

Licence Sciences Mathématique et Informatique

**Projet De Fin D’étude**

CAMEL : Développement d’un Chabot basé sur LLAMA 2

**Réalisé par :**

ELKAISSOUNI ELMEHDI

GOURGAIZ BRAHIM

HANDACHE MOHAMMED

**Soutenu le**03 juin 2024 **devant le jury composé de :**

Prof. BEN LAHMAR EL HABIB - Encadrant

Année Universitaire : 2023/2024



#### Dédicaces

Grâce à la bénédiction de Dieu et son aide nous avons pu accomplir ce modeste travail.

Nous dédions ce travail A nos chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de nos études.

A nos chères sœurs et frères pour leurs encouragements permanents, et leur soutien moral.

A tous les étudiants de la promotion 2023/2024,

Option : Science Mathématique et informatique.

Sans oublier toute l’équipe pédagogique de la faculté des sciences BEN M’SIK à qui on garde un grand respect et gratitude pour leur enseignement, aide, et encadrements pertinents et inconditionnel.

#### Remerciement

Au préambule à ce projet de fin d’étude nous remerciant Dieu qui nous aide et nous donne la patience et le courage durant ces années d’études.

Il est souvent difficile de remercier les gens qui vous aident à accomplir les tâches qui vous sont données, et pourtant nous nous devons d’exprimer l’entière gratitude que nous ressentons envers eux.

Nous tenons à exprimer notre profonde gratitude à notre encadrant Prof. BEN LAHMAR EL HABIB pour son encadrement, sa disponibilité, ses conseils avisés et ses efforts pertinents, qui ‘il nous a accordé tout au long de ce projet

Nos remerciements s’adressent également à l’ensemble du Corps Professoral de FSBM pour tout leur savoir-faire qu’ils ont su nous procurer.

Sans oublier nos parents, nos familles, pour leurs contributions, leurs soutiens, leurs patiences et encouragement au cours de la réalisation de ce projet de fin d’étude, pour que nous puissions arriver là où nous en sommes.

Enfin, nous gardons une pensée reconnaissante envers tous ceux qui nous ont aidés de près ou de loin à réaliser ce travail.

#### Résumé

Ce rapport présente toutes les étapes que nous avons suivies pour réaliser notre projet de fin d’études, qui consiste à créer un **chatbot** basé sur le modèle **Llama2**, dans le but de réduire les **biais**, en particulier concernant le sujet du **Palestine**, Motivé par les biais présents dans les chatbot populaires, nous avons consisté à sélectionner soigneusement les données de notre **DataSet** et d’utiliser la méthode de **fine-tuning** pour entraîner le modèle.

Par la suite, nous avons développé une application web pour offrir une expérience utilisateur optimisée. Grâce à cette application, il est facile d’interagir avec le modèle final.

**Mot clés**: chatbot, Llama2, biais, Palestine, DataSet, fine-tuning.

#### ABSTRACT

This report presents all the steps we followed to complete our final-year project, which involves creating a **chatbot** based on the **Llama2** model, with the goal of reducing **biases**, particularly regarding the topic of **Palestine**. Motivated by the biases present in popular chatbots, we carefully selected the data for our **dataset** and used the **fine-tuning** method to train the model.

Subsequently, we developed a web application to offer an optimized user experience. With this application, it is easy to interact with the final model.

**Keywords:** chatbot, Llama2, bias, Palestine, dataset, fine-tuning.

#### Table des matières

[Dédicaces 1](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807306)

[Remerciement 2](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807307)

[Résumé 3](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807308)

[ABSTRACT 4](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807309)

[Table des matières 5](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807310)

[Liste des figures 7](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807311)

[Liste des Abréviations 8](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807312)

[INTRODUCTION GENERALE 9](#_Toc166807313)

[Chapitre I : Contexte général du projet 10](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807314)

[Introduction 10](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807315)

[1. Contexte 11](#_Toc166807316)

[2. Problématique 11](#_Toc166807317)

[3. Objectifs 11](#_Toc166807318)

[4. Périmètre du projet 11](#_Toc166807319)

[5. Besoins fonctionnels 12](#_Toc166807320)

[6. Besoins non fonctionnels 12](#_Toc166807321)

[7. Méthodologie et formalismes adoptés 12](#_Toc166807322)

[i. Méthode SCRUM 12](#_Toc166807330)

[ii. Outils de gestion 13](#_Toc166807338)

[iii. Diagramme de Gantt 14](#_Toc166807339)

[Conclusion 14](#_Toc166807340)

[Chapitre II : L’état de l’art 15](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807341)

[Introduction 15](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807342)

[1. LLAMA 2 : Des LLMs open-source performants et sécurisés 16](#_Toc166807343)

[2. Introduction au développement de Chatbots 16](#_Toc166807344)

[3. Les Chatbots modernes :  Revue technique 17](#_Toc166807345)

[4. Développement d’un Chatbot pour l'éducation en machine Learning 17](#_Toc166807346)

[Conclusion 18](#_Toc166807347)

[Chapitre III : Etude préalable du projet 19](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807348)

[Introduction 19](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807349)

[1. Généralités sur LLM 20](#_Toc166807350)

[i. Définition 20](#_Toc166807352)

[ii. Les principes de base 20](#_Toc166807353)

[iii. Défis et limites 21](#_Toc166807354)

[2. Fine tuning 22](#_Toc166807355)

[3. Généralités sur LLAMA-2 22](#_Toc166807356)

[i. Introduction 22](#_Toc166807357)

[ii. Comparaison de LLAMA-2 avec GPT-3.5 23](#_Toc166807358)

[4. Chatbot 24](#_Toc166807359)

[5. Dataset 25](#_Toc166807360)

[Conclusion 25](#_Toc166807361)

[Chapitre IV : Conception du Chatbot 26](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807362)

[Introduction 26](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807363)

[1. Définition de l’UML : 27](#_Toc166807364)

[2. Conception du Chatbot 27](#_Toc166807365)

[i. Diagramme de cas d’utilisation 27](#_Toc166807366)

[ii. Diagrammes de séquence 29](#_Toc166807367)

[iii. Diagramme de classes 32](#_Toc166807368)

[Conclusion 34](#_Toc166807369)

[Chapitre V : Réalisation du Chatbot 35](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807370)

[Introduction 35](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807371)

[CONCLUSION & PRESPECTIVES 36](#_Toc166807372)

[Références 37](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807373)

[Webographie 38](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807374)

#### Liste des figures

[Figure 1: méthode SCRUM 13](#_Toc166807386)

[Figure 2: Architecture Transformers 21](#_Toc166807387)

[Figure 3: Fine tuning process 22](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807388)

[Figure 4: L'entraînement de LLAMA-2-chat 23](#_Toc166807389)

[Figure 5: Benchmark des performances des LLMs 24](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807390)

[Figure 6: Benchmark de la sécurité des LLMs 24](#_Toc166807391)

[Figure 7: Diagramme de cas d'utilisation 29](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807392)

[Figure 8: Diagramme de séquence - Connexion 30](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807393)

[Figure 9: Diagramme de séquence - Inscription 31](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807394)

[Figure 10: Diagramme de séquence - Accéder à l'historique 32](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807395)

[Figure 11: Diagramme de séquence - saisie d'un message 32](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807396)

[Figure 12: Dictionnaire de données 33](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807397)

[Figure 13: Diagramme de classes 34](file:///C:\Users\brahim\Downloads\Telegram%20Desktop\Rapport.mod1%20(4)%20(3).docx#_Toc166807398)

#### Liste des Abréviations

|  |  |
| --- | --- |
| Abréviation | Description |
| LLAMA | Large Language Model Meta AI (Meta IA pour les Modèles de Langage Large) |
| LLM | Large Language Model (Modèle de Langage Large) |
| NLP | Natural Language Processing (Traitement Automatique du Langage Naturel) |
| GPT | Generative Pre-trained Transformers (Générateur Pré-entrainé de Transformer) |
| AI | Artificial Intelligence (Intelligence Artificielle) |
| UML | Unified Modeling Language (langage de modélisation unifié) |
| SFT | Supervised Fine-Tuning (Affinage Supervisé) |
| RLHF | Reinforcement Learning with Human Feedback (Apprentissage par renforcement avec retour humain) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## INTRODUCTION GENERALE

Dans une ère où les technologies de traitement du langage naturel évoluent rapidement, notre projet s'inscrit dans une démarche novatrice visant à concevoir et développer un Chatbot avancé. Ce Chatbot, alimenté par (LLM) tel que LLAMA 2, aspire à répondre à une variété de questions et à interagir de manière intelligente avec les utilisateurs. Notre motivation principale réside dans la nécessité de surmonter les limites observées dans les modèles actuels, notamment en ce qui concerne le comportement potentiellement biaisé, tel que celui observé chez les ai.

Le comportement parfois évasif ou biaisé des ai, particulièrement notable dans des domaines sensibles comme la situation en Palestine, soulève des préoccupations majeures quant à l'impartialité et à la fiabilité des systèmes de dialogue automatisés. Ces comportements peuvent découler de divers facteurs, qu'il s'agisse de directives de conception délibérées ou de biais présents dans les données d'entraînement. Ainsi, afin d'instaurer une interaction juste et impartiale, il devient impératif de concevoir un Chatbot capable de fournir des informations précises et objectives, sans aucun parti pris.

Pour répondre à ces défis, notre objectif est de développer un Chatbot avancé, capitalisant pleinement sur les avancées du traitement du langage naturel offertes par le (LLM). Guidés par une méthodologie rigoureuse, notre priorité est d'éliminer tout potentiel de partialité et de garantir l'intégrité de nos réponses, surtout sur des sujets sensibles tels que la situation en Palestine.

Pour concrétiser ces objectifs, notre projet se concentrera sur des piliers essentiels. Nous entreprendrons une analyse approfondie des données d'entraînement, élaborerons des stratégies pour éviter les biais et minimiser les risques d'erreurs dans les réponses du Chatbot. Grâce à une collaboration multidisciplinaire, notre ambition est de hisser le domaine des interfaces de dialogue automatisées vers des standards inégalés d'impartialité et de fiabilité.

**Chapitre I**



Le but de ce chapitre est de situer notre projet dans son contexte général. Nous débuterons par une présentation de ce contexte, mettant en lumière les défis et les enjeux auxquels il répond. Ensuite, nous exposerons les objectifs et les fonctionnalités du projet, tout en délimitant clairement son périmètre et déterminant ses besoins. Nous aborderons également la méthodologie et les formalismes adoptés pour sa mise en œuvre, fournissant ainsi un cadre structuré pour notre travail.

# Chapitre I : Contexte général du projet

## Introduction

## Contexte

Notre projet consiste à développer un Chatbot exploitant LLAMA 2, ayant pour objectif principal d'éliminer les biais, en particulier ceux concernant la question sensible de la Palestine. Ce Chatbot sera spécifiquement conçu pour fournir des informations impartiales et objectives sur ce sujet, évitant toute partialité dans ses interactions avec les utilisateurs.

## **Problématique**

Notre choix de ce sujet découle du comportement parfois évitant des Chatbots en général, notamment dans le traitement des questions relatives à la Palestine. Ce comportement peut être dû à une directive de conception délibérée et peut être biaisé. Les Chatbots en général peuvent refléter les biais présents dans les données d'entraînement. Par exemple, si le modèle a été entraîné sur des données présentant des stéréotypes sexistes, racistes ou politiques, cela peut se refléter dans ses réponses.

## Objectifs

* Notre objectif est de développer un Chatbot en utilisant LLAMA 2 comme modèle de base, capable de fournir des réponses sans biais sur le sujet de la Palestine. Pour ce faire, nous lui fournirons une base de données vaste, bien structurée et honnête, reposant sur des ressources vérifiées et contenant des informations authentiques. Nous nous assurerons qu'il puisse fournir des réponses précises et cohérentes, sans introduire d'erreurs ou d'interprétations erronées.
* Concevoir une interface utilisateur conviviale est essentiel pour offrir une expérience utilisateur positive.

## Périmètre du projet

Notre projet vise à élaborer un périmètre clair et précis. Nous allons développer un Chatbot exploitant LLAMA 2, pour objectif central l'éradication des biais, notamment ceux liés à la question délicate de la Palestine. Ce Chatbot sera spécifiquement conçu pour fournir des informations impartiales et objectives sur ce sujet.

## Besoins fonctionnels

* D’abord l’utilisateur va se connecter avec son email et son mot de passe, s’il n’avait pas un compte il doit le créer avec un nom, un email et un nouveau mot de passe bien sécurisé.
* Après compléter la formule d’inscription, l’utilisateur doit valider son email.
* Puis, en accédant à la page principale qui se compose d’un input pour saisir et une partie d’affichage du texte généré.
* L’utilisateur peut revenir à son historique de chat qu’il peut le supprimer, l’archiver ou le renommer et qui se trouve au côté gauche de la page.
* En tous moment, l’utilisateur peut se déconnecter.

## Besoins non fonctionnels

Le Chatbot doit être développé avec une capacité robuste à répondre de manière précise et impartiale à une gamme diversifiée de questions. Cette capacité devrait reposer sur l'utilisation de données vérifiées et de sources fiables, tout en évitant les biais. Il est également essentiel que le système soit adaptable à différentes langues dans la mesure du possible. De plus, la capacité du système à prendre en compte le contexte de la question ainsi que les informations connexes est cruciale pour fournir des réponses pertinentes.

## Méthodologie et formalismes adoptés



### Méthode SCRUM



SCRUM est un cadre de travail agile populaire utilisé dans le développement logiciel, mais également dans d'autres domaines où la gestion de projet nécessite flexibilité et adaptation. Voici quelques-unes de ses caractéristiques principales :

* Itératif et Incrémental
* Rôles Définis
* Réunions Flexibilité
* Transparence et Inspection
* Auto-organisation

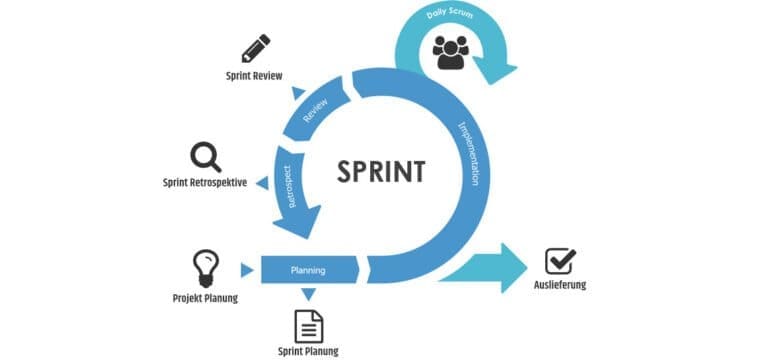


Figure 1: méthode SCRUM

### Outils de gestion

**GitHub** est une plateforme de développement collaboratif utilisée pour héberger des projets, gérer les versions du code source et faciliter la collaboration entre les développeurs.

* **Gestion des ressources :** Avec GitHub, nous avons réussi à garantir une collaboration fluide et bien organisée entre les membres de l’équipe, en plus de simplifier le partage des ressources dans notre référentiel de projet.

**Figure à ajouter**

* **Gestion des Sprints :** GitHub a été précieux pour la gestion efficace de nos Sprints. Nous avons exploité une structure de tableau pour ordonner les Sprints selon leur avancement (à faire, en cours, en prévisualisation, terminé), assurant ainsi une transparence totale sur le progrès de nos tâches.

**Figure à ajouter**

* **Gestion des événements :** GitHub a été in outil essentiel pour structurer, planifier et gérer les événements tout au long de notre projet. Nous avons utilisé un tableau pour organiser les différents événements tels que les réunions de planification de Sprint, les revues de Sprint et les rétrospectives de Sprint. Chaque événement était défini avec son sujet, son état (réalisé ou à faire), sa date et son type.

**Figure à ajouter**

### Diagramme de Gantt

L'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet, couramment utilisé en gestion de projet.

On a fait une planification avec Gantt :

**Figure à ajouter**

## Conclusion

Ce chapitre a permis de situer le contexte général du projet et d’expliquer la vision stratégique à laquelle il adhère. A ce niveau, il faut s’interroger comme s'est passé avec les projets dont le sujet est similaire.

**C**

# Chapitre II : L’état de l’art

**Chapitre II**



L’état de l’art représente un ensemble de connaissances et d’évolutions récentes dans le domaine dans lequel nous souhaitons nous immerger. Pour notre projet, des connaissances à jour sont cruciales car elles nous permettent de nous appuyer sur les meilleures pratiques. En incluant des analyses dans notre rapport de projet, nous visons à démontrer la rigueur scientifique de la démarche et à justifier les choix opérés. Cela nous permet également de relier notre projet à des œuvres existantes et de mettre en valeur son apport original. En fin de compte, la maîtrise des dernières technologies est un gage de qualité et d'adéquation à tout projet.

## Introduction

## LLAMA 2 : Des LLMs open-source performants et sécurisés

* **Pré-entraînement :** LLAMA 2 est pré-entraîné sur un corpus massif de données textuelles publiques, sans données de Meta. Des améliorations architecturales ont été apportées pour augmenter la longueur du contexte et la performance.
* **Affinement :** LLAMA 2-Chat est affiné en utilisant l'apprentissage supervisé (SFT) et l'apprentissage par renforcement avec retour humain (RLHF) pour optimiser les dialogues. L'accent est mis sur la qualité des données d'annotation.
* **Sécurité :** Des analyses approfondies ont été menées pour identifier et atténuer les biais et les risques potentiels liés à la toxicité, la véracité et les biais sociaux. Des techniques d'affinement spécifiques à la sécurité et des évaluations humaines ont été utilisés pour améliorer la sécurité des modèles.
* **Observations :** Le document présente des observations intéressantes sur l'émergence de l'utilisation d'outils et la perception temporelle dans LLAMA 2-Chat. [1]

Cet article est extrêmement pertinent pour notre projet de Chatbot impartial sur la Palestine car il fournit un modèle de base solide (LLAMA-2-Chat) avec ses performances et sa sécurité, pourrait servir de base solide pour notre Chatbot. Nous pourrions l'affiner d’avantage avec des données spécifiques à la Palestine pour améliorer sa compréhension du sujet. Il partage des techniques d'atténuation des biais, détaille les différentes approches utilisées par Meta AI pour atténuer les biais. Et il souligne l'importance de la qualité des données et met l'accent sur la nécessité de données d'annotation de haute qualité pour l'affinement.

## Introduction au développement de Chatbots

Le livre commence par une introduction aux chatbots, leur popularité croissante et leur impact sur divers secteurs. Il souligne l'importance du NLP pour comprendre le langage naturel et répondre aux requêtes des utilisateurs. Le livre détaille les étapes de création, d'entraînement et de déploiement de chatbots sur Facebook Messenger et Slack comme exemple. [2]

Ce livre fournit une base solide en NLP et L’apprentissage automatique appliqué aux chatbots, ce qui est essentiel pour comprendre les défis et les solutions potentielles pour éliminer les biais. Les techniques de NPL décrites dans le livre, comme l'analyse de sentiment et la détection de biais, peuvent être intégrées au chatbot pour identifier et atténuer les biais dans les réponses.

## Les Chatbots modernes : Revue technique

L'article souligne des éléments clés de l'architecture des Chatbots modernes :

* Les Chatbots peuvent être à domaine ouvert (connaissances générales) ou à domaine fermé (connaissances spécifiques).
* Les méthodes de récupération et de génération sont souvent combinées dans une approche hybride.
* Les word embeddings sont devenus essentiels pour représenter le sens des mots et les relations sémantiques.
* Les réseaux de neurones sont omniprésents dans le traitement du langage naturel et la génération de réponses.
* L'évaluation de la qualité des réponses reste un défi, l'évaluation humaine étant toujours cruciale.[3]

Cet article souligne la difficulté d'évaluer la qualité des réponses et les biais potentiels introduits par l'évaluation humaine. Cela nous encourage à réfléchir aux métriques d'évaluation pour le Chatbot et à la manière de minimiser les biais.

## Développement d’un Chatbot pour l'éducation en machine Learning

* **Combinaison de technologies :** Le Chatbot combine des modèles NLP avancés (Llama7B, GPT-2) avec des techniques (TF-IDF, similarité cosinus) pour une meilleure performance.
* **Jeu de données :** Le Chatbot est entraîné sur un ensemble de données diversifié provenant de diverses sources, couvrant le machine learning, les mathématiques et les statistiques.
* **Traitement du langage naturel :** Des techniques de prétraitement de texte, comme la lemmatisation et la suppression des mots vides, sont utilisées pour améliorer la précision des réponses.
* **Interface utilisateur :** L'interface utilisateur est conçue pour être simple et intuitive, avec des options de saisie textuelle et vocale.
* **Évaluation des performances :** Différentes mesures, comme la similarité BERT et les scores ROUGE, sont utilisées pour évaluer les performances des différents modèles.[4]

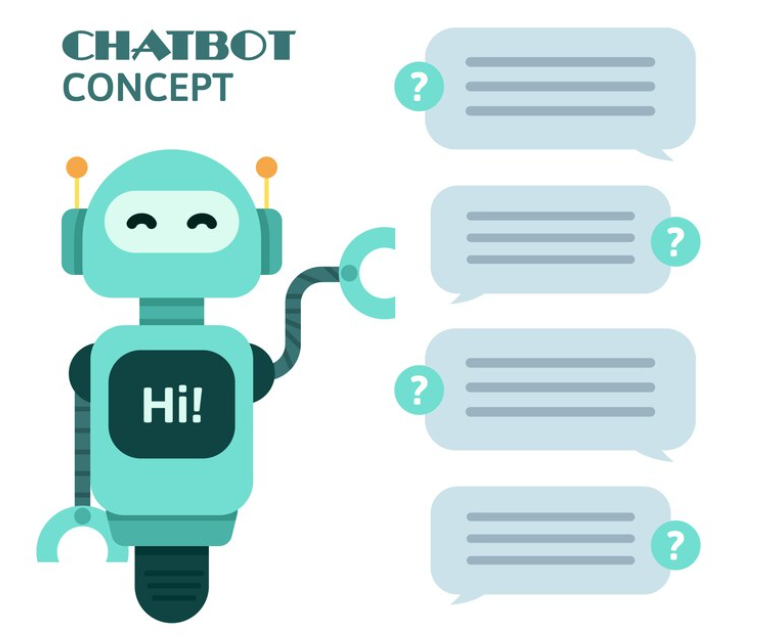
Les mesures d'évaluation utilisées dans cet article, comme la similarité BERT et les scores ROUGE, peuvent être appliquées à notre Chatbot pour évaluer son impartialité et sa précision. Le document souligne aussi l'importance d'une interface utilisateur intuitive et accessible, ce qui est crucial pour un Chatbot public comme le nôtre. Et malgré que l'article ne se concentre pas spécifiquement sur la gestion des biais, il met en évidence l'importance de la diversité des sources de données et des modèles pour fournir des réponses plus complètes et impartiales.

## Conclusion

Dans ce chapitre, une idée a été présentée concernant l'architecture à adopter afin de repérer les lacunes et les opportunités, et d'éviter de reproduire les erreurs du passé dans notre projet, comme nous l'avons mentionné précédemment dans les articles. En outre, nous avons exposé la solution envisagée ainsi que les diverses méthodes à mettre en œuvre.

# Chapitre III : Etude préalable du projet

**Chapitre III**



Dans ce chapitre, nous abordons les principes fondamentaux de notre sujet en explorant son contexte et ses bases. Notre objectif est de saisir les concepts clés qui constituent la base de notre travail, tout en établissant clairement le cadre dans lequel notre projet s'inscrit**.**

## Introduction

## Généralités sur LLM



### Définition

Les LLM font partie des modèles utilisés en AI, plus précisément dans le domaine du traitement du langage naturel (NLP). Les LLM sont des modèles d'apprentissage automatique qui sont spécifiquement conçus pour comprendre et générer du texte en se basant sur des exemples et des modèles statistiques appris à partir de grandes quantités de données textuelles.

### Les principes de base

* **Apprentissage supervisé :** Les LLM sont entraînés sur d'énormes corpus de texte dans un processus appelé apprentissage supervisé. Ils apprennent à prédire un mot ou une phrase suivante en se basant sur les mots ou phrases précédents.
* **Compréhension du contexte :** Les LLM ont la capacité de comprendre le contexte d'une phrase ou d'un paragraphe. Ils peuvent prédire non seulement le mot suivant dans une phrase, mais aussi comprendre l'intention derrière cette phrase.
* **Apprentissage automatique :** Les LLM sont des exemples d'apprentissage automatique, où ils apprennent à partir des données sans être explicitement programmés pour cela. Ils s'améliorent avec le temps et avec plus de données.
* **Apprentissage profond** : Les LLM font partie du domaine de l'apprentissage profond, un sous-domaine de l'IA, qui se concentre sur les réseaux neuronaux à plusieurs couches. Ces modèles utilisent des techniques d'apprentissage profond pour traiter et comprendre le langage humain.
* **Transformer Architecture :** Les modèles LLM modernes, tels que GPT, utilisent une architecture appelée Transformer. Cette architecture permet aux modèles de mieux comprendre le contexte des mots en tenant compte de tous les autres mots de la phrase, pas seulement des mots précédents.

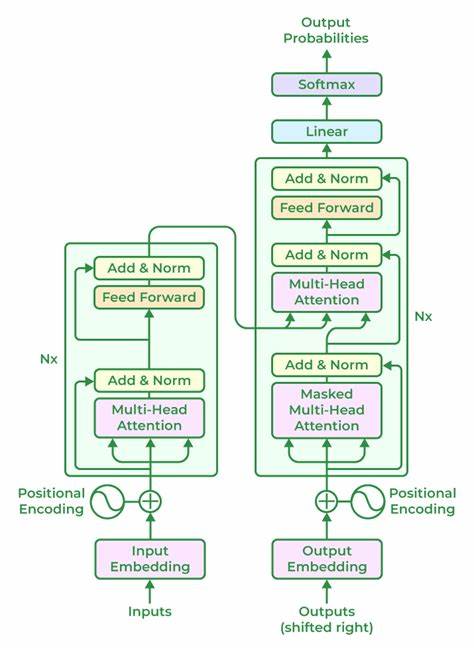


Figure 2: Architecture Transformers

### Défis et limites

Bien que les LLM. Aient fait des progrès impressionnants, mais n'oubliez pas que ces modèles ne comprennent pas vraiment le langage de la manière dont les humains le font, ils font des prédictions basées sur les modèles qu'ils ont appris lors de leur entraînement. Ils sont encore confrontés à des défis et à des limites importantes :

* **Comprendre le contexte** **:** LLM ne comprend pas toujours correctement le contexte, ce qui peut conduire à des réponses inappropriées ou incorrectes.
* **Biais :** les réponses générées peuvent refléter des biais présents dans les données de formation, tels que le sexe, la race, la géographie et la culture.
* **Qualité des données de formation :** la précision du LLM dépend fortement de la qualité et de la quantité des données de formation utilisées.

## Fine tuning

Le réglage fin (ou "fine tuning") d'un LLM fait référence au processus d'adaptation d'un modèle pré-entraîné à une tâche spécifique. Dans le contexte de l'apprentissage automatique, le réglage fin consiste généralement à prendre un modèle qui a été pré-entraîné sur une grande quantité de données (par exemple, l'ensemble du contenu de Wikipédia), et à l'entraîner davantage sur un ensemble de données spécifique à une tâche (par exemple, des critiques de films pour une tâche d’analyse des sentiments).

Le but de ce processus est de bénéficier à la fois de la capacité du modèle à comprendre le langage humain à un niveau général (grâce à l'entraînement initial sur une grande quantité de données) et à un niveau spécifique à la tâche (grâce à l'entraînement supplémentaire sur l'ensemble de données spécifique à la tâche).

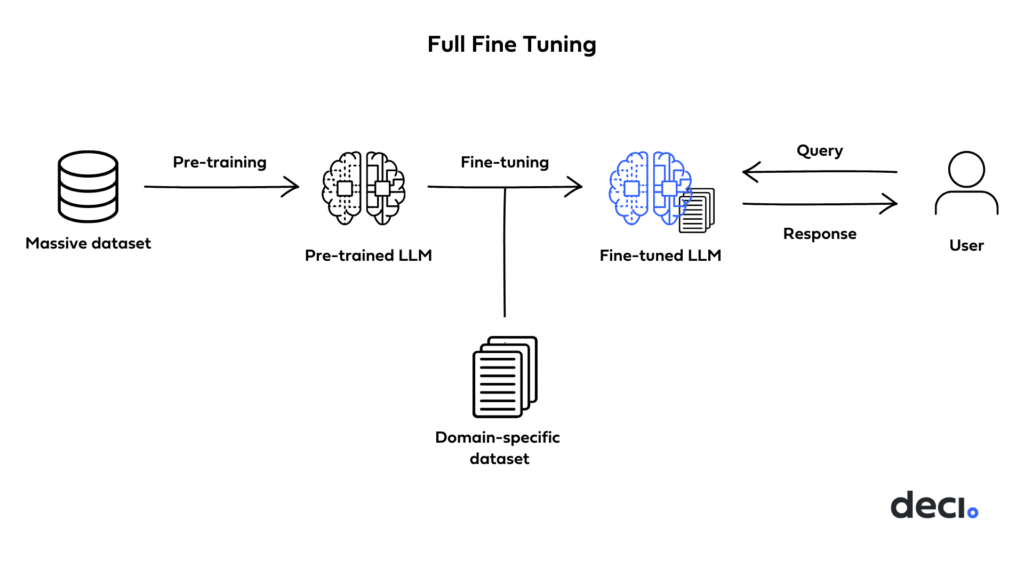


Figure 3: Fine tuning process

## Généralités sur LLAMA-2

### Introduction

Un modèle fondamental de langage de pointe conçu pour aider les chercheurs à faire progresser leurs travaux dans ce sous-domaine de l’IA. Des modèles plus petits et plus performants tels que LLAMA permettent à d’autres membres de la communauté de recherche qui n’ont pas accès à de grandes quantités d’infrastructures d’étudier ces modèles, démocratisant ainsi davantage l’accès dans ce domaine important et en évolution rapide.

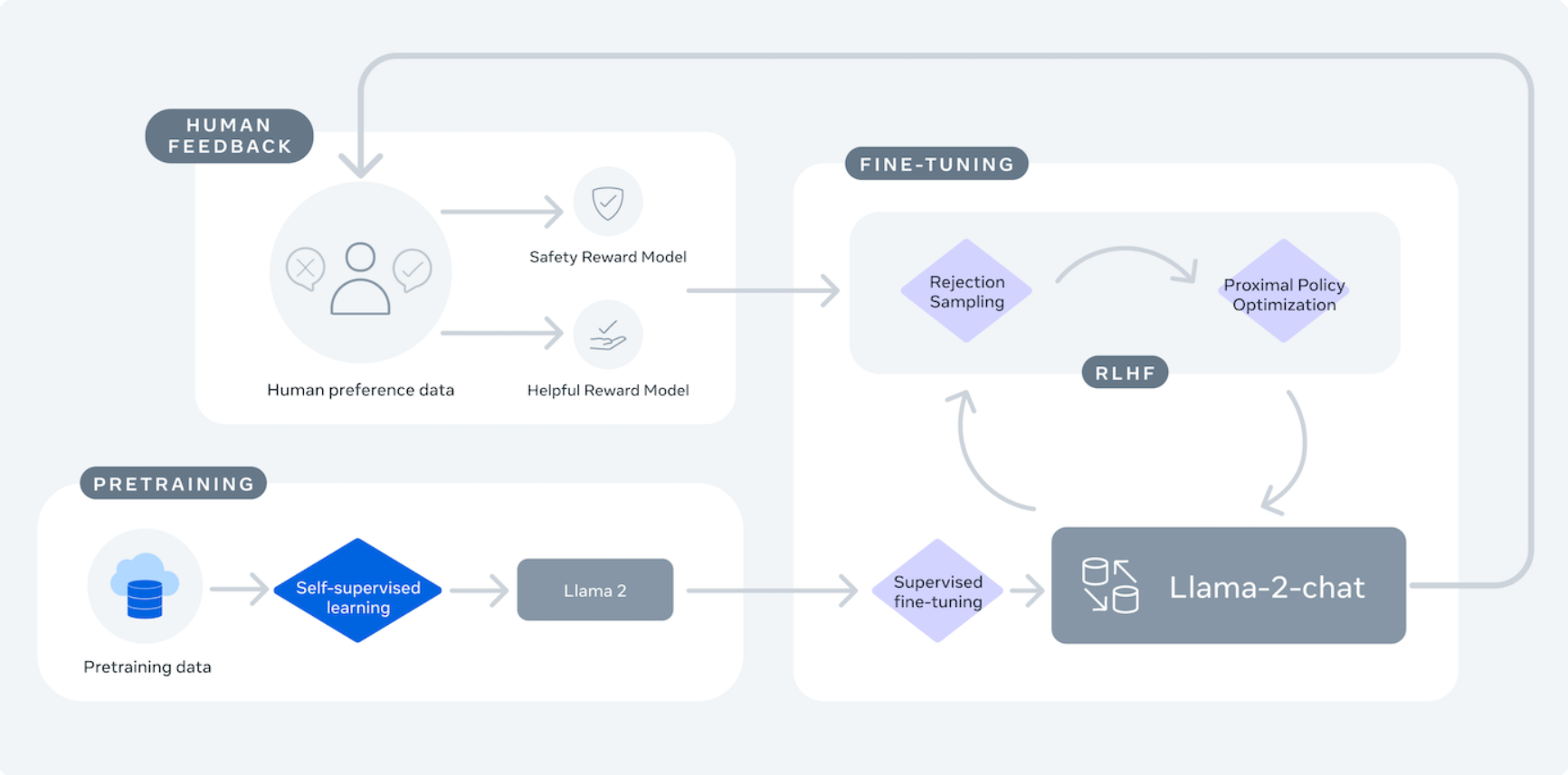


Figure 4: L'entraînement de LLAMA-2-chat

### Comparaison de LLAMA-2 avec GPT-3.5

S'il existe généralement des différences mineures entre les modèles linguistiques du LLAMA 2 et du GPT-3.5 d'OpenAI, il y a aussi des différences majeures qui priment sur les autres en fonction de l'utilisation que tu comptes en faire.

* Les données entraînées du modèle linguistique GPT-3.5 sont plus élevées que celles de LLAMA 2.
* La taille des paramètres du modèle linguistique GPT-3.5 est plus importante que celle de LLAMA 2.
* Le modèle linguistique LLAMA 2 génère des résultats plus sûrs que GPT-3.5.
* Le modèle linguistique LLAMA 2 offre des données plus récentes que le modèle GPT-3.5 d'OpenAI.
* GPT-3.5 est plus accessible que LLAMA 2.
* La performance de LLAMA 2 est supérieure à celle de ChatGPT. En évaluant les distinctions entre LLAMA 2 et GPT-4, il est évident que ces deux Grands Modèles Linguistiques apportent des avancées significatives dans le domaine de l’intelligence artificielle. La nature open-source de LLAMA 2 présente une voie prometteuse pour des améliorations portées par la communauté, le rendant ainsi un choix efficace et adaptable.

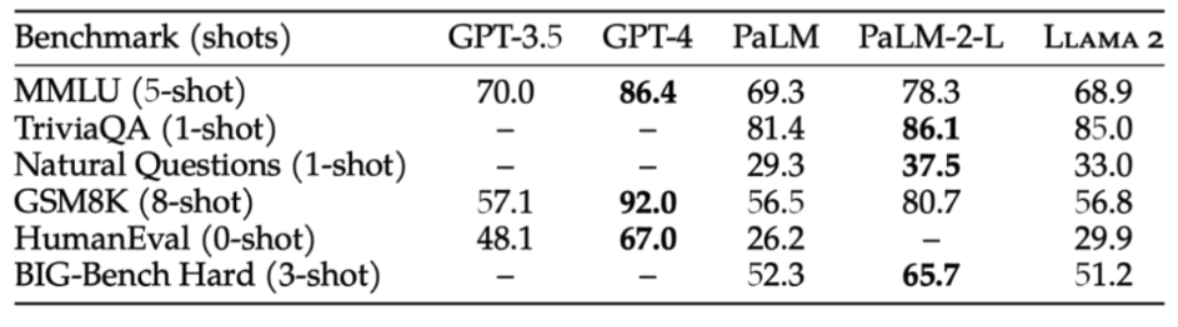


Figure 5: Benchmark des performances des LLMs

Il est possible de voir que le modèle de langage LLAMA 2 surpasse le modèle de langage GPT-3.5 dans la plupart des tests de référence. Par exemple, dans les tests de sécurité utilisant l'invite 2000, le modèle de langage LLAMA 2 a généré un résultat beaucoup plus sûr.

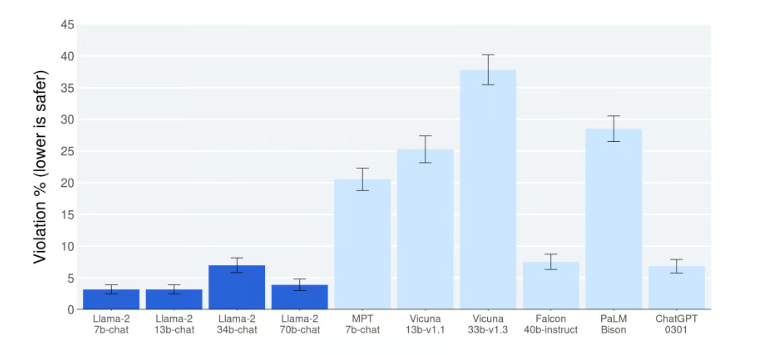


Figure 6: Benchmark de la sécurité des LLMs

## Chatbot

Au niveau le plus fondamental, un Chatbot est un programme informatique qui simule et traite une conversation humaine (écrite ou parlée), permettant aux humains d’interagir avec des terminaux digitaux comme s’ils communiquaient avec une personne réelle. Les Chatbots peuvent être aussi simples que des programmes rudimentaires répondant à une requête simple avec une réponse sur une seule ligne, ou aussi sophistiqués que des assistants digitaux qui apprennent et évoluent pour fournir des niveaux de personnalisation croissants à mesure qu’ils collectent et traitent des informations.

## Dataset

Un DataSet est une collection de données structurées associées à un domaine, une thématique ou un secteur d’activité spécifique. Les datasets peuvent contenir différents types d’informations, comme des chiffres, du texte, des images, des vidéos et même des fichiers audios. Ils peuvent se présenter sous divers formats comme CSV, JSON ou SQL.

Les datasets jouent un rôle fondamental dans de nombreux domaines, comme la [data science](https://www.intelligence-artificielle-school.com/cest-quoi-la-data-science/), la recherche, le Machine Learning. Ils permettent, entre autres, de découvrir des tendances, d’extraire des connaissances et de créer des modèles prédictifs.

## Conclusion

Ce chapitre a posé les bases de notre projet, en voulant comprendre le contexte et en analyse les concepts sous-jacents. Nous avons cadré notre travail, ce qui permettra à notre démarche d’avancer et vont maintenant pouvoir passer au chapitre suivant qui couvrira la conception du Chatbot.



# Chapitre IV : Conception du Chatbot

**Chapitre IV**

La plupart des utilisateurs ne sont pas des experts en informatique, ce qui souligne l’importance de fournir un moyen simple pour exprimer leurs besoins. C’est là que conception interviennent. Dans ce chapitre, nous présenterons la modélisation de notre future application web. Nous illustrerons une vue globale de la solution à l’aide d’un diagramme de cas d’utilisation, une vue statique avec le diagramme de classes, ainsi qu’une vue dynamique à travers des diagrammes de séquence.

## Introduction

## Définition de l’UML :

UML a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

## Conception du Chatbot

Cette partie vise à affiner la description de processus en détaillant les aspects de sa mise en œuvre, pour parvenir à une représentation très proche de notre application. Nous allons utiliser des diagrammes de cas d’utilisation, de classes et de séquence pour modéliser toute son architecture.

### Diagramme de cas d’utilisation

Le but du diagramme de cas d’utilisation est de définir ce que chaque acteur attend de système, à partir d’une représentation de l’interaction entre l’acteur et le système. Il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

* **Identification des acteurs :** L’acteur principal et le seul acteur dans notre système est :

**L’utilisateur :** c’est la personne qui interagit avec notre Chatbot pour poser des questions, recevoir des réponses, effectuer des actions spécifiques.

* **Identification des cas d’utilisation :** Notre acteur principal a la possibilité d’accomplir plusieurs actions au sein de système :
* Connexion : Permet à l’utilisateur de se connecter à son compte existant en fournissant ses identifiants.
* Inscription : Permet à un nouvel utilisateur de créer un compte.
* Saisie les informations d’inscription : Pendant le processus d’inscription, l’utilisateur saisit les informations requises telles que son nom, son email et mot de passe.
* Confirmer email : Après l’inscription, l’utilisateur doit confirmer son adresse email en cliquant sur un lien de confirmation envoyé par email.
* Accéder à la page principale : Une fois connecté, l’utilisateur est dirigé vers la page principale du Chatbot où il peut accéder aux différentes fonctionnalités.
* Saisir un nouveau message : l’utilisateur peut saisir un nouveau message pour interagir avec le Chatbot.
* Voir l’historique des conversations : l’utilisateur peut consulter l’historique de ses conversations passées avec le Chatbot.
* Ouvrir une nouvelle conversation : lorsque l’utilisateur démarre une nouvelle interaction avec le Chatbot. L’utilisateur initie la conversation en posant une requête spécifique.
* Archiver conversation : Permet l’utilisateur d’archiver une conversation déjà présente dans la liste des conversations des derniers mois.
* Déconnexion : Permet à l’utilisateur de se déconnecter de son compte.

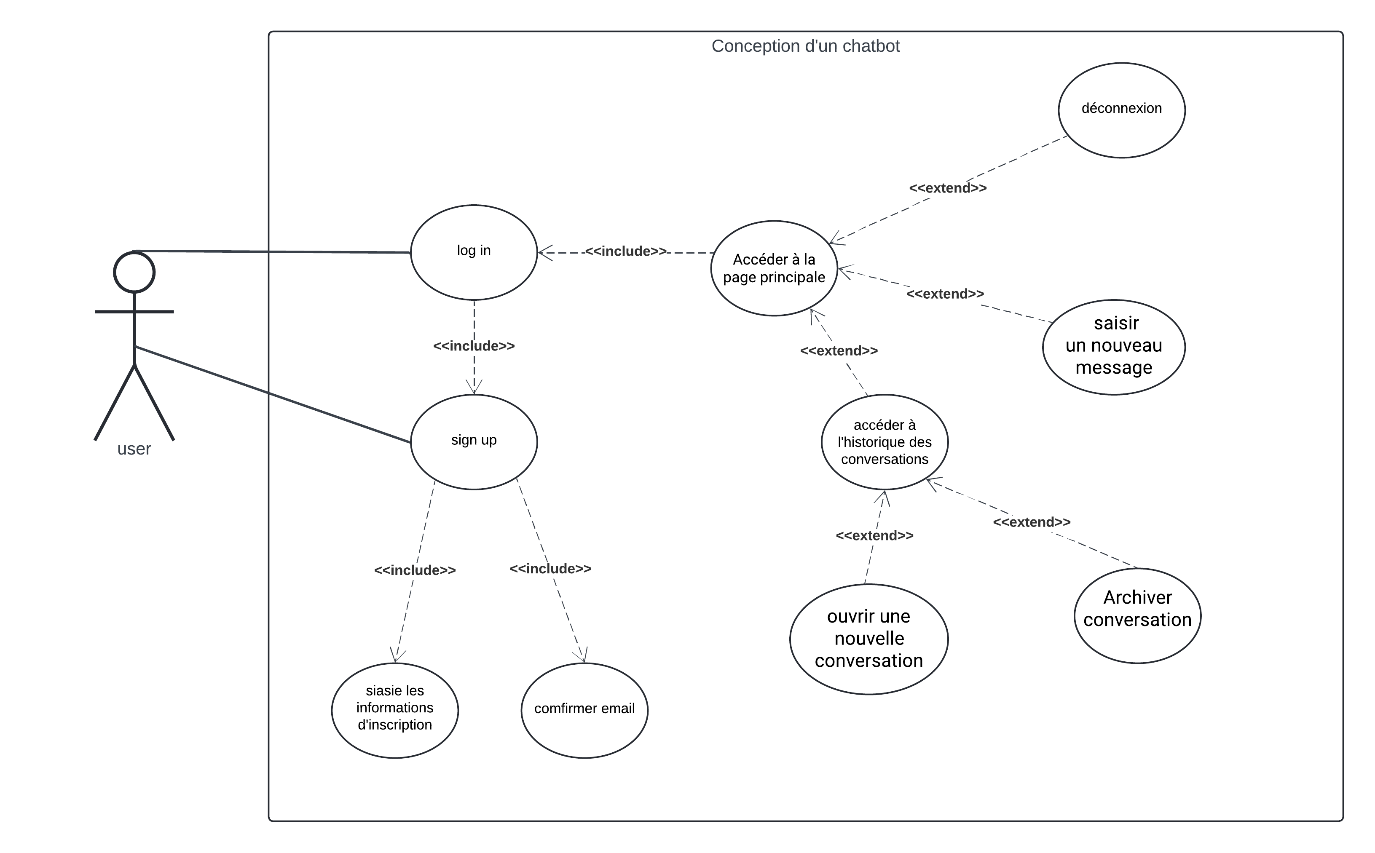


Figure 7: Diagramme de cas d'utilisation

### Diagrammes de séquence

Les diagrammes de séquence sont conçus pour représenter graphiquement les interactions entre les objets dans l’ordre chronologique. Cette représentation est souvent basée sur des cas d’utilisation détaillant divers scénarios et permet de visualiser les interactions entre les acteurs et le système. Le diagramme de séquence montre comment le système fonctionne en temps réel. Il illustre les échanges entre les objets dans l’ordre chronologique, en se basant sur les différents cas d’utilisation et leurs scénarios associés.

* **Connexion**

Ce diagramme de séquence représente le processus de connexion d’un utilisateur au système. Il détaille les étapes tell que la saisie des identifiants de connexion.

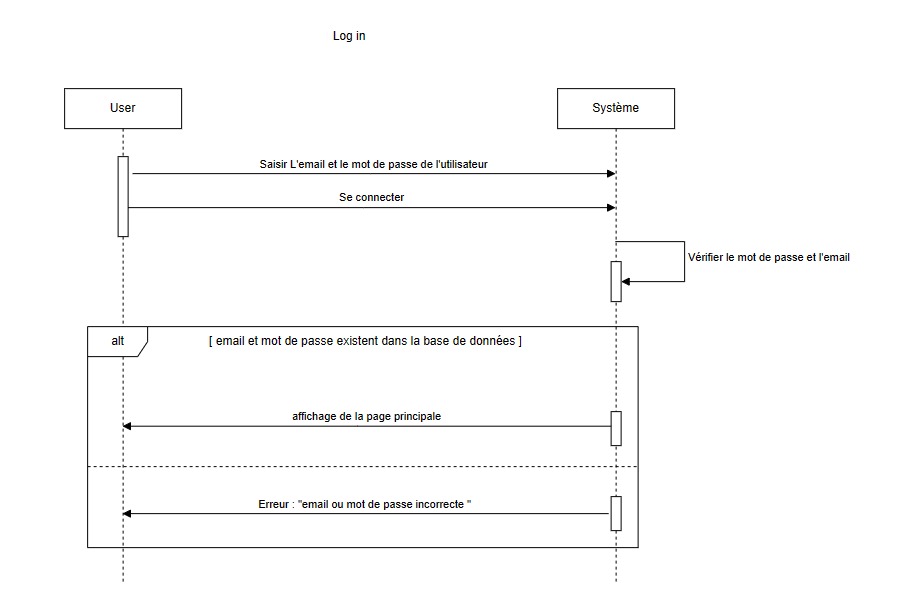


Figure 8: Diagramme de séquence - Connexion

* **Inscription**

Ce diagramme décrit le déroulement de l’inscription d’un nouvel utilisateur au système

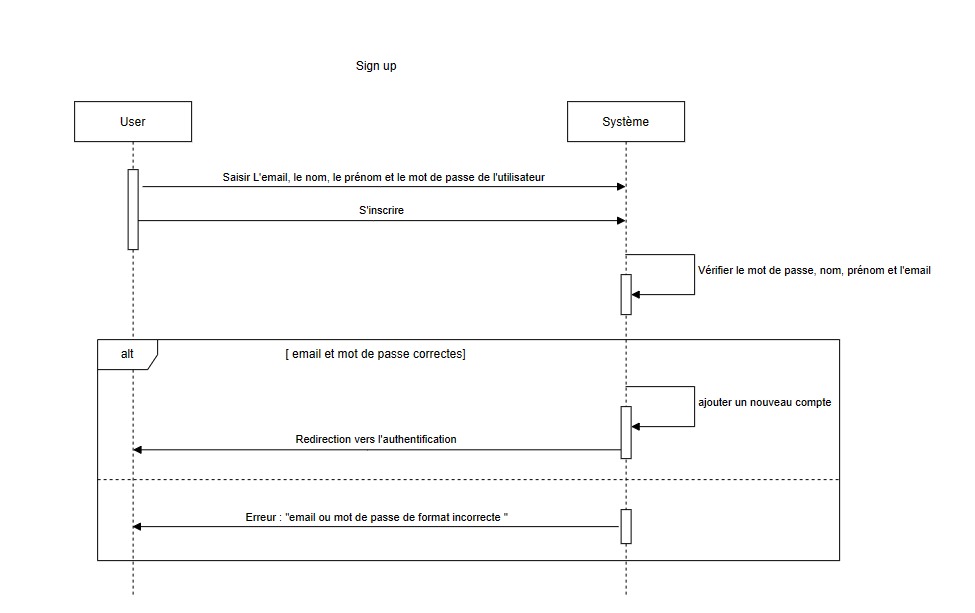


Figure 9: Diagramme de séquence - Inscription

* **Accès à l’historique des conversations**

Il décrit le processus selon lequel un utilisateur accède et navigue dans les conversations archivées

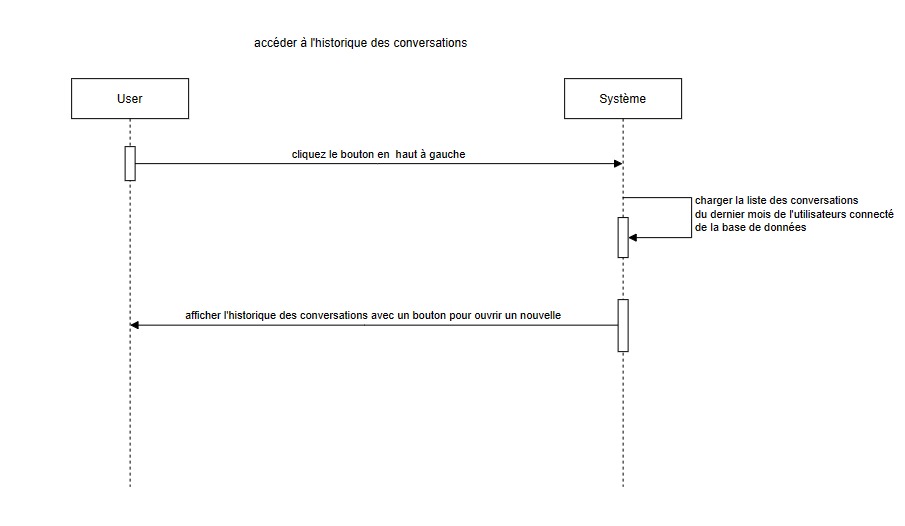


Figure 10: Diagramme de séquence - Accéder à l'historique

* **Saisir un nouveau message**

Il explique comment le système va traiter un nouveau message au niveau de l’interface, la base de données et LLM



Figure 11: Diagramme de séquence - saisie d'un message

### Diagramme de classes

Le diagramme de classes est un schéma utilisé dans l’ingénierie logicielle pour présenter les classes d’un système ainsi que leurs relations, qui appartient à la partie statique de l’UML, ne traite pas des aspects temporels ou dynamique. Chaque classe décrit les responsabilités, le comportement et le type d’un ensemble d’objets. Il permet de donner une vue statique du système en termes de classes d’objets et des relations entre classes.

* **Dictionnaire de données :** Les diagrammes de séquence étudié auparavant contient des données, pour ne pas perdre ces données on utilise un dictionnaire de données.

Voici notre dictionnaire de données :

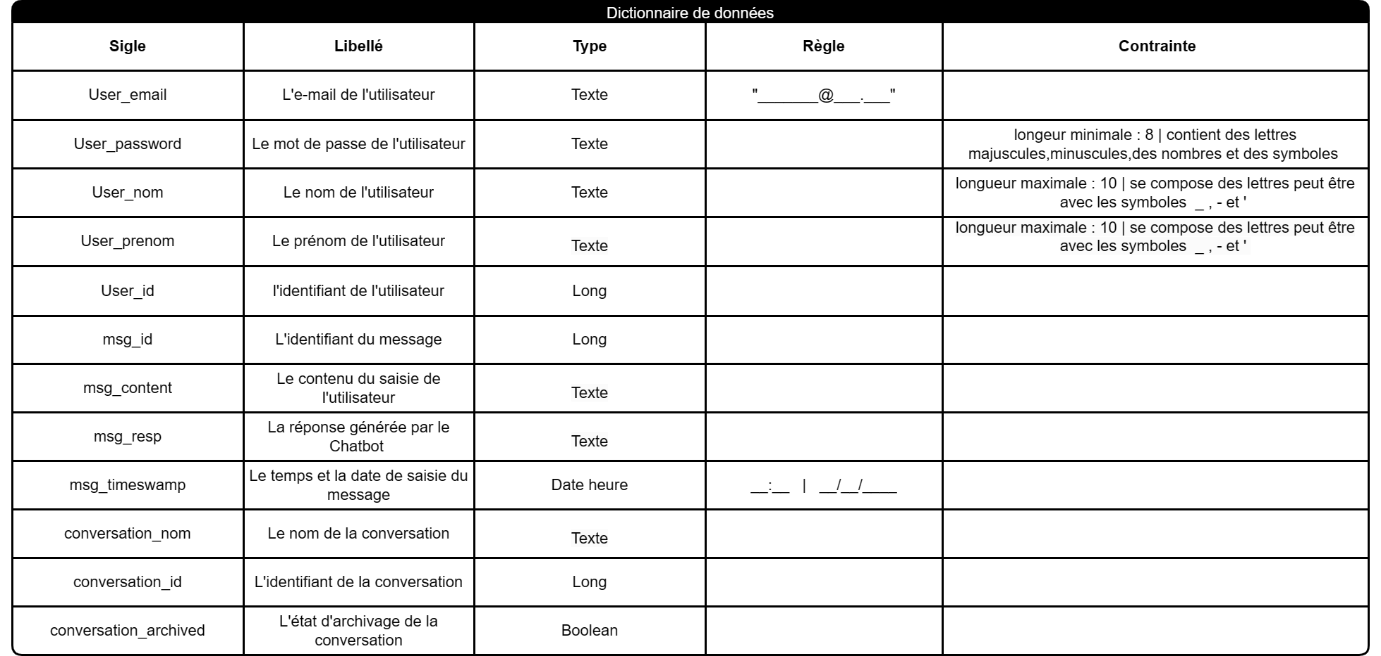


Figure 12: Dictionnaire de données

Les principales classes de notre application sont les suivants :

* User : contient toutes les informations relatives aux utilisateurs de l’application
* Massage : cette classe représente le message échangé entre l’utilisateur et le Chatbot.
* Conversation : cette classe représente une conversation entre l’utilisateur et le Chatbot. Elle peut contenir plusieurs messages échangés dans le cadre d’une interaction.

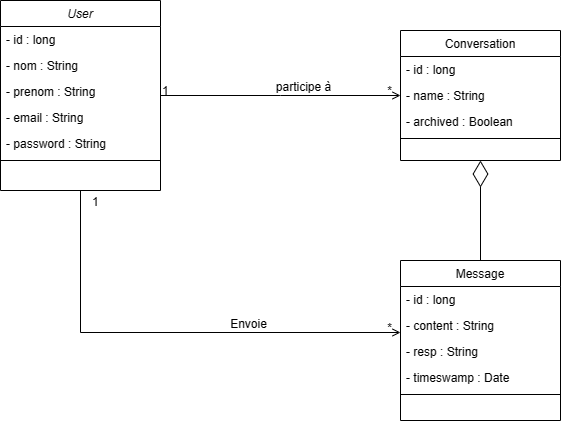


Figure 13: Diagramme de classes

## Conclusion

Dans cette partie, on a présenté l’analyse et la conception de notre système, notamment les interactions entre l’utilisateur et les différents composants du système à travers les cas d’utilisations et des scénarios pour mieux comprendre le déroulement de notre application. Cependant, Cette modélisation sera exploitée lors de la phase de réalisation.

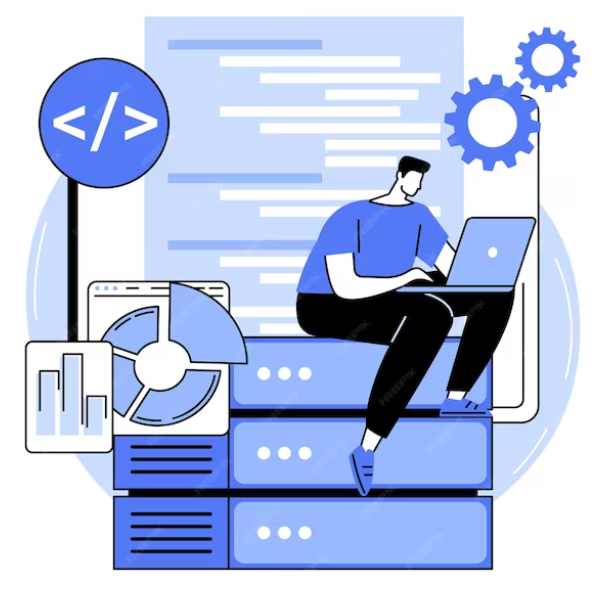
# Chapitre V : Réalisation du Chatbot

**Chapitre V**

**Chapitre V**

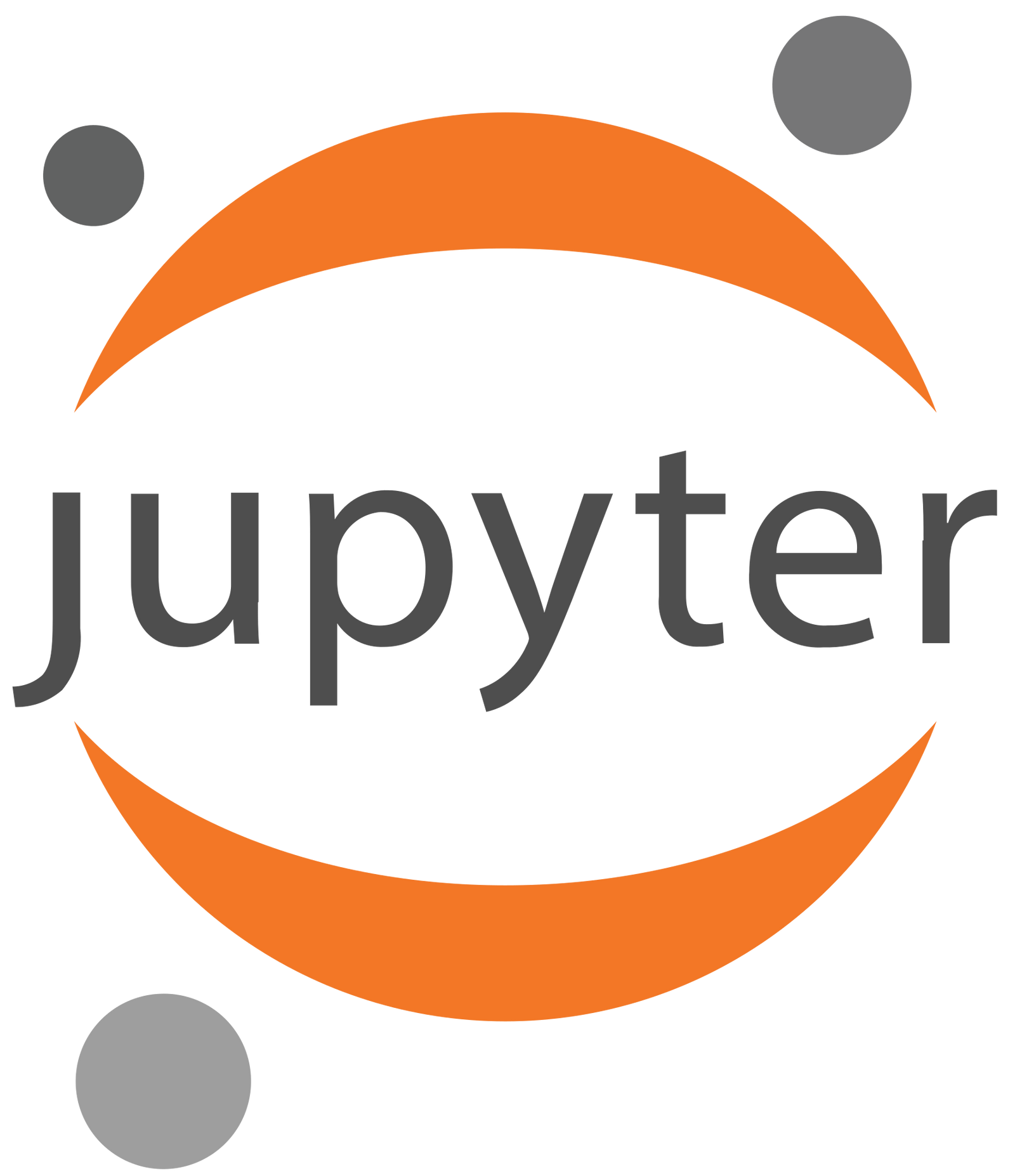
Ce chapitre de réalisation présente les différentes étapes nécessaires pour mener à bien notre projet. C'est la clôture de ce rapport, où nous détaillons les phases que nous avons suivies pour réaliser notre projet. Nous débuterons en présentant les outils et technologies utilisés. Ensuite, nous aborderons la préparation du dataset, le fine-tuning du modèle, l'évaluation de ses performances et enfin, le développement de l'interface utilisateur.

## Introduction



1. **Technologies et Outils Utilisés**
2. Les outils

## Jupyter Notebook

 Jupyter Notebook est une application Web open source apparue en 2015. L'outil, anciennement connu sous le nom d'IPython Notebook, est un environnement informatique interactif qui permet aux développeurs de présenter leur travail et d'interagir avec les autres membres du projet.

Par conséquent, Jupyter Notebook permet aux utilisateurs de créer des documents interactifs appelés blocs-notes. Ces documents peuvent contenir du code informatique, des commentaires, des équations mathématiques, des graphiques ou du contenu multimédia. L'application Jupyter permet également d'exporter des documents dans plusieurs formats (document unique, page web, fichier PDF, etc.).

## Kaggle Platform

Kaggle est une plateforme Web regroupant la plus grande communauté de science des données au monde, avec plus de 536 000 membres actifs dans 194 pays et recevant près de 150 000 soumissions chaque mois, fournissant des outils et des ressources puissants pour aider à tout ce qui concerne le progrès de la science des données. Kaggle fournit un environnement Jupyter Notebooks personnalisable et sans configuration. Il existe des GPU gratuits et une multitude de données et de codes publiés par la communauté. Dans Kaggle, vous trouverez tout le code et les données dont vous avez besoin pour mener à bien votre projet de science des données. Il existe plus de 50 000 ensembles de données publics et 400 000 blocs-notes publics accessibles à tous.

* **Visual studio code**

Visual Studio (VS) Code est un éditeur de code open-source principalement utilisé pour corriger et réparer les erreurs de codage des applications Cloud et web. VS Code est développé par Microsoft et prend en charge les systèmes d'exploitation MacOs, Linux et Windows. Les outils de VS Code peuvent être utilisés pour améliorer la fonctionnalité de tout code écrit.

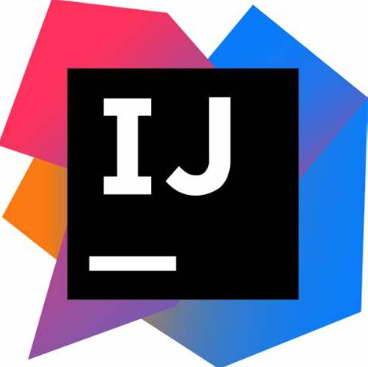
* **Postman**

Postman est un outil de développement d'API (interface de programmation d'application) qui permet de créer, de tester et de modifier des API. Presque toutes les fonctionnalités dont un développeur pourrait avoir besoin sont encapsulées dans cet outil. Il est utilisé par plus de 5 millions de développeurs chaque mois pour rendre leur développement d'API facile et simple. Il a la capacité de faire différents types de requêtes HTTP (GET, POST, PUT, PATCH), de sauvegarder des environnements pour une utilisation ultérieure, de convertir l'API en code pour différents langages (comme JavaScript, Python).

* **MySQL**

MySQL est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

* **IntelliJ IDEA**

 IntelliJ IDEA est un environnement de développement intégré (IDE) Java créé par JetBrains. En mettant l'accent sur la productivité et l'efficacité, IntelliJ IDEA propose une large gamme de fonctionnalités et d'outils pour aider les développeurs à écrire, tester et déboguer leur code plus rapidement et plus facilement. De la détection automatique des erreurs à la restructuration intelligente du code, IntelliJ IDEA est devenu un choix populaire parmi les programmeurs Java du monde entier

* **Figma**

Figma est un outil de conception graphique collaborative basé sur le cloud. Il peut fonctionner sur MacOS ou sur un PC exécutant Windows ou Linux. C'est un outil qui vous propose un plan gratuit où vous pouvez créer et stocker 3 projets actifs à la fois.

En fait, Figma est une application web d'édition graphique qui permet le partage en temps réel sur le même fichier, ce qui signifie que toutes les parties prenantes du projet peuvent interagir et collaborer sur le projet en tenant compte de chaque mise à jour et changement. Cela permet évidemment de gagner du temps et d’augmenter l’efficacité.

* **Hugging Face**



Hugging Face est une plateforme et une communauté qui fournissent des outils pour créer, former et déployer des modèles d'apprentissage automatique basés sur la technologie et le code open source. Elle offre également un espace permettant aux chercheurs, ingénieurs et passionnés d’IA de se réunir pour échanger des idées, obtenir du soutien et contribuer à des projets open source.

* **StarUML**

Logiciel de modélisation open source utilisé dans Unified Modeling Language où la modélisation de systèmes et de logiciels prend en charge les concepts UML en fournissant différents types de diagrammes UML pour générer du code dans différents langages afin de développer des plates-formes rapides et flexibles pour répondre aux besoins et aux exigences.

1. Technologies

* **Pythons**

Python est le langage de programmation informatique le plus populaire et le plus utilisé, notamment dans le domaine de la Data Science et du Machine Learning. De plus, Python est un langage multiplateforme qui fonctionne sur divers systèmes d'exploitation, tels que Windows, macOS et Linux, ce qui en fait un choix idéal pour les développeurs travaillant sur différents environnements.

* **HTML**

HTML signifie « HyperText Markup language » qu’on peut traduire par « langage de balise pour l’hypertexte ». Il est utilisé afin de créer et de représenter le contenu d’une page web et sa structure.

Nous avons utilisé ce langage pour élaborer la structure principale des pages de l’interface.

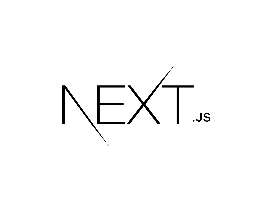
* **CSS**

 Le CSS « Cascading Style sheets » est un langage permettant de mettre en forme des pages HTML, grâce au CSS, il est possible de créer un design attractif pour un site web et de le rendre ‘responsive’, c’est-à-dire adaptable aux différents formats d’écran.

* **TypeScript**

TypeScript est un langage de programmation développée par Microsoft en 2012. Son ambition principale est d’améliorer la productivité d’applications complexe. C’est un langage open source, développé comme un sur-ensemble de JavaScript. Ce qu’il faut comprendre, c’est que tout code valide en JavaScript l’est également en TypeScript. Cependant, ce langage introduit des fonctionnalités optionnelles comme le typage ou la programmation orientée objet.

* **NEXT JS**

****

NEXT JS est un Framework pour le développement web basé sur React.il facilite la création d’application web rapides et performantes.

J’ai utilisé NEXT JS pour développer notre interface en exploitant les fonctionnalités offertes par ses composants.

* **Spring**

Spring Framework fournit un modèle complet de programmation et de configuration pour les applications d’entreprise modernes basées sur Java, sur tout type de plate-forme de déploiement.

Un élément clé de Spring est le support de l’infrastructure au niveau des applications : Spring se concentre sur la « plomberie » des applications d’entreprise afin que les équipes puissent se concentrer sur le logique métier au niveau des applications, sans liens inutiles avec des environnements de déploiement spécifiques.

* **Java**

Java est un langage de programmation originellement proposé par Sun Microsystems et maintenant par Oracle depuis son rachat de Sun Microsystems en 2010.

Java a été conçue avec deux objectifs principaux :

* Offrir un langage orienté objet avec une bibliothèque standard riche
* Permettre aux développeurs d’écrire des logiciels indépendants de l’environnement hardware d’exécution.
* **LangChaine**

LangChaine est un Framework open-source développé en python et agit comme une solution logicielle dont l’objectif est de standardiser et simplifier le développement d’applications web qui utilisent des LLM.

1. Préparation du DataSet

Dans cette partie, nous explorerons les étapes nécessaires pour préparer une DataSet à partir d’un groupe de fichier, des articles et des journaux, extraire le texte pertinent, formater l’ensemble de données pour qu’il soit compatible avec Llama2, puis le déployer sur Hugging Face. Ce processus est essentiel pour ceux qui souhaitent former ou fine tuner des Llm.

## Regroupement des données

### Identification des Sources de Données

Nous détaillerons comment identifier et collecter les différentes sources de données  pertinentes pour votre projet. Nos sources cibles comprennent :

* **Des livres soutenant la cause palestinienne**

Nous avons collecté des livres et des publications qui défendent la cause palestinienne et documenteront les injustices subies par les Palestiniens et les crimes commis par Israël.

**Source :** [مؤسسة الدراسات الفلسطينية | (palestine-studies.org)](https://www.palestine-studies.org/ar)

**Source :** [Z-Library est la plus grande bibliothèque électronique du monde. Libre accès au savoir et à la culture.](https://fr.z-library.se/)

* **Sites Pro-Palestiniens**

Nous avons identifié des sites web qui défendent la cause palestinienne et qui proposent des informations sur les entreprises soutenant l’occupation Israélienne, des articles et des informations en faveur des Palestiniens.

**Source:** [Who Profits - The Israeli Occupation Industry](https://www.whoprofits.org/)

* **Documents de Résistance Palestinienne**

Nous avons regroupé des documents et des ressources prévenant desgroupes de résistance palestiniens, tels quedes déclarations, des rapports et des manifestes.

**Source :** [الوثائق التاريخية للجبهة الشعبية لتحرير فلسطين – Historical Documents of the PFLP (Popular Front for the Liberation of Palestine) (pflp-documents.org)](https://pflp-documents.org/)

* **Des Documentaires sur la Vie en Palestine**

Des sites web et des plateformes ont été ajoutés pour présenter des documentaires qui mettent en évidence la vie difficile des Palestiniens sous occupation, mettant en évidence les réalités quotidiennes et les abus subis.

**Source :** [أفلام فلسطين (palestinefilms.org)](https://palestinefilms.org/)

**Source :** [فلسطين .. حكاية الأرض | الدحيح | Palestine .. a Tale of Land (youtube.com)](https://www.youtube.com/watch?v=f0oy-NicIgE&t=4s)

### Extraction des ressources

Les sites contenant des documents et des livres, ainsi que les journaux, Nous avons utilisé la technologie de web Scraping pour l’extraction du contenu.

Le web Scraping est une technique qui consiste à extraire automatiquement des données et informations sur le web. Cette méthode permet de collecter rapidement des données à partir de différents sites internet. En raison de cette praticité, le web scraping est adopté par de nombreuses organisations.

Le web Scraping nous a permis de récupérer automatiquement l'intégralité du contenu de ces sites, car nous étions