO	**Ingénieur**	**RUBYX**
Examinateur 2 : Madické DIOP	**Ingénieur**	**DIT**

Année Académique : 2023 – 2024

DÉDICACE

À la mémoire de mon oncle, défunt Innocent BEUGEU, qui m'a transmis sa passion pour la science et a illuminé mon chemin vers la connaissance. Ce mémoire, que tu ne pourras voir, est le fruit de tes précieux enseignements.

En hommage à mon ancien directeur du collège, défunt Monsieur Alladoum DILLAH, dont la vision de l'excellence et les conseils avisés continuent de guider mon parcours académique. Votre sagesse reste une source d'inspiration.

À ma famille, et particulièrement à mon père et à ma mère, ce modeste travail est le fruit de votre amour inconditionnel et de votre soutien indéfectible. Vos sacrifices et votre foi en moi ont été ma plus grande force. Au-delà des mots, ce mémoire témoigne de la persévérance que vous m'avez inculquée et des valeurs que vous m'avez transmises. Puisse-t-il être à la hauteur de tout ce que vous m'avez donné.

REMERCIEMENTS

Je remercie avant tout Dieu, le Tout-Puissant, pour ses grâces infinies et ses merveilles, qui m'ont soutenue et guidée tout au long de la réalisation de ce projet et de la rédaction de ce mémoire.

À ma famille, ce travail est l'aboutissement de votre amour et de votre soutien sans faille. Vos sacrifices, votre foi en moi et les valeurs que vous m'avez transmises ont été ma plus grande source de force et d'inspiration. Merci pour tout.

Je tiens à exprimer ma profonde gratitude à mon encadreur, **M. Abdoul Wahab DIALLO**, pour son accompagnement précieux, ses conseils avisés et sa disponibilité tout au long de ce projet.

Mes remerciements s'adressent également au corps professoral et administratif du **Dakar Institute of Technology (DIT)**, dont la qualité de l'enseignement et l'engagement envers la réussite des étudiants ont largement contribué à ma formation.

À mes chers camarades de la deuxième promotion du Master en Intelligence Artificielle du DIT, merci pour votre solidarité, votre esprit d'équipe et vos encouragements constants qui ont enrichi mon parcours.

Enfin, une mention particulière à **M. Amath SOW**, ainsi qu'à toutes les personnes qui ont, de près ou de loin, contribué à la réalisation de ce travail. À vous tous, merci infiniment.

GLOSSAIRE

DIT : Dakar Institute of Technology

IA : Intelligence Artificielle

LLMs : Large Language Models (Modèles de Langage de Grande Taille)

GPT : Generative Pre-trained Transformer

BERT : Bidirectional Encoder Representations from Transformers

WHO : World Health Organization (Organisation mondiale de la santé)

OMS : Organisation mondiale de la santé

ELIZA : Un chatbot conversationnel basé sur des règles , développé en 1966

Siri : Intelligent Personal Assistant développé par Apple

Alexa : Assistant vocal intelligent développé par Amazon

BioGPT : Generative Pre-trained Transformer spécialisé pour la littérature biomédicale

ClinicalBERT : Modèle BERT conçu pour les dossiers médicaux électroniques

CPLLM : Clinical Prediction with Large Language Models

MedQA : Medical Question Answering, une tâche d'évaluation pour les modèles dans le domaine médical

USMLE : United States Medical Licensing Examination

RGPD : Règlement Général sur la Protection des Données

NLP : Natural Language Processing (Traitement du Langage Naturel)

NLLB-200 : No Language Left Behind 200 (modèle de traduction multilingue de Meta)

GPU : Graphics Processing Unit

LLaMA : Large Language Model Meta AI

LoRA : Low-Rank Adaptation

BLEU : Bilingual Evaluation Understudy (métrique pour évaluer la qualité des traductions automatiques)

WER : Word Error Rate (Taux d'Erreur des Mots)

TTS : Text-to-Speech (Synthèse Vocale)

AED : Analyse Exploratoire des Données

IDE : Integrated Development Environment (Environnement de Développement Intégré)

API : Application Programming Interface

REST : Representational State Transfer

LISTE DES FIGURES

LISTE DES TABLEAUX

RÉSUMÉ

L'évolution rapide de l'intelligence artificielle (IA) ouvre la voie à des applications transformatrices dans le domaine de la santé. Parmi ces avancées, les agents conversationnels, en particulier les chatbots médicaux alimentés par les modèles de langage (LLMs), émergent comme des outils innovants pour améliorer l'accès aux soins.

Ce travail présente **Medbot**, un chatbot médical multilingue conçu pour améliorer l'accès aux soins. En combinant des modèles de langage avancés avec des connaissances médicales issues de ressources scientifiques reconnues, Medbot offre une assistance préliminaire via une interface intuitive, permettant aux utilisateurs de décrire leurs symptômes par texte, audio ou images. Le système fournit des recommandations personnalisées et, si nécessaire, oriente les utilisateurs vers des structures de santé appropriées.

Notre analyse approfondie des solutions existantes révèle des opportunités d'innovation dans les chatbots médicaux, particulièrement concernant les barrières linguistiques et l'accès aux informations médicales. Les résultats démontrent le potentiel de Medbot pour améliorer l'accessibilité aux soins de santé, particulièrement dans les zones mal desservies, tout en optimisant les ressources médicales. Ce travail contribue à la démocratisation des soins de santé en proposant un outil de pré-diagnostic fiable et accessible.

Mots-clés : Chatbot médical, Accessibilité aux soins, Modèles de langage, IA multilingue, Systèmes de pré-diagnostic, Santé numérique .

ABSTRACT

The rapid evolution of artificial intelligence (AI) has paved the way for transformative applications in healthcare. Among these advancements, conversational agents, particularly medical chatbots powered by large language models (LLMs), have emerged as innovative tools for improving access to healthcare.

This work presents **Medbot**, a multilingual medical chatbot designed to improve healthcare accessibility. By combining advanced language models with medical knowledge from recognized scientific resources, Medbot provides preliminary assistance through an intuitive interface, allowing users to describe their symptoms via text, audio, or images. The system provides initial recommendations and, when necessary, directs users to appropriate healthcare facilities.

Our comprehensive analysis of existing solutions reveals innovation opportunities in medical chatbots, particularly regarding language barriers and access to medical information. Results demonstrate Medbot's potential to enhance healthcare accessibility, particularly in underserved areas, while optimizing medical resources. This work contributes to healthcare democratization by providing a reliable and accessible guidance tool.

Keywords: Medical chatbot, Healthcare accessibility, Large Language models, Multilingual AI, Healthcare guidance systems, Digital health .

SOMMAIRE

INTRODUCTION GENERALE

1. Contexte et justification

Depuis sa naissance dans les années 1950 [1] , l'intelligence artificielle (IA) n'a cessé de se développer pour repousser les limites de ce que la technologie peut accomplir. Des premiers algorithmes de reconnaissance de motifs aux modèles de langage avancés (LLMs), tels que GPT [2], BERT [3] et leurs successeurs, l'IA a transformé la manière dont nous interagissons avec les machines. Ces avancées technologiques permettent aujourd'hui à des systèmes d'intelligence artificielle de traiter et d'interpréter des données complexes avec une précision et une rapidité inégalée.

Dans le domaine médical, les applications de l'IA et des LLMs sont particulièrement prometteuses [4]. De l'analyse d'imagerie médicale au soutien à la prise de décision clinique, ces technologies apportent des solutions innovantes pour améliorer la qualité des soins [5]. Les LLMs, grâce à leur capacité à comprendre et à générer du texte de manière naturelle, ouvrent de nouvelles perspectives pour l'interaction avec les patients, permettant notamment de fournir des réponses aux préoccupations médicales et des conseils personnalisés [9].

Cependant, une partie significative de la population mondiale souffre encore de l'absence d'une prise en charge médicale rapide, comme le souligne l'Organisation mondiale de la santé (WHO) [6]. Cette situation s'explique principalement par le manque d'infrastructures médicales et de professionnels de santé qualifiés. Ces lacunes sont particulièrement prononcées dans les zones rurales et sous-desservies, où les délais d'attente et les distances à parcourir pour consulter un médecin aggravent considérablement les risques pour les patients. Selon l'OMS, des millions de vies pourraient être sauvées chaque année grâce à une détection précoce et une intervention rapide [7] .

Face à ces défis, l'IA pourrait jouer un rôle important en comblant le fossé entre les patients et les professionnels de santé [8] . Les agents conversationnels, en particulier, peuvent offrir un premier niveau de réponse efficace aux préoccupations des patients, fournir des conseils adaptés, et orienter si nécessaire vers les structures médicales appropriées. Cette approche s'inscrit dans une logique d'optimisation des ressources médicales et de réduction de la charge de travail des praticiens.

2. Objectifs de la recherche

Ce mémoire a pour objectif de concevoir et de développer un chatbot médical intelligent capable

de :

1. **Analyser les symptômes décrits par les patients** afin de fournir des recommandations initiales pertinentes.
2. **Fournir des conseils personnalisés**, adaptés aux préoccupations des utilisateurs.
3. **Orienter les patients vers des professionnels ou des structures de santé adaptées** en fonction de la gravité de leur situation.
4. Offrir une **interface multilingue** (français et anglais) accessible pour répondre aux besoins d'une large population.
5. Réduire la surcharge des systèmes de santé en filtrant les cas non urgents et en facilitant l'accès à des informations médicales fiables.

3. Structure du mémoire

Ce mémoire est structuré comme suit :

- ✓ **Chapitre I : État de l'art** – Une revue des avancées dans le domaine des modèles de langage et des systèmes de pré-diagnostic médical, accompagnée d'une analyse critique des solutions existantes.
- ✓ **Chapitre II : Analyse des besoins et méthodologie** – Une présentation des exigences fonctionnelles et non fonctionnelles du système, des sources de données utilisées, et des approches méthodologiques adoptées.
- ✓ **Chapitre III : Implémentation et résultats** – Une description détaillée de l'implémentation technique, suivie d'une évaluation des performances du système.
- ✓ **Conclusion et perspectives** – Une analyse des résultats obtenus, des limitations identifiées, et des perspectives pour les futures améliorations du système.

Conclusion

En somme, ce mémoire vise à répondre aux défis actuels dans le domaine de la santé grâce à un chatbot médical innovant et accessible. Les sections suivantes explorent les fondements théoriques, les choix méthodologiques, et les résultats obtenus, tout en mettant en lumière les contributions potentielles de ce projet à l'amélioration de l'accès aux soins de santé.

CHAPITRE I: ETAT DE L'ART

Introduction

L'émergence des chatbots, propulsée par les avancées en intelligence artificielle, a transformé la manière dont les humains interagissent avec les systèmes informatiques. Ces agents conversationnels, initialement conçus pour des interactions simples, ont évolué vers des systèmes sophistiqués capables de comprendre et de générer du langage naturel de manière contextuelle. Leur application dans le domaine médical représente l'une des évolutions les plus prometteuses, offrant de nouvelles perspectives pour l'accessibilité aux soins et l'optimisation des ressources médicales.

Ce chapitre présente une analyse approfondie de l'état de l'art des chatbots, en commençant par leur définition et leur historique, avant d'examiner leurs différentes typologies et architectures. Une attention particulière sera portée aux technologies sous-jacentes, notamment les modèles de langage récents qui révolutionnent ce domaine. Nous nous concentrerons ensuite sur leur application spécifique dans le secteur médical, examinant les enjeux actuels et les perspectives d'évolution de ces technologies, positionnant ainsi notre solution Medbot dans ce paysage en constante mutation.

1. Introduction aux chatbots médicaux et systèmes de pré-diagnostic

1.1 Définition et évolution des chatbots

Un chatbot est une application logicielle ou une interface web conçue pour mener des conversations textuelles ou vocales. Les chatbots modernes utilisent souvent des systèmes d'intelligence artificielle avancés, comme le traitement du langage naturel et l'apprentissage profond, pour générer des réponses interactives et contextuelles [11] .

L'évolution des chatbots peut être divisée en trois grandes générations :

a) **Première génération (1960-1990)** : Cette période marque l'avènement des systèmes conversationnels basés sur des règles strictes. Par exemple, ELIZA, développé en 1966, utilisait des modèles simples de correspondance de mots pour simuler une interaction humaine. Bien qu'innovante à son époque, cette approche manquait de compréhension contextuelle et se limitait à des interactions superficielles.

b) **Deuxième génération (1990-2010)** : Avec l'introduction de l'apprentissage automatique, les chatbots ont gagné en complexité. Ils pouvaient analyser le langage naturel de manière

plus sophistiquée grâce aux systèmes experts. Ces modèles ont permis une interaction plus fluide, mais restaient limités par la nécessité d'un entraînement manuel intensif.

c) **Troisième génération (2010-présent)** : L'apparition des modèles de langage avancés (LLMs) comme GPT, Claude, et Llama a transformé le domaine. Ces modèles, alimentés par des algorithmes d'apprentissage profond, offrent des interactions riches et adaptatives, tout en comprenant mieux le contexte des préoccupations médicales des utilisateurs.

Ainsi, les chatbots évoluent continuellement pour s'adapter aux besoins croissants dans divers domaines.La figure ci-dessous résume l'évolution des chatbots depuis 1950 jusqu'à nos jours.

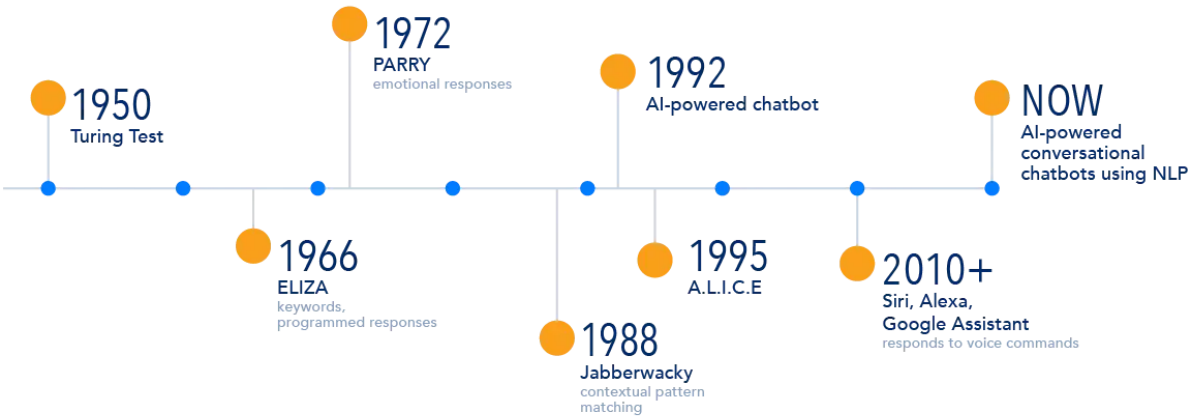


Figure 1: Evolution des chatbots [20]

1.2 Types de chatbots

Dans cette sous-section, nous introduisons quelques types de chatbots, différenciés par leur méthode de traitement.

A. Chatbots Basés sur des Menus/Boutons

Les chatbots basés sur des menus ou des boutons sont les plus simples et les plus basiques existants aujourd'hui. Ils fonctionnent en hiérarchies d'arborescences décisionnelles présentées à l'utilisateur sous forme de boutons. Ces chatbots rappellent les menus automatisés téléphoniques que nous utilisons quotidiennement, où les utilisateurs doivent prendre des décisions pour obtenir des

réponses précises. L'utilisateur est guidé à travers ces choix pour arriver à une réponse appropriée générée par l'IA. Cependant, ces chatbots sont souvent les plus lents pour atteindre une réponse désirée et sont inefficaces dans des scénarios complexes où il existe trop de variables ou de connaissances à traiter. Ils deviennent peu adaptés lorsque les utilisateurs ont besoin de réponses spécifiques dans des situations avancées [13] .

B. Chatbots Basés sur la Reconnaissance de Mots-Clés

Contrairement aux chatbots basés sur des menus, ceux-ci reconnaissent des mots-clés spécifiques pour produire un résultat souhaité. Ces chatbots analysent ce que les utilisateurs saisissent et y répondent de manière appropriée en utilisant une liste de mots-clés personnalisée et une application d'IA. Cependant, ces chatbots rencontrent des problèmes lorsque plusieurs questions similaires entraînent des redondances dans les mots-clés. Il est courant de trouver des chatbots hybrides qui combinent reconnaissance de mots-clés et menus. Ces chatbots offrent aux utilisateurs la possibilité de poser directement leurs questions ou d'utiliser les boutons de menu si la reconnaissance des mots-clés ne produit pas de résultats satisfaisants ou si l'utilisateur a besoin d'aide pour trouver une réponse [13] .

C. Chatbots Basés sur des Règles (Chatbots Linguistiques)

Les chatbots linguistiques ou basés sur des règles utilisent une logique conditionnelle de type "si/alors". Ces systèmes reposent sur des conditions linguistiques définies à l'avance, permettant d'évaluer les mots, leur ordre, les synonymes, etc. Si l'entrée de l'utilisateur correspond aux conditions définies, le chatbot peut fournir une aide appropriée rapidement. Cependant, toutes les permutations et combinaisons de chaque question doivent être définies, faute de quoi le chatbot ne comprendra pas les requêtes de l'utilisateur. Ce type de chatbot, bien que très répandu, est souvent lent à développer et manque de flexibilité. Il nécessite des mises à jour fréquentes pour rester pertinent [14] .

D. Chatbots Contextuels (Basés sur l'Apprentissage Automatique)

Les chatbots contextuels, également appelés chatbots basés sur l'apprentissage automatique, sont les plus avancés parmi les types mentionnés. Ils utilisent des technologies comme le traitement du langage naturel, la reconnaissance vocale, la conversion parole-texte, la traduction automatique et d'autres algorithmes pour analyser les intentions de l'utilisateur et fournir des réponses adaptées. Ces chatbots mémorisent l'état des conversations avec des utilisateurs spécifiques pour apprendre et s'améliorer au fil du temps. L'idée principale est de détecter les intentions des utilisateurs et de fournir des réponses raisonnées en analysant les modèles dans une base de données [13] .

E. Chatbots Vocaux

Les chatbots vocaux rendent les interfaces conversationnelles plus naturelles et plus faciles à utiliser. Ils ont gagné en popularité ces dernières années, notamment avec des applications comme Siri (Apple), Alexa (Amazon) et Google Assistant. Ces systèmes permettent aux utilisateurs de parler directement au lieu de taper, offrant une expérience plus agréable et intuitive [15] .L'image suivante montre les types de chatbots qui existent.

Figure 2 : Types de chatbots [21]

1.3 Architecture des Chatbots

La conception et le développement de systèmes de chatbots basés sur l'IA nécessitent une variété de techniques de prétraitement, telles que le traitement du langage naturel ou la génération de langage [16] . Pour construire des chatbots précis et pertinents, les développeurs doivent comprendre les fonctionnalités offertes par le chatbot et la catégorie à laquelle il appartient pour choisir les bons algorithmes, plateformes et outils. Cette approche aide également les utilisateurs finaux à mieux comprendre ce qu'ils peuvent attendre du chatbot [17] .

La première étape dans la conception de tout système de chatbot consiste à diviser le système en parties selon une norme, ce qui rend le développement plus simple et modulaire [18] .La figure qui suit montre l'architecture générale d'un chatbot.

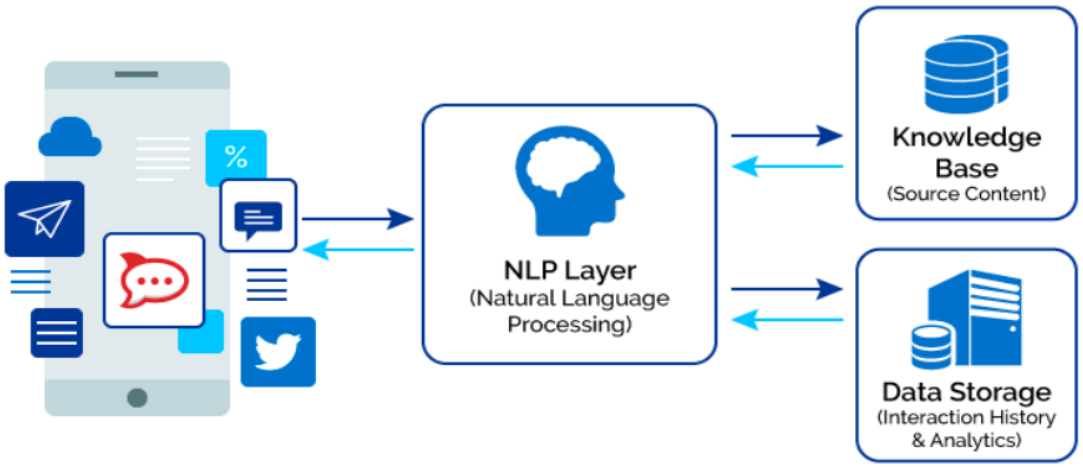


Figure 3 : Architecture générale d'un chatbot [19]

1.4 Importance des systèmes de pré-diagnostic

Les systèmes de pré-diagnostic occupent une place cruciale dans le paysage médical actuel. En fournissant un premier niveau d'analyse des symptômes, ces systèmes aident les patients à mieux comprendre leur état de santé avant de consulter un professionnel. Leur importance peut être examinée sous plusieurs angles :

1. **Amélioration de l'accès aux soins** : Dans des régions où les infrastructures médicales sont insuffisantes, ces outils permettent de compenser le manque de ressources humaines. Un patient vivant dans une zone rurale peut, grâce à un chatbot, recevoir une première évaluation de ses symptômes sans avoir à parcourir de longues distances.
2. **Optimisation des ressources médicales** : En filtrant les cas non urgents, les systèmes de pré-diagnostic réduisent la charge pesant sur les professionnels de santé. Cela leur permet de consacrer plus de temps aux cas critiques, améliorant ainsi l'efficacité globale des soins.
3. **Réassurance pour les utilisateurs** : En fournissant des réponses rapides et adaptées, ces systèmes aident à réduire l'anxiété des patients face à des symptômes inconnus. Cela favorise une prise de décision plus éclairée et proactive.

Malgré ces avantages, les systèmes actuels présentent des limites, notamment en termes de précision et d'accessibilité multilingue, que des solutions comme Medbot cherchent à résoudre.

1.5 Enjeux de l'accessibilité aux informations médicales

L'accès aux informations médicales fiables reste un défi majeur pour une grande partie de la

population mondiale. Ces obstacles sont exacerbés par plusieurs facteurs :

1. **Prolifération de l'information en ligne** : Avec l'émergence de plateformes comme Google ou Bing, les patients disposent d'un accès quasi illimité à des informations médicales. Cependant, cette abondance d'informations pose des problèmes de tri et de vérification, car une grande partie des contenus n'est pas validée par des professionnels. *Exemple* : Lorsqu'un utilisateur recherche des solutions pour des maux de tête, il peut trouver des suggestions allant de simples remèdes maison à des diagnostics alarmants comme des tumeurs cérébrales, sans hiérarchie de fiabilité.
2. **Barrières linguistiques** : La majorité des ressources médicales en ligne sont en anglais, ce qui exclut les locuteurs de langues minoritaires. Cela aggrave les inégalités d'accès aux soins, en particulier dans les pays en développement.
3. **Complexité des termes médicaux** : Les informations médicales sont souvent rédigées dans un langage technique inaccessible au grand public. Par exemple, un patient lambda pourrait ne pas comprendre des termes comme "angine de poitrine" ou "ischémie".

Pour surmonter ces défis, un chatbot comme Medbot vise à rendre les informations accessibles dans des langues locales, tout en valorisant les contenus médicaux pour les rendre compréhensibles par tous.

2. État de l'art sur les modèles de langage en santé

2.1 Modèles généralistes récents : ChatGPT, Claude, Llama et Gemini

Les modèles de langage généralistes, comme ChatGPT et Claude, ont transformé la manière dont les utilisateurs interagissent avec les systèmes d'IA. Ces modèles se distinguent par leur capacité à comprendre et générer du texte de manière fluide et contextuelle.

1. **Avantages des modèles généralistes** :
    - *Polyvalence* : Ces modèles peuvent répondre à des requêtes variées, allant de questions techniques à des préoccupations médicales générales.
    - *Interface conviviale* : Leur accessibilité via des plateformes comme des applications mobiles ou des sites web les rend populaires auprès du grand public.
2. **Limites des modèles généralistes** :
    - *Manque de spécialisation* : Ces modèles, bien que efficaces dans des contextes généralistes, manquent de précision lorsqu'ils traitent des données médicales

complexes.

- *Absence de validation médicale* : Les réponses ne sont pas toujours fiables, ce qui pose un risque pour les utilisateurs cherchant des conseils critiques.

2.2 Modèles spécialisés en biomédical : BioGPT, ClinicalBERT, et CPLLM

Contrairement aux modèles généralistes, les modèles spécialisés en biomédical ont été conçus pour traiter des données spécifiques au domaine médical.

- ✓ **BioGPT** : Ce modèle excelle dans l'analyse de la littérature biomédicale, en extrayant des informations pertinentes à partir de bases de données comme PubMed.
- ✓ **ClinicalBERT** : Conçu pour les dossiers médicaux électroniques, il est utile pour analyser et structurer des informations cliniques complexes.
- ✓ **CPLLM (Clinical Prediction with Large Language Models)** : Ce modèle prédictif utilise des données historiques pour anticiper les diagnostics futurs, surpassant des approches traditionnelles comme les régressions logistiques.

Cependant, ces modèles restent confinés au milieu académique ou aux environnements de recherche, avec une accessibilité limitée pour le grand public.La figure suivante resume l'évolution des différents modèles de langage en santé.

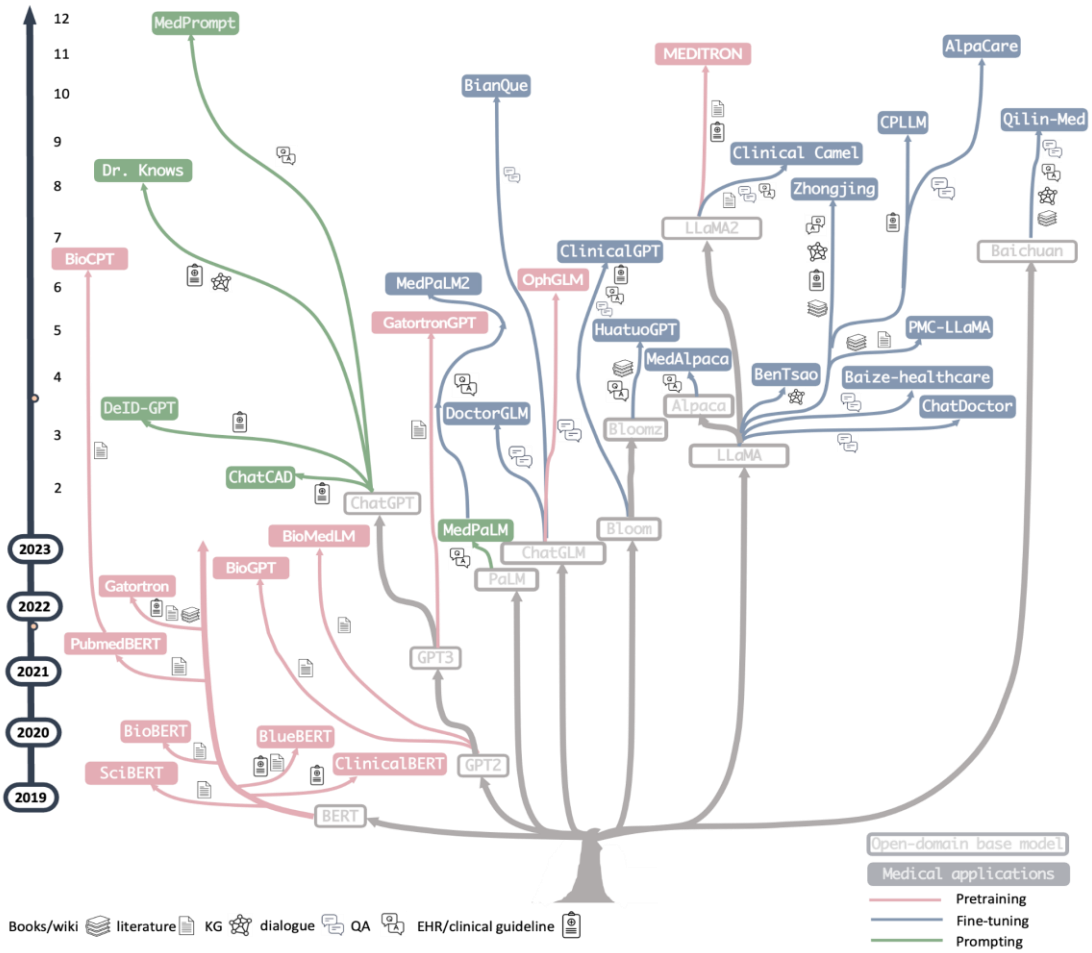


Figure 4 : Évolution des modèles de langage en santé et leurs applications [22]

2.3 Analyse comparative des approches existantes

Le tableau ci-dessous faire une analyse comparative entre les modèles généralistes et spécialisés dans le domaine médical.

Critères	**Modèles généralistes**	**Modèles spécialisés**
Flexibilité	Adaptés à de multiples domaines.	Spécifiques au domaine médical.
Performance médicale	Faible précision sans fine-tuning.	Haute précision sans adaptation supplémentaire.
Accessibilité	Interfaces simples et intuitives.	Complexes, souvent réservées aux professionnels.
Multilinguisme	Multilingue.	Majoritairement en anglais.

Tableau 1: Analyse comparative des approches existantes

La figure ci-dessous illustre les performances des différents modèles d'IA en santé, mesurées sur la tâche MedQA (USMLE-style accuracy).

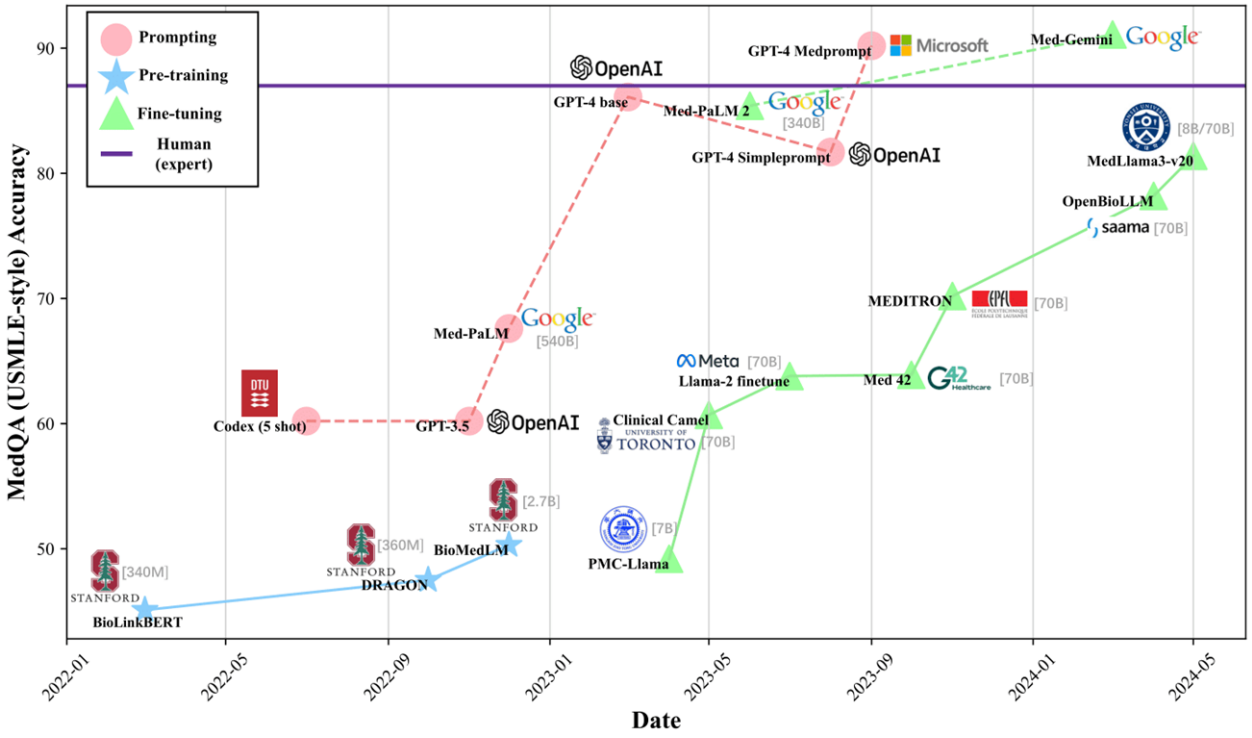


Figure 5: Performances des modèles de langage dans le domaine médical (MedQA) [22]

3. Positionnement de Medbot

Medbot vise à combiner les avantages des modèles généralistes et spécialisés en proposant une solution :

- **Multilingue et accessible** : Disponible en français et anglais, avec une interface intuitive.
- **Fiable** : Validé médicalement pour garantir la précision des recommandations.
- **Personnalisé** : Capable de contextualiser les réponses selon les symptômes et les besoins des utilisateurs.

Conclusion

Ce chapitre a mis en lumière les avancées réalisées dans les chatbots médicaux et les modèles de langage, tout en identifiant les lacunes des solutions actuelles. Medbot se positionne comme une solution innovante, combinant accessibilité, fiabilité, et précision, pour répondre aux besoins des systèmes de santé modernes.La suite de ce mémoire explorera la méthodologie adoptée pour développer cette solution.

CHAPITRE II: ANALYSE DE BESOINS ET METHODOLOGIE

L'élaboration d'un chatbot médical nécessite une analyse approfondie des besoins et une méthodologie rigoureuse pour garantir sa fiabilité et son efficacité. Ce chapitre présente l'ensemble des étapes de conception de Medbot, depuis l'identification des besoins jusqu'aux protocoles d'évaluation. Notre approche méthodologique vise à créer un système robuste capable de répondre aux exigences spécifiques du domaine médical tout en assurant une expérience utilisateur optimale.

I. Spécification des besoins et des données

1.1 Analyse et spécification des besoins

Le développement de **Medbot** repose sur des besoins clairement définis, tant sur le plan fonctionnel que non fonctionnel, afin de répondre aux exigences du domaine médical et d'assurer une expérience utilisateur optimale.

a. Besoins fonctionnels

L'un des principaux objectifs de Medbot est d'offrir une interaction multimodale adaptée aux divers besoins des utilisateurs. À cette fin, le système intègre trois modalités principales d'interaction. Tout d'abord, une **interface textuelle** permet aux patients de décrire leurs symptômes, leurs antécédents médicaux ou leurs préoccupations de manière détaillée. Cette interface textuelle constitue la base des interactions classiques, en particulier pour les utilisateurs qui préfèrent une communication écrite. Ensuite, le système intègre un **module de reconnaissance vocale**, qui offre une interaction naturelle et fluide, en convertissant les entrées audios en texte. Cette fonctionnalité améliore l'accessibilité pour les utilisateurs qui ne souhaitent pas écrire ou qui se sentent plus à l'aise avec la parole. Enfin, une **capacité d'analyse d'images médicales** est incluse, permettant aux patients de télécharger des photos de symptômes visibles, comme des éruptions cutanées ou des blessures, pour une analyse automatique.

Le cœur du système repose sur une **analyse intelligente des symptômes**, qui s'appuie sur des algorithmes avancés pour comprendre le contexte des descriptions fournies par les patients. Cette compréhension contextuelle est essentielle pour générer des recommandations pertinentes, basées sur des sources médicales validées et scientifiquement rigoureuses. De plus, Medbot propose un **suivi personnalisé** de l'évolution des symptômes, permettant aux utilisateurs de recevoir des conseils actualisés en fonction des changements rapportés.

En outre, le système garantit un **support multilingue**, avec une prise en charge initiale du français

et de l'anglais, afin de répondre aux besoins d'un public diversifié. Pour améliorer l'accessibilité, Medbot inclut également une **synthèse vocale**, qui permet de fournir des réponses sous forme audio. Cette fonctionnalité est particulièrement utile pour les personnes ayant des difficultés de lecture ou préférant une interaction auditive. Enfin, l'interface est conçue pour être intuitive et adaptée à différents niveaux de littératie numérique, afin de garantir une utilisation facile, même pour les utilisateurs non spécialisés.

b. Besoins non fonctionnels

La **fiabilité** et la **sécurité** sont des priorités absolues dans le développement de Medbot, compte tenu de la sensibilité des données médicales et de la nécessité de fournir des informations précises. La précision des réponses médicales est assurée par l'intégration de sources validées scientifiquement, garantissant ainsi la pertinence des recommandations fournies. En parallèle, la **protection des données personnelles** est strictement conforme aux réglementations, notamment le RGPD, afin de garantir la confidentialité et la sécurité des informations des utilisateurs.

La **performance** du système est également un aspect critique, Medbot étant conçu pour être optimisé afin de gérer un grand nombre d'interactions simultanées. Cela garantit une expérience utilisateur fluide, même en cas de forte demande. Enfin, l'interface de Medbot est pensée pour être accessible et intuitive, rendant l'utilisation du système simple et efficace pour tous, indépendamment du niveau de compétence technique ou médicale des utilisateurs.

Ainsi, Medbot s'inscrit dans une démarche alliant innovation technologique et accessibilité, répondant aux besoins spécifiques du domaine médical tout en assurant une expérience utilisateur fiable et sécurisée.

1.2 Sources de données et structure

Le succès de Medbot repose sur des données fiables, variées et validées pour garantir des recommandations médicales précises et contextualisées. Trois principales sources de données alimentent le système, chacune jouant un rôle essentiel dans la formation et l'amélioration continue du modèle :

a) Medical Meadow Wikidoc

Cette base de données est issue de *Wikidoc* [24], une plateforme collaborative dédiée aux professionnels de santé pour partager des connaissances médicales actualisées. Les données sont structurées en paires

question-réponse, ce qui permet de modéliser des scénarios médicaux réalistes.

- **Input** : Questions ou descriptions de situations médicales (ex. : "Quels sont les symptômes d'une angine bactérienne ?").
- **Output** : Explications scientifiques validées, extraites d'un "textbook médical vivant".

Les questions sont générées ou reformulées automatiquement à partir des titres des sections via des modèles NLP avancés comme GPT-3.5-Turbo. Cela permet de couvrir un large éventail de spécialités médicales.

b) Medical Meadow MedQA

Le dataset *MedQA* [23] est une collection de questions-réponses tirées des examens des conseils médicaux professionnels. Il se distingue par sa couverture multilingue : anglais, chinois simplifié et chinois traditionnel. Chaque exemple contient :

- **Input** : Une question issue d'un examen médical (ex. : "Quel est le traitement initial pour une crise cardiaque ?").
- **Output** : Une réponse validée par des experts médicaux, souvent sous forme de choix multiples ou de réponses détaillées.

Les données sont organisées en trois ensembles géographiques (États-Unis, Chine continentale, Taïwan), et chaque langue bénéficie d'une division claire en ensembles d'entraînement, de validation et de test. Ce dataset met également en évidence les limites des systèmes OpenQA, même pour les modèles les plus avancés, en raison de la complexité des questions posées.

c) Medical Meadow Health Advice

Ce dataset se concentre sur les conseils de santé et les recommandations médicales basées sur des descriptions de symptômes [25] .

- **Input** : Descriptions de symptômes ou situations médicales fournies par les utilisateurs.
- **Output** : Conseils pratiques et recommandations validées (ex. : "Pour une fièvre modérée, hydratez-vous et prenez du paracétamol selon la posologie indiquée").

Ce dataset est aligné avec les besoins des patients cherchant des conseils immédiats et pratiques, et il s'intègre parfaitement aux scénarios de pré-diagnostic.

1.3 Structure des conversations

Pour garantir une standardisation optimale, toutes les interactions entre l'utilisateur et le chatbot suivent un format uniforme basé sur des rôles bien définis :

- **User** : Correspond à l'entrée utilisateur (texte, audio ou image décrivant les symptômes).
- **Assistant** : Réponse générée par le modèle, incluant les recommandations ou orientations.

Exemple de la structure en Python :

```
conversations = [
    {"role": "user", "content": example["input"]},
    {"role": "assistant", "content": example["output"]}
]
```

Figure 6: Script python sur la structure des conversations

La structure des interactions entre l'utilisateur et Medbot est soigneusement conçue pour garantir cohérence et efficacité. Chaque échange distingue clairement les rôles, avec l'utilisateur fournissant des symptômes ou des questions, et l'assistant générant des réponses adaptées. Cette standardisation améliore la qualité des données d'entraînement et s'intègre parfaitement aux frameworks modernes comme Hugging Face, simplifiant le prétraitement et optimisant les performances des modèles.

En séparant explicitement les rôles, le système réduit les confusions et renforce la capacité du modèle à générer des recommandations précises. De plus, cette structure flexible permet d'intégrer des données multimodales comme des images ou des audio, rendant Medbot adaptable à des scénarios plus complexes. Elle constitue ainsi une base solide pour un apprentissage robuste et une évolution future du système, répondant aux besoins actuels et émergents en santé.

1.3.Répartition des données

Pour une évaluation rigoureuse des performances du modèle, les données sont divisées en :

- **Ensemble d'entraînement** (80%) : Utilisé pour ajuster les poids des modèles.
- **Ensemble de validation** (20%) : Permet d'évaluer la capacité du modèle à généraliser sur des données inconnues.

Cette répartition suit les meilleures pratiques en machine learning, assurant un équilibre entre apprentissage et validation [27].

1.4 Prétraitement des données

Le prétraitement des données constitue une étape fondamentale dans le développement de tout modèle d'intelligence artificielle.Elle permet de garantir la qualité, la cohérence et l'exploitabilité des données pour maximiser les performances du modèle. Dans notre cas, les techniques employées vont du nettoyage à la tokenisation en passant par l'utilisation de modèles spécialisés pour la traduction et l'adaptation multilingue. Cette section détaille chaque étape clé du prétraitement.

a. Nettoyage et normalisation des données

Les données initiales issues de sources variées (MedQA, WikiDoc, Health Advice) nécessitent un nettoyage rigoureux pour éliminer les incohérences et garantir leur qualité. Les étapes spécifiques incluent :

- Élimination des doublons : Les entrées répétées sont identifiées et supprimées pour éviter les biais pendant l'entraînement.
- Correction des anomalies syntaxiques : Les fautes d'orthographe et les erreurs grammaticales sont corrigées.
- Traitement des données manquantes : Les valeurs absentes sont imputées ou exclues, selon leur impact potentiel sur les performances du modèle.

b. Standardisation des données conversationnelles

Pour garantir une cohérence dans les interactions, les données sont formatées selon une structure conversationnelle standardisée. Cette structure distingue clairement les rôles (utilisateur et assistant) et facilite le traitement par les modèles LLM. Voici un exemple de ce format standard :

```
conversations = [
    {"role": "user", "content": example["input"]},
    {"role": "assistant", "content": example["output"]}
]
```

Figure 7: Formatage des conversations

Cette normalisation assure une séparation claire des questions et des réponses une meilleure compatibilité avec les modèles de langage.

c. Traduction et adaptation multilingue

Pour répondre aux besoins d'un public francophone et anglophone, notre chatbot intègre un modèle de traduction multilingue avancé, **facebook/nllb-200-distilled-600M**. NLLB-200 est un modèle de traduction automatique destiné principalement à la recherche en traduction automatique, en particulier pour les langues à faibles ressources. Il permet de traduire une seule phrase parmi 200 langues [30] . Ce modèle permettra une traduction fluide et contextuelle des interactions. Aussi, une standardisation linguistique des termes médicaux pour garantir la précision des réponses.

d. Tokenisation et segmentation

Les données sont tokenisées pour être exploitées efficacement par les modèles. La tokenisation consiste à découper les entrées textuelles en unités plus petites (tokens), permettant au modèle de traiter chaque élément séparément. Une longueur maximale de **2048** tokens est fixée pour optimiser l'utilisation de la mémoire GPU et garantir la compatibilité avec les architectures des modèles.

e. Fusion et division des datasets

Les données issues des sources **MedQA**, **WikiDoc**, et **Health Advice** sont fusionnées pour former un dataset unique. Une division stratégique en ensembles d'entraînement (80%) et de validation (20%) est effectuée pour garantir une évaluation fiable des performances.

Code correspondant :

```
dataset_split = dataset.train_test_split(test_size=0.2, seed=42)
train_dataset = dataset_split["train"]
eval_dataset = dataset_split["test"]
```

Figure 8: Script de séparation des données

Bien que nous utilisions un modèle pré-entraîné, plusieurs étapes de prétraitement s'avèrent nécessaires pour optimiser les performances du système. Le processus de prétraitement comprend trois phases essentielles.

La première phase concerne le nettoyage et la normalisation des données. Toutes les entrées textuelles et audio sont standardisées pour éliminer les incohérences potentielles qui pourraient affecter la qualité des analyses. La deuxième phase traite l'aspect multilingue du système, s'appuyant sur le modèle facebook/nllb-200-distilled-600 [31] pour assurer une traduction précise

entre le français et l'anglais. La dernière phase implique la tokenisation et l'annotation des données, transformant les entrées utilisateur en format exploitable par les modèles d'analyse.

La Figure ci-dessous présente le processus complet de prétraitement des données, illustrant les différentes étapes de transformation des entrées utilisateur.

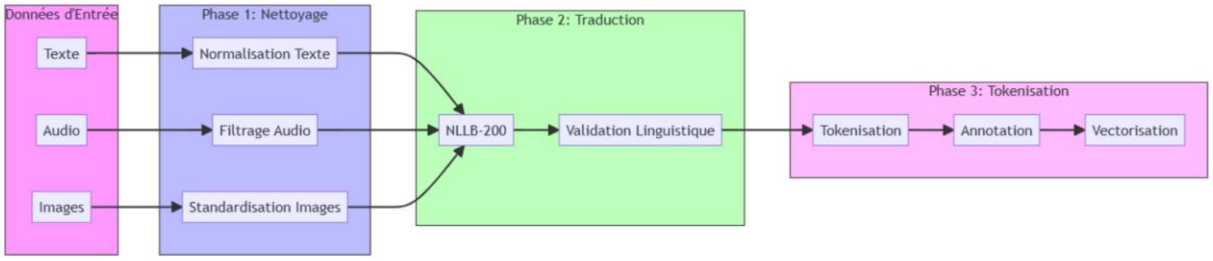


Figure 9: Processus de prétraitement des données complet de Medbot

II. Méthodologie

1. Choix des Algorithmes et Modèles

L'approche méthodologique repose sur l'utilisation de plusieurs composants clés, chacun répondant à des besoins spécifiques :

a. Modèle principal : LLaMA 3.2-3B-Instruct

Le modèle Llama-3.2-3B-Instruct, basé sur la famille LLaMA (Large Language Model Meta AI), a été choisi pour son efficacité dans les tâches de compréhension du langage naturel et son optimisation pour des contextes conversationnels complexes.

Depuis sa première version publiée par Meta AI en février 2023, LLaMA a évolué pour inclure des modèles de différentes tailles (de 1B à 405B paramètres). À partir de LLaMA 2, Meta AI a introduit des versions spécialement affinées pour des tâches d'instruction. Cependant, la version utilisée ici, publiée par Unsloth, une organisation spécialisée dans l'optimisation des grands modèles de langage (unsloth/Llama-3.2-3B-Instruct), apporte des optimisations supplémentaires, telles que la quantification en 4 bits, afin de réduire la consommation de mémoire et d'accélérer les performances d'inférence.

La dernière version officielle de la famille LLaMA, LLaMA 3.3, a été publiée en décembre 2024.L'image qui suit représente l'architecture du modèle Llama 3.

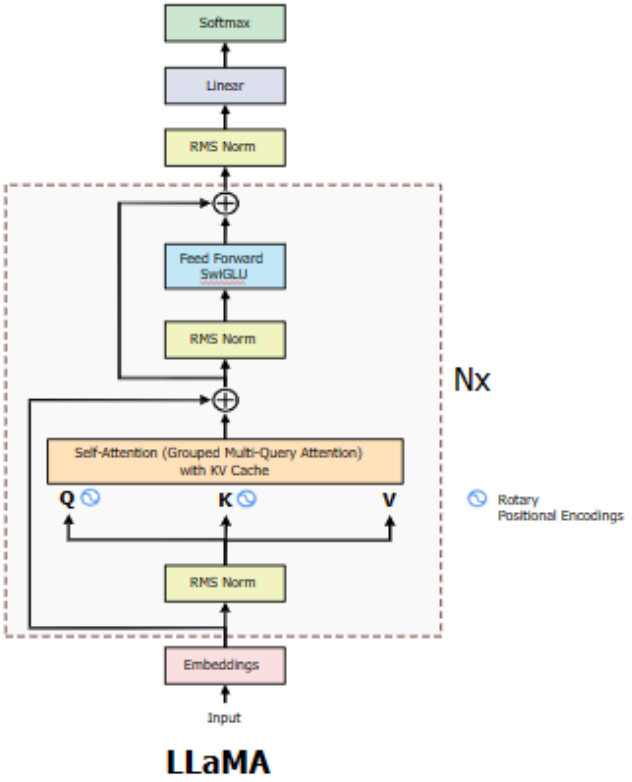


Figure 10: Architecture du Llama 3 [33]

Le Llama-3.2-3B-Instruct se distingue par :

- Une gestion fluide des dialogues longs grâce à une longueur maximale de séquences de 2048 tokens, avec support du RoPE Scaling intégré pour des séquences étendues.
- Des améliorations significatives en précision contextuelle, essentielles pour des cas d'usage tels que le domaine médical.
- Une optimisation via la quantification en 4 bits, permettant de réduire l'empreinte mémoire, tout en maintenant une haute performance, idéale pour des déploiements sur des GPU limités ou dans des environnements contraints.
- Une flexibilité pour la compréhension et la génération de langage naturel, parfaitement adaptée aux contextes conversationnels complexes.

b. Fine-Tuning avec LoRA

La méthode LoRA (Low-Rank Adaptation) permet de spécialiser le modèle pour des tâches médicales sans nécessiter un fine-tuning complet. Contrairement aux méthodes classiques, LoRA

:

- Modifie uniquement certaines couches du modèle, réduisant ainsi les ressources nécessaires.
- Facilite l'adaptation rapide à de nouveaux domaines, comme les recommandations médicales.

Configuration LoRA utilisée :

```
model = FastLanguageModel.get_peft_model(
    model,
    r=8,
    target_modules=["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj"],
    lora_alpha=16,
    lora_dropout=0.0
)
```

Figure 11: Configuration Lora

a. Quantification en 4 bits

La quantification réduit la précision numérique des poids du modèle, passant de 32 bits à 4 bits, tout en préservant la qualité des prédictions. Les avantages comprennent :

- Réduction de la mémoire requise, rendant le modèle accessible sur des GPU avec des ressources limitées.
- Amélioration des performances en termes de vitesse d'inférence.

```
load_in_4bit = True
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name="unsloth/Llama-3.2-3B-Instruct",
    max_seq_length=2048,
    load_in_4bit=load_in_4bit
)
```

Figure 12: Script quantification

b. Reconnaissance vocale et synthèse vocale

- *Whisper (openai/whisper-small)* : Utilisé pour convertir des entrées audio en texte avec une haute précision. Cela garantit une interaction fluide pour les utilisateurs préférant parler plutôt qu'écrire.
- *MMS-TTS (facebook/mms-tts-fra)* : Génère des réponses vocales en français avec une qualité naturelle, rendant le système accessible à un public plus large.

c. Traduction multilingue

Le modèle *facebook/nllb-200-distilled-600M* est intégré pour traduire les entrées et sorties entre le français et l'anglais, avec un accent particulier sur la standardisation des termes médicaux.

Justification des Choix

Les modèles et algorithmes sélectionnés répondent spécifiquement aux besoins identifiés pour Medbot :

- *Modèle principal LLaMA* : Sa capacité à traiter des séquences longues et des dialogues complexes en fait un choix idéal pour les scénarios de pré-diagnostic médical.
- *LoRA* : permet une adaptation efficace aux tâches médicales spécifiques tout en minimisant les ressources nécessaires. Cette approche modifie uniquement certaines couches du modèle, réduisant ainsi considérablement les coûts d'entraînement tout en maintenant la qualité des prédictions.Cette méthode est économique et flexible, adaptée à l'adaptation rapide de Medbot pour des cas médicaux variés.
- *Quantification en 4 bits* : Cette méthode, tout en préservant la qualité des prédictions, permet de réduire significativement l'empreinte mémoire du modèle et d'améliorer les temps d'inférence. Cette optimisation est particulièrement importante pour garantir des performances fluides dans des environnements aux ressources limitées.Elle permet de déployer Medbot sur des infrastructures légères, rendant l'application accessible même dans des environnements à ressources limitées.
- La reconnaissance vocale est assurée par le modèle *Whisper (openai/whisper-small)*, choisi pour sa précision dans la conversion des entrées audio en texte
- La synthèse vocale utilise le modèle *MMS-TTS (facebook/mms-tts-fra)* pour générer des réponses audios naturelles en français
- La traduction multilingue s'appuie sur le modèle *NLLB-200 (facebook/nllb-200-distilled-600M)*, garantissant une communication précise dans les différentes langues supportées.

2. Configuration et paramètres d'optimisation

Les hyperparamètres ont été optimisés pour maximiser les performances :

- ✓ La longueur maximale des séquences est fixée à 2048 tokens
- ✓ La taille du batch est de 2, avec une accumulation de gradient sur 4 steps
- ✓ Le taux d'apprentissage est maintenu à 1e-5
- ✓ L'évaluation est effectuée tous les 50 steps, avec une sauvegarde des checkpoints tous les 100 steps

L'infrastructure technique s'appuie sur Google Colab avec des GPU NVIDIA A100, permettant un entraînement et une validation efficaces du modèle.

3. Méthodologie d'évaluation

L'évaluation de Medbot suit une approche rigoureuse et multidimensionnelle visant à garantir la fiabilité et l'efficacité du système. Le processus d'évaluation repose sur des critères quantitatifs et qualitatifs soigneusement sélectionnés pour mesurer les différents aspects des performances du système.

a. Métriques et critères d'évaluation

La précision des transcriptions audio est évaluée à travers le Word Error Rate (WER), tandis que la qualité linguistique des réponses générées est mesurée par le BLEU Score. L'exactitude diagnostique, métrique importante pour un système médical, évalue la correspondance entre les recommandations du système et les diagnostics établis par des professionnels de santé. Les temps de réponse sont également mesurés pour garantir une expérience utilisateur fluide et réactive.

b. Protocole de validation

Le protocole de validation comprend trois phases distinctes et complémentaires :

La première phase utilise une validation croisée sur les jeux de données segmentés pour vérifier la robustesse du modèle dans différentes conditions. La deuxième phase implique une comparaison approfondie avec des solutions existantes reconnues, notamment Ada Health, permettant de situer les performances de Medbot dans le contexte actuel des systèmes de pré-diagnostic. La dernière phase consiste en des tests en conditions simulées, reproduisant des scénarios cliniques réalistes pour évaluer la capacité du système à gérer des situations complexes et variées.

Conclusion

Ce chapitre a présenté la méthodologie complète de développement de Medbot, depuis l'analyse initiale des besoins jusqu'aux protocoles d'évaluation rigoureux. L'utilisation combinée de modèles avancés, associée à une approche systématique de validation, pose les bases d'une solution innovante et fiable pour l'assistance médicale préliminaire. Les choix méthodologiques présentés reflètent un équilibre essentiel entre performance technique et accessibilité utilisateur. Le prochain chapitre détaillera la mise en œuvre technique de ces choix et présentera les résultats expérimentaux obtenus.

CHAPITRE III: IMPLEMENTATION

Ce chapitre présente le processus d'implémentation de **Medbot**, depuis le développement des modules jusqu'à leur intégration dans une solution fonctionnelle. Les résultats obtenus, ainsi que les outils et technologies utilisés, sont décrits en détail pour illustrer les performances et la méthodologie adoptée.

I. Résultats

1) Analyse exploratoire des données

L'analyse exploratoire des données (AED) a permis de comprendre la structure des données, de détecter des anomalies, et d'orienter le processus de préparation pour le fine-tuning du modèle. Les visualisations obtenues offrent des perspectives détaillées sur la composition des jeux de données utilisés.

a. Nuage de mots des jeux de données

Le nuage de mots dans la figure ci-dessous fournit une représentation visuelle des termes les plus fréquents dans chaque dataset (WikiDoc, MedQA et Health Advice). Il met en évidence des mots-clés dominants tels que *patient, disease, advice, treatment*, qui reflètent les thématiques principales abordées. Par exemple, *treatment* et *disease* dominent dans WikiDoc, soulignant son orientation vers des descriptions de maladies et traitements spécifiques. Dans MedQA, les termes tels que *cause* et *following* mettent en évidence des scénarios cliniques ou des explications médicales détaillées.

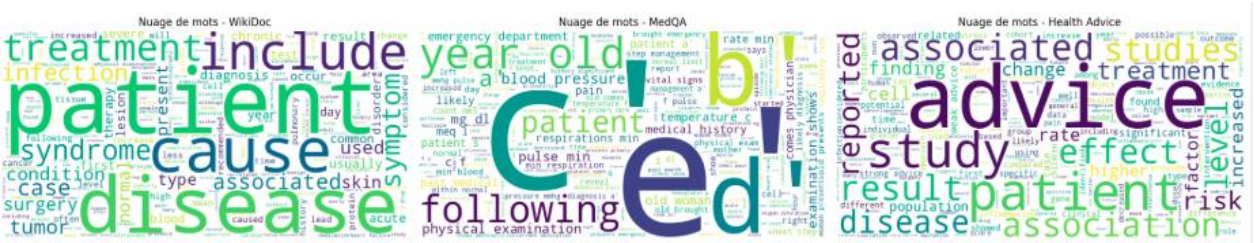


Figure 13: Nuage de mots

b. Distribution des longueurs des entrées et sorties

La distribution des longueurs des inputs et outputs a été visualisée séparément pour chaque dataset (Figure 13). On observe que :

- **WikiDoc** se caractérise par des inputs courts (moyenne de 65 tokens) et des outputs très longs, allant jusqu'à 2 000 tokens. Cela reflète son format orienté vers des descriptions détaillées de

conditions médicales.

- **MedQA**, avec une longueur moyenne des inputs de 906 tokens, contient des questions complexes nécessitant des réponses concises.
- **Health Advice** présente des inputs et outputs relativement équilibrés (inputs de 182 tokens en moyenne), adaptés à la nature pratique de ses recommandations.

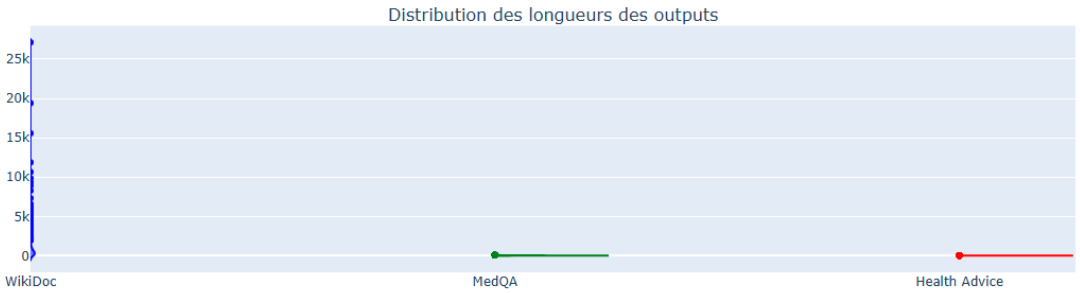


Figure 14: Distribution des longueurs

Ces résultats mettent en lumière les différences structurelles entre les datasets, influençant les stratégies de fine-tuning.

c. Distribution des catégories médicales

Une analyse plus fine a permis de catégoriser les données en cinq thématiques : *diagnostic*, *traitement*, *tests médicaux*, *prévention*, et *anatomie*. Le radar plot (Figure 3.4) montre une répartition équilibrée dans les datasets, bien que certaines catégories, comme le *traitement* et le *diagnostic*, soient plus dominantes. Cela indique que le système formé sur ces données pourra répondre efficacement à ces thématiques critiques.

La heatmap (Figure 14) illustre également la proportion de chaque catégorie par dataset. Par exemple, **Health Advice** se concentre davantage sur la *prévention*, tandis que **MedQA** couvre majoritairement les diagnostics.

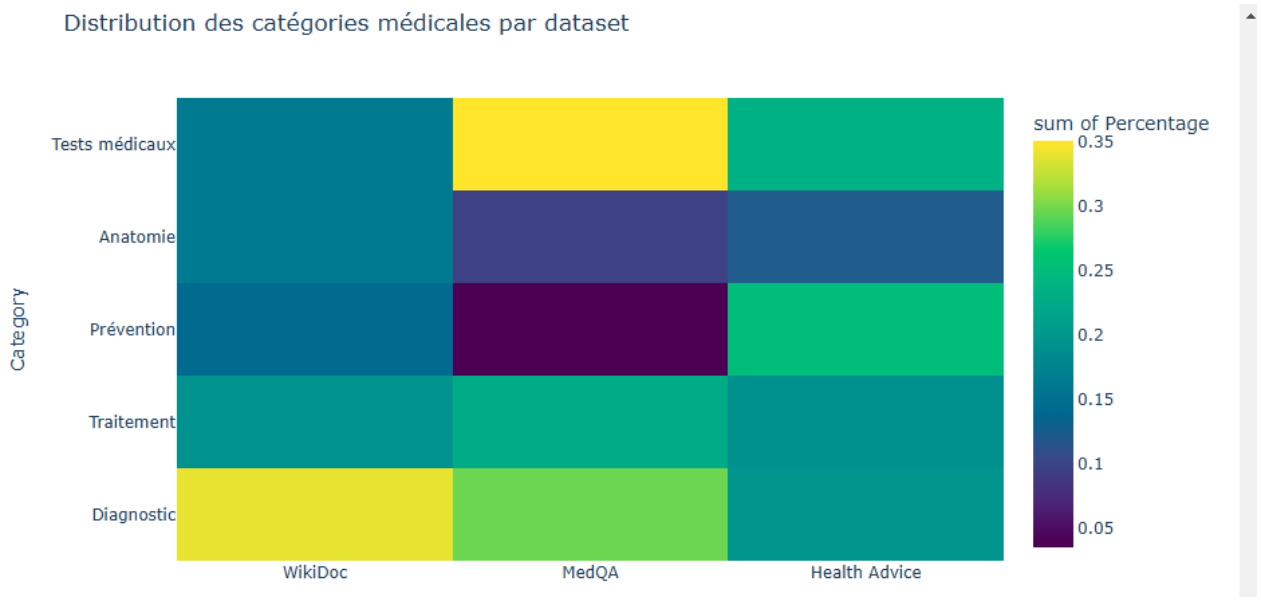


Figure 15: Heatmap des datasets

d. Statistiques descriptives des datasets

Les statistiques détaillées (Tableau 2) révèlent des informations quantitatives importantes :

- **WikiDoc** compte 10 000 exemples avec des outputs très longs (915 tokens en moyenne).
- **MedQA** contient 10 178 exemples, avec une variabilité notable dans les inputs.
- **Health Advice**, avec 8 676 exemples, est plus homogène en termes de longueur des données.

Dataset	Nombre d'exemples	Input (moyenne)	Output (moyenne)	Ratio Output/Input
WikiDoc	10 000	65	915	14.01
MedQA	10 178	906	283	0.31
Health Advice	8 676	182	85	0.47

Tableau 2: Statistiques des datasets

Ces différences doivent être prises en compte lors de la configuration du modèle pour garantir des performances optimales sur chaque type de données.

e. Interprétation et impact

Les observations issues de l' Analyse Exploratoire des données (AED) confirment que chaque dataset apporte une perspective complémentaire à la base de connaissances de Medbot. Cette diversité garantit que le système formé puisse couvrir un large éventail de cas d'utilisation, allant des questions médicales complexes aux conseils de santé pratiques. Ces résultats guideront également le choix des hyperparamètres et des stratégies de fine-tuning, comme la gestion des séquences longues pour WikiDoc ou l'ajustement des couches LoRA pour les outputs courts de MedQA.

2) Résultats des Modèles

Le processus de fine-tuning du modèle LLaMA 3.2-3B-Instruct, utilisant LoRA et une quantification en 4 bits, a permis d'atteindre une convergence stable avec une perte de validation finale de **1.5988**. Ces résultats témoignent de la capacité du modèle à généraliser efficacement sur des données non vues, malgré les contraintes imposées par les ressources limitées.

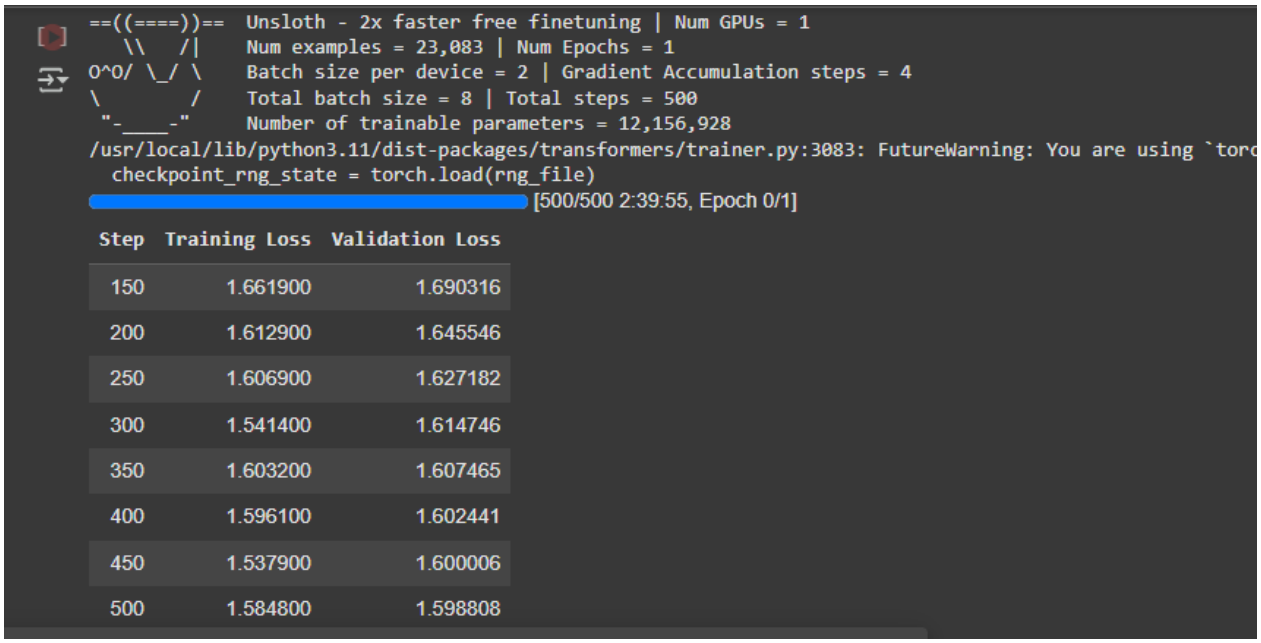


Figure 16: Entrainement du modèle

Les tests effectués sur des cas pratiques montrent des réponses pertinentes et adaptées aux questions posées. Par exemple, lorsque l'utilisateur demande *"J'ai mal au ventre depuis trois jours. Que faire ?"*, le modèle fournit une réponse structurée comprenant des conseils sur l'hydratation, l'alimentation, et les analgésiques, tout en recommandant une consultation médicale si nécessaire. De manière similaire, face à des questions ouvertes comme *"Qu'est-ce que la santé mentale ?"*, le modèle répond avec une définition claire et contextualisée.

En termes de performances techniques, les temps de traitement des requêtes varient entre **28 ms** et **90 ms**, en fonction de la complexité des questions. Cette rapidité garantit une expérience utilisateur fluide, ce qui est essentiel pour une interaction en temps réel.

Figure 17: Test sur l'API question

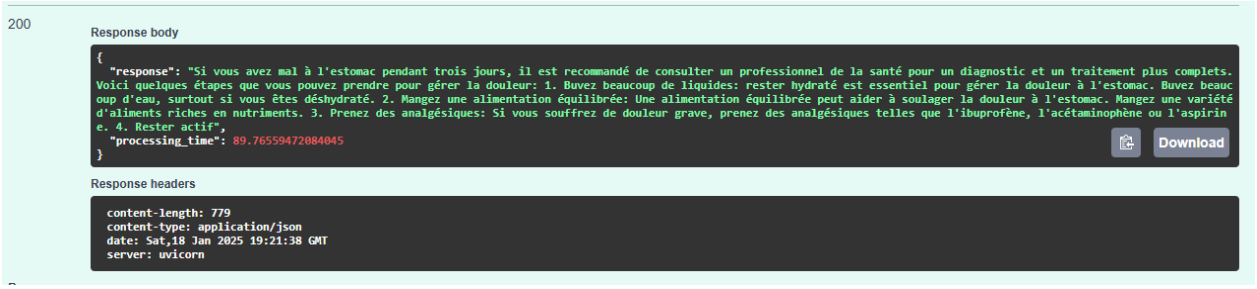


Figure 18: Test sur l'API réponse

3) Comparaison avec la Littérature

En se référant aux travaux existants, Medbot se positionne favorablement par rapport à d'autres modèles utilisés dans le domaine médical. La combinaison de LoRA et de la quantification en 4 bits s'est avérée être une stratégie efficace pour réduire les coûts de calcul sans compromettre la qualité des réponses. De plus, l'intégration de données diversifiées et multilingues confère à Medbot une polyvalence qui dépasse celle de nombreux systèmes existants.

Cependant, certaines limites subsistent. Par exemple, la gestion des séquences très longues reste un défi, et les performances en langues multiples nécessitent des validations supplémentaires pour s'assurer d'une qualité constante dans toutes les langues prises en charge.

4) Discussion

Les résultats obtenus soulignent le potentiel de Medbot à devenir un outil fiable pour la préconsultation médicale. Néanmoins, plusieurs améliorations sont envisageables. Il serait bénéfique d'enrichir les données pour couvrir davantage de spécialités médicales et d'optimiser les hyperparamètres pour améliorer encore les performances. Par ailleurs, une évaluation en conditions réelles permettrait de mieux comprendre les limites du modèle et d'identifier les ajustements

nécessaires.

Enfin, bien que les performances actuelles soient prometteuses, des efforts supplémentaires pourraient être investis dans la réduction des temps de traitement pour garantir une interaction encore plus fluide, notamment dans des scénarios à forte charge.

II. Outils de Développement et de Déploiement

Le développement de Medbot a nécessité l'intégration d'outils technologiques avancés et l'application de méthodologies modernes pour garantir une solution efficace, scalable et user-friendly. Ce chapitre présente les environnements utilisés, les technologies adoptées, ainsi que les architectures de développement et de déploiement mises en œuvre.

1) Environnement de Développement

Le choix de l'environnement de développement a joué un rôle central dans l'efficacité du projet. Les outils suivants ont été sélectionnés pour leurs avantages spécifiques :

- *Google Colab* : Ce choix s'est imposé grâce à son accès aux GPU haute performance, notamment les NVIDIA A100, qui ont permis d'entraîner et de tester les modèles dans des délais réduits. Colab offre également une interface collaborative pour partager le code et visualiser les résultats en temps réel.
- *Visual Studio Code (VSCode)* : Utilisé comme IDE principal pour la gestion des scripts Python, la conception des services backend, et l'intégration du frontend. Les extensions de VSCode ont simplifié la détection des erreurs et l'intégration avec GitHub.
- *Hugging Face Hub* : Cet outil a été utilisé pour héberger et gérer les modèles fine-tunés, facilitant leur réutilisation et leur déploiement.

Ces outils ont permis de maintenir une approche modulaire, collaborative et orientée performance.

2) Technologies de Développement

Plusieurs technologies modernes ont été mobilisées pour construire Medbot. Elles ont été soigneusement sélectionnées pour répondre aux exigences du projet, tant au niveau des performances que de la simplicité d'intégration. La figure ci-dessous montre les différentes technologies utilisées.

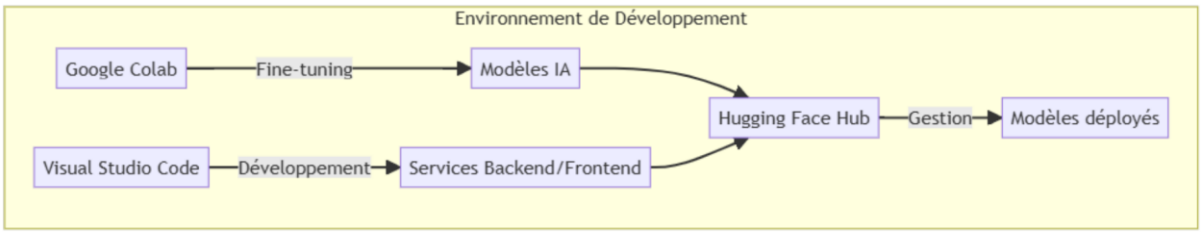


Figure 19: Technologies utilisées

A. Backend :

- FastAPI : Framework performant pour la gestion des requêtes utilisateur et la création de points de terminaison RESTful. Il offre une documentation interactive intégrée via Swagger.
- PyTorch : Utilisé pour le fine-tuning du modèle LLaMA 3.2-3B-Instruct. PyTorch a été préféré pour sa flexibilité et son support des tâches NLP.
- Transformers (Hugging Face) : Fournit des outils avancés pour manipuler les modèles pré-entraînés et les adapter à des tâches spécifiques.

B. Frontend :

- **React.js :** Choisi pour développer une interface utilisateur intuitive, permettant aux utilisateurs de poser des questions via texte ou audio.La figure ci-dessous présente l'interface React de l'application.

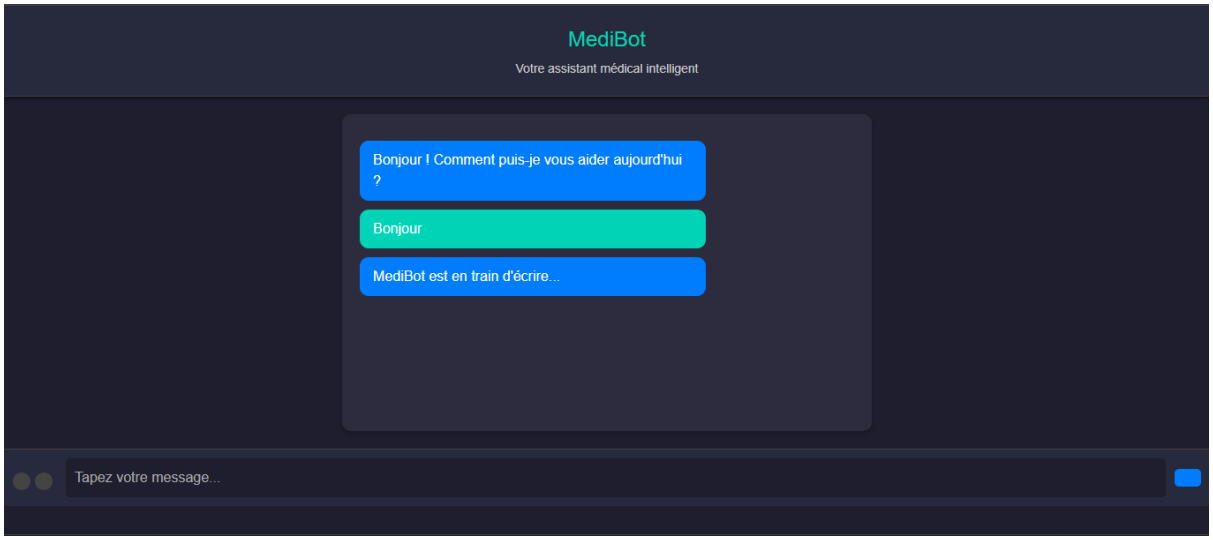


Figure 20: Interface React

- **Gradio :** Utilisé pour tester rapidement les fonctionnalités interactives du chatbot.

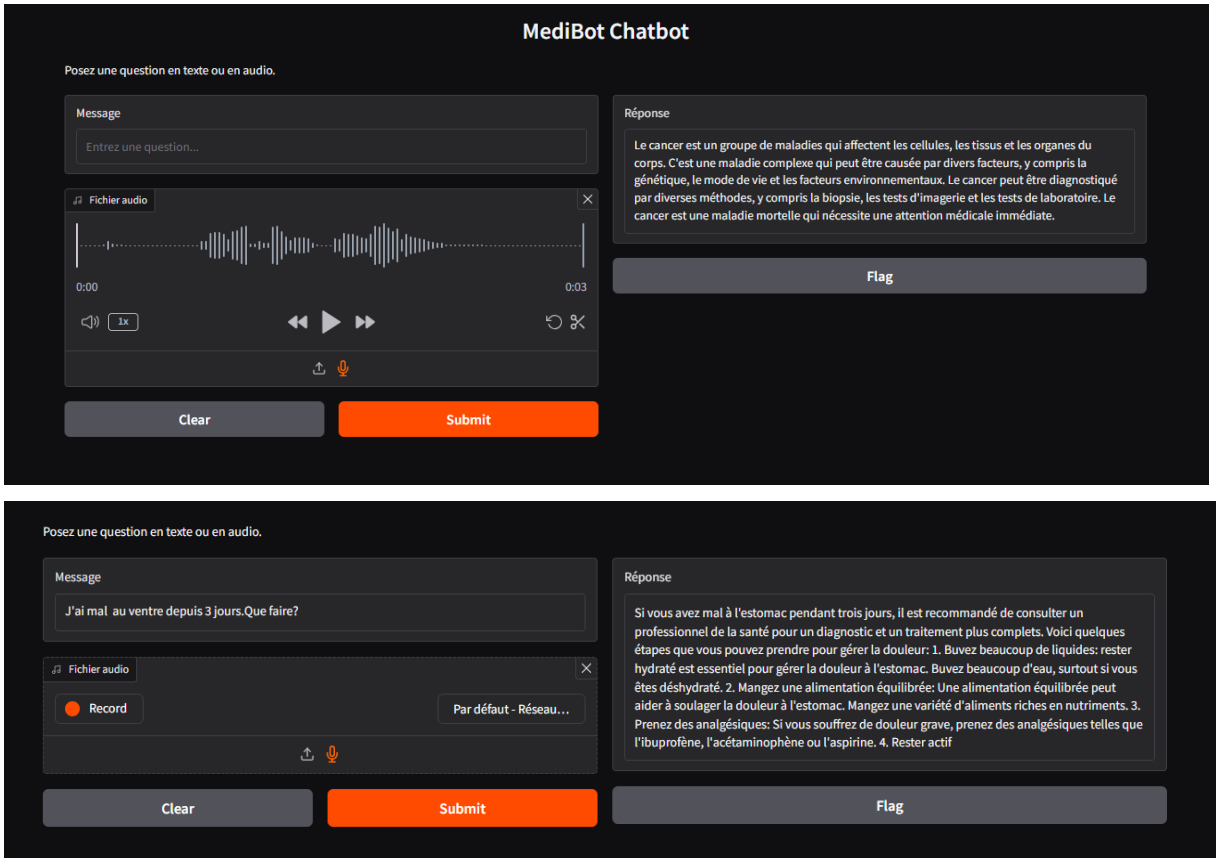


Figure 21: Interface test gradio

C. Gestion des versions et collaboration :

- **GitHub :** Plateforme de gestion du code source, facilitant la collaboration et l'intégration continue.

Le tableau suivant présente les technologies utilisées pour chaque composant du système.

Composant	Technologie	Rôle
Backend	FastAPI, PyTorch	Gestion des requêtes, fine-tuning
Frontend	React.js, Gradio	Interface utilisateur interactive
Gestion de versions	GitHub	Versionnement et collaboration
Hébergement des modèles	Hugging Face Hub	Stockage et déploiement des modèles

Tableau 3: Technologies utilisées pour chaque composant du système.

3) Architecture de la Solution

Medbot repose sur une architecture modulaire et cohérente, facilitant l'intégration des différents composants. Cette structure est optimisée pour des interactions multimodales, comprenant texte, audio, et images.

- Structure globale :

1. Interface Utilisateur (React.js) : Permet aux utilisateurs d'interagir avec le chatbot via des requêtes textuelles ou vocales.
2. API Backend (FastAPI) : Reçoit les requêtes utilisateur, les transmet au modèle LLaMA, et renvoie les réponses au frontend.
3. Pipeline Multimodal :
    - Texte : Les requêtes textuelles sont directement traitées par le modèle LLaMA.
    - Audio : Le modèle openai/whisper-small convertit les entrées audio en texte.
    - Images : Bien qu'encore en développement, un module basé sur des modèles de vision sera intégré pour analyser les symptômes visibles.

- Service de Traduction :

Pour garantir une interaction multilingue, le modèle facebook/nllb-200-distilled-600M est utilisé, traduisant efficacement les requêtes et réponses entre le français et l'anglais.

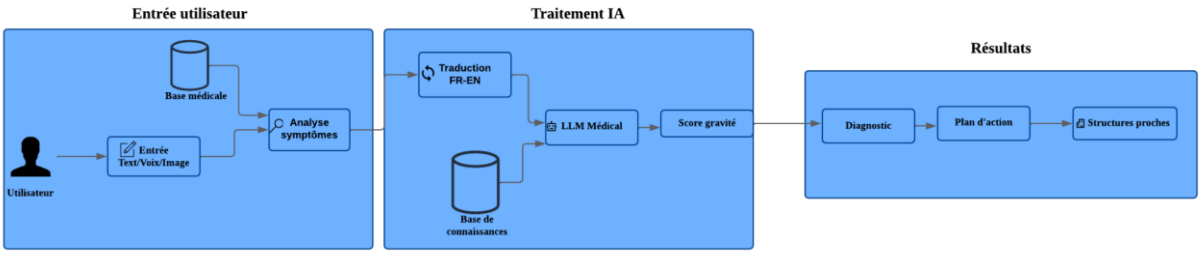


Figure 22: Architecture globale de Medbot illustrant les interactions entre les différents modules

4) Architecture et Services de Déploiement

- Conteneurisation : Bien que Docker ne soit pas encore utilisé, son intégration est envisagée pour simplifier les déploiements futurs dans des environnements variés.
- Hébergement :Les services backend sont hébergés sur des plateformes cloud comme Render, garantissant une haute disponibilité et scalabilité.Les modèles fine-tunés

quant à eux sont stockés sur Hugging Face Hub, assurant une gestion centralisée et un déploiement simplifié.

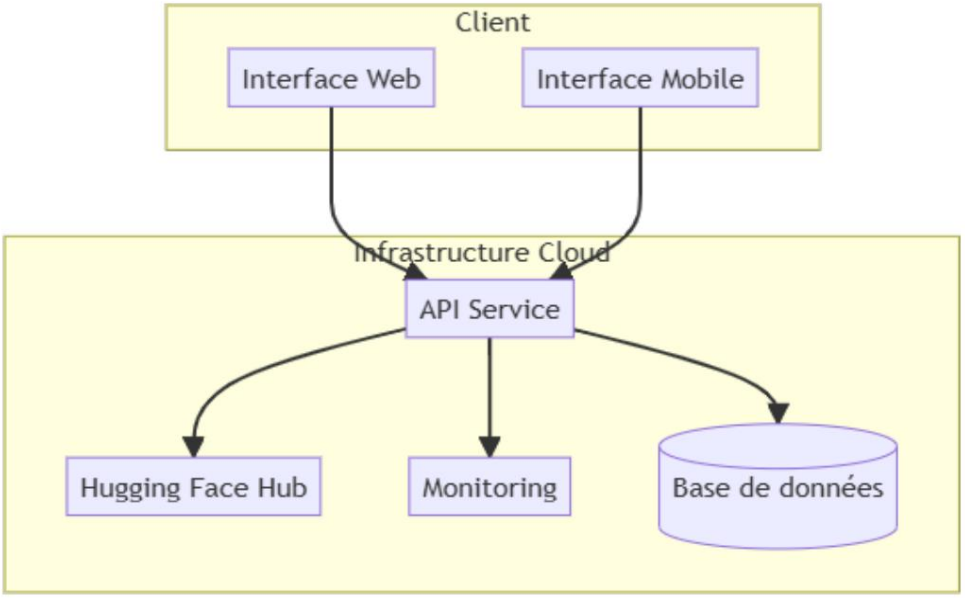


Figure 23: Services du déploiement

Conclusion

Ce chapitre a détaillé l'implémentation de **Medbot**, depuis le développement jusqu'aux résultats obtenus. Les outils et technologies utilisés, comme FastAPI et React, ont permis de concevoir une solution cohérente et performante. Les résultats montrent que le chatbot est prêt à répondre aux besoins des utilisateurs, tout en ouvrant la voie à des améliorations futures. Le prochain chapitre se concentrera sur les perspectives et recommandations pour optimiser cette plateforme.

CONCLUSION GENERALE

Le développement de Medbot constitue une contribution prometteuse dans l'utilisation de l'intelligence artificielle pour répondre aux besoins de santé, en particulier dans les contextes sous-desservis. Ce projet a permis de concevoir un assistant médical intelligent capable d'interagir avec les utilisateurs via texte et audio, tout en générant des recommandations adaptées à partir de données validées. L'analyse approfondie des données et les modèles fine-tunés, associés à une architecture performante, ont démontré la pertinence et la faisabilité de cette solution.

Cependant, le travail accompli constitue une première étape, et de nombreuses perspectives s'offrent pour enrichir et perfectionner Medbot. Une priorité essentielle est l'intégration de langues africaines telles que le wolof, afin de rendre l'outil encore plus accessible et inclusif. Par ailleurs, le développement de la fonctionnalité d'analyse d'images médicales, en s'appuyant sur des données spécifiques aux réalités africaines, renforcera les capacités de diagnostic visuel de l'assistant.

L'orientation des utilisateurs vers des professionnels de santé constitue également une perspective clé. Cette fonctionnalité permettra à Medbot d'aller au-delà des conseils initiaux en guidant les patients vers les soins adaptés à leurs besoins. En parallèle, le respect des considérations éthiques, notamment en matière de protection des données, sera au cœur des futures améliorations pour garantir la confiance des utilisateurs.

Enfin, rendre le chatbot accessible au grand public représente une étape importante. Cela nécessitera un déploiement à grande échelle, des tests rigoureux en conditions réelles, et des ajustements pour répondre aux retours des utilisateurs. En intégrant ces évolutions, Medbot aspire à devenir un outil indispensable pour améliorer l'accès aux soins de santé, en particulier dans les régions où les ressources médicales sont limitées.

Ainsi, la conclusion de cette étude ouvre la voie à un avenir prometteur pour Medbot, où innovation technologique et impact social convergent pour transformer les pratiques de santé dans des environnements variés et complexes.

WEBOGRAPHIE

[1] McCarthy, J., Minsky, M. L., Rochester, N., Shannon, C. E., *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*, *AI Magazine*, vol. 27, no. 4, pp. 12-14, 2006. Disponible à : https://doi.org/10.1609/aimag.v27i4.1904. Consulté le : 05/01/2025.

[2] Brown, T., Mann, B., Ryder, N., et al., *Language Models are Few-Shot Learners*, *arXiv preprint*, vol. 2005.14165v4 [cs.CL], Juin 2020. Disponible à : https://arxiv.org/abs/2005.14165. Consulté le : 05/01/2025.

[3] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, *arXiv preprint*, vol. 1810.04805v2 [cs.CL], Mai 2019. Disponible à : https://arxiv.org/abs/1810.04805. Consulté le : 05/01/2025.

[4] Topol, E. J., *High-performance medicine: the convergence of human and artificial intelligence*, *Nature Medicine*, vol. 25, no. 1, pp. 44-56, 2019. Disponible à : https://doi.org/10.1038/s41591-018-0300-7. Consulté le : 05/01/2025.

[5] Yu, K. H., Beam, A. L., Kohane, I. S., *Artificial intelligence in healthcare*, *Nature Biomedical Engineering*, vol. 2, no. 10, pp. 719-731, 2018. Disponible à : https://doi.org/10.1038/s41551-018-0305-z. Consulté le : 05/01/2025.

[6] World Health Organization, *Universal Health Coverage: Key Facts*, *WHO Fact Sheets*, 2023. Disponible à : https://www.who.int/news-room/fact-sheets/detail/universal-health-coverage-(uhc). Consulté le : 05/01/2025.

[7] World Health Organization, *Global Health Workforce Statistics*, *WHO Data Repository*, 2023. Disponible à : https://www.who.int/data/gho/data/themes/topics/health-workforce. Consulté le : 05/01/2025.

[8] Jiang, F., Jiang, Y., Zhi, H., et al., *Artificial intelligence in healthcare: past, present and future*, *Stroke and Vascular Neurology*, vol. 2, no. 4, pp. 230-243, 2017. Disponible à : http://dx.doi.org/10.1136/svn-2017-000101. Consulté le : 05/01/2025.

[9] Liu, S., Davison, E., Schork, N. J., *The role of large language models in medicine*, *Nature Medicine*, vol. 29, pp. 2722-2724, 2023. Disponible à : https://doi.org/10.1038/s41591-023-02542-x. Consulté le : 05/01/2025.

[10] Weizenbaum, J., *ELIZA—A Computer Program For the Study of Natural Language Communication Between Man and Machine*, *Communications of the ACM*, vol. 9, no. 1, pp. 36-45, 1966. Disponible à : https://doi.org/10.1145/365153.365168. Consulté le : 11/01/2025.

[11] Wikipédia, *Chatbot*. Disponible à : https://en.wikipedia.org/wiki/Chatbot. Consulté le : 23/01/2025.

[12] Gupta, M., Ranjan, R., Bhatia, S., *Survey on Chatbot Frameworks: Types and Applications*, *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 8, no. 1, pp. 1-12, 2020. Disponible à : https://www.ijircce.com. Consulté le : 05/01/2025.

[13] Carvalho D'Ávila, T., Kino, *An Approach for Rule-Based Chatbot Development, Monitoring and Evaluation*, Dissertation, Université Fédérale du Minas Gerais, 2018. Disponible à : https://repositorio.ufmg.br/bitstream/1843/30618/1/DissertacaoThiagoDAvila.pdf. Consulté le : 05/01/2025.

[14] Braun, D., Matthes, F., *Towards a Framework for Classifying Chatbots*, *ICEIS Proceedings*, vol. 1, pp. 496–501, 2019. Disponible à : https://www.researchgate.net/publication/332902947_Towards_a_Framework_for_Classifying_Chatbots. Consulté le : 05/01/2025.

[15] Ahmad, N. A., Che, M. H., Zainal, A., Abd Rauf, M. F., Adnan, Z., *Review of Chatbots Design Techniques*, *International Journal of Computer Applications*, vol. 181, no. 8, pp. 7–10, 2018. Disponible à : https://www.researchgate.net/publication/327097910_Review_of_Chatbots_Design_Techniques. Consulté le : 05/01/2025.

[16] Nimavat, K., Champaneria, T., *Chatbots: An Overview Types, Architecture, Tools and Future Possibilities*, *International Journal of Scientific Research and Development*, vol. 5, no. 7, pp. 1019–1024, 2017. Disponible à : https://www.researchgate.net/publication/320307269_Chatbots_An_overview_Types_Architecture_Tools_and_Future_Possibilities. Consulté le : 05/01/2025.

[17] Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K., *A Survey of Design Techniques for Conversational Agents*, *International Conference on Information, Communication and Computing Technology Proceedings*, pp. 336–350, 2017. Disponible à : https://www.researchgate.net/publication/320303397_A_Survey_of_Design_Techniques_for_Conversational_Agents. Consulté le : 05/01/2025.

[18] Dilmegani, C., *Natural Language Platforms: Top NLP APIs & Comparison*, AIMultiple, 2021. Disponible à : https://research.aimultiple.com/natural-language-platforms. Consulté le : 05/01/2025.

[19] *Evolution of Chatbots*, Capacity, 2025. Disponible à : https://capacity.com/chatbots/evolution-of-chatbots/. Consulté le : 02/01/2025.

[20] *Types of Chatbots*, Yellow.ai, 2025. Disponible à : https://yellow.ai/blog/types-of-chatbots/. Consulté le : 02/01/2025.

[21] *MedLLMs Practical Guide*, GitHub Repository, 2024. Disponible à : https://github.com/AI-in-Health/MedLLMsPracticalGuide. Consulté le : 04/11/2024.

[22] Jin, D., Pan, E., Oufattole, N., et al., *What Disease Does This Patient Have? A Large-scale Open Domain Question Answering Dataset from Medical Exams*, *arXiv preprint*, vol. 2009.13081v1 [cs.CL], Sept. 2020. Disponible à : https://arxiv.org/pdf/2009.13081v1. Consulté le : 04/11/2024.

[23] WikiDoc, *Medical Knowledge Sharing Platform for Professionals*. Disponible à : https://www.wikidoc.org/index.php/Main_Page. Consulté le : 04/11/2024.

[24] WikiDoc Dataset Description, *Patient Information and Living Textbook for Medical Knowledge*. Disponible à : https://www.wikidoc.org/index.php/Dataset_Description. Consulté le : 04/11/2025.

[25] Yu, B., Li, Y., Wang, J., *Detecting Causal Language Use in Science Findings*, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, 2019, pp. 4664-4674. Disponible à : https://aclanthology.org/D19-1473. Consulté le : 04/11/2024.

[26] Radford, A., Wu, J., Child, R., et al., *Language Models Are Few-Shot Learners*, *arXiv preprint*, vol. 2005.14165v4 [cs.CL], Juin 2020. Disponible à : https://arxiv.org/abs/2005.14165. Consulté le : 04/11/2024.

[27] Rasmy, L., Min, E., et al., *MedGPT: An Expert-level Medical Question Answering System*, *arXiv preprint*, 2021. Disponible à : https://arxiv.org/abs/2111.08514. Consulté le : 04/11/2025.

[28] Shoham, Y., Rappoport, A., *Clinical Language Models and Their Role in Modern Healthcare*, *Bioinformatics Review*, vol. 38, no. 5, 2023. Disponible à :

https://doi.org/10.1093/bioinformatics/btaa123. Consulté le : 04/11/2025.

[29] *NLLB-200 Distilled 600M*, Hugging Face, 2024. Disponible à : https://huggingface.co/facebook/nllb-200-distilled-600M. Consulté le : 10/11/2024.

[30] Barektain, M., Nawalgaria, A., Mankowitz, D. J., Merey, M. A., *Foundational Large Language Models & Text Generation*. Disponible à : https://datasciencedojo.com/blog/llama-model-debate/. Consulté le : 10/10/2024.

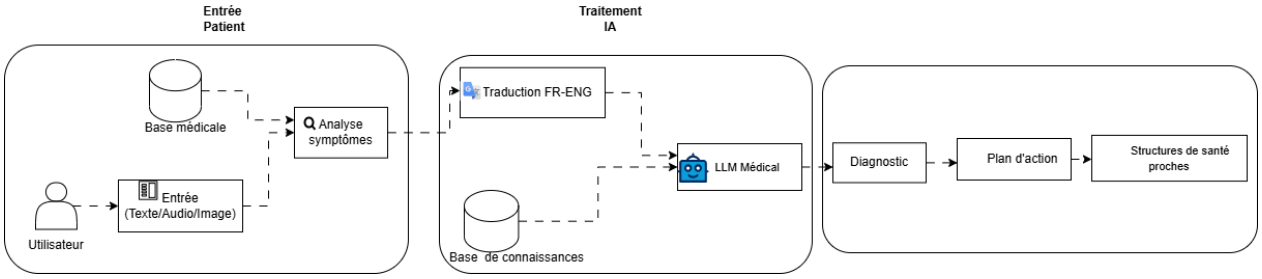
[31] Wikipedia, *Natural Language Processing*. Disponible à : https://en.wikipedia.org/wiki/Natural_language_processing. Consulté le : 10/10/2024.

[32] *LLaMA Model Debate*, Data Science Dojo Blog. Disponible à : https://datasciencedojo.com/blog/llama-model-debate/. Consulté le : 20/10/2024.

[33] vignesh yaadav ,Exploring and building the LLaMA 3 Architecture : A Deep Dive into Components, Coding, and Inference Techniques. Disponible à : https://medium.com/@vi.ai_/exploring-and-building-the-llama-3-architecture-a-deep-dive-into-components-coding-and-43d4097cfbbb Consulté le : 10/01/2025

ANNEXES

Annexe 1 : Diagramme globale de la solution



Annexe 2 : Script de développement

```python
from fastapi.middleware.cors import CORSMiddleware
from pydantic import BaseModel
from typing import Optional
from transformers import pipeline
import time
import os
import soundfile as sf
import logging

# Initialisation de l'application FastAPI
app = FastAPI(title="MedBot API")

# Configuration des CORS
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

# Configuration des logs
logging.basicConfig(level=logging.INFO)

# Initialisation des pipelines
translator = pipeline("translation", model="facebook/nllb-200-distilled-600M")
medical_model = pipeline("text-generation", model="LaurianeMD/MedLlama3.2-3B-V2")
asr_model = pipeline("automatic-speech-recognition", model="openai/whisper-small")
narrator_fr = pipeline("text-to-speech", model="facebook/mms-tts-fra")
```

```python
@app.post("/api/chat")
async def process_chat(
    message: Optional[str] = Form(None),
    file: Optional[UploadFile] = None
):
    if not message and not file:
        return {"error": "Veuillez fournir un message ou un fichier audio."}

    try:
        start_time = time.time()
        final_response = None

        # Traitement pour le message texte
        if message:
            logging.info("Traitement du message texte.")
            english_text = translator([message], src_lang="fra_Latn", tgt_lang="eng_Latn")
            translated_text = english_text[0]["translation_text"]

            llm_response = medical_model(translated_text, max_length=500, num_return_sequences=1)
            generated_text = llm_response[0]["generated_text"]

            french_response = translator([generated_text], src_lang="eng_Latn", tgt_lang="fra_Latn")
```

```
        # Traitement pour le fichier audio
        elif file:
            logging.info("Traitement du fichier audio.")
            audio_path = f"temp_audio_{int(time.time())}.wav"
            with open(audio_path, "wb") as audio_file:
                audio_file.write(await file.read())

            asr_output = asr_model(audio_path)
            recognized_text = asr_output["text"]

            english_text = translator([recognized_text], src_lang="fra_Latn", tgt_lang="eng_Latn")
            translated_text = english_text[0]["translation_text"]

            llm_response = medical_model(translated_text, max_length=500, num_return_sequences=1)
            generated_text = llm_response[0]["generated_text"]

            french_response = translator([generated_text], src_lang="eng_Latn", tgt_lang="fra_Latn")
            final_response = french_response[0]["translation_text"]

            narrated_text = narrator_fr(final_response)
            narrated_audio = narrated_text["audio"]
            narrated_audio_path = f"narrated_audio_{int(time.time())}.wav"
            sf.write(narrated_audio_path, narrated_audio, samplerate=narrated_text["sampling_rate"])
```

```
            # Supprimer le fichier temporaire
            if os.path.exists(audio_path):
                os.remove(audio_path)

        processing_time = time.time() - start_time
        return {
            "response": final_response,
            "processing_time": processing_time
        }

    except Exception as e:
        logging.error("Erreur : %s", str(e))
        return {"error": str(e)}

@app.get("/")
def root():
    return {"message": "Bienvenue sur l'API MedBot !"}
```

Annexe 3 : Script de l'interface test avec Gradio

```python
import gradio as gr
import requests
from transformers import pipeline

# Initialisation du modèle de reconnaissance vocale
asr_model = pipeline("automatic-speech-recognition", model="openai/whisper-small")

# Fonction pour appeler l'API backend
def chatbot_interaction(message=None, audio_file=None):
    try:
        if audio_file:
            # Reconnaissance vocale locale
            recognized_text = asr_model(audio_file)["text"]
            # Envoi du texte reconnu au backend
            data = {'message': recognized_text}
            response = requests.post("https://medllm.onrender.com/api/chat", data=data)
        elif message:
            # Envoi de message texte directement
            data = {'message': message}
            response = requests.post("https://medllm.onrender.com/api/chat", data=data)
        else:
            return "Veuillez fournir un message ou un fichier audio."

        # Retourne la réponse de l'API
        return response.json().get('response', 'Erreur lors du traitement.')

    except Exception as e:
        return f"Erreur : {str(e)}"
```

```python
# Interface utilisateur avec Gradio
interface = gr.Interface(
    fn=chatbot_interaction,
    inputs=[
        gr.Textbox(label="Message", placeholder="Entrez une question..."),
        gr.Audio(type="filepath", label="Fichier audio"),
    ],
    outputs=gr.Textbox(label="Réponse"),
    title="MedBot",
    description="Posez une question en texte ou en audio."
)

# Lancer l'interface
if __name__ == "__main__":
    interface.launch()
```

TABLE DES MATIÈRES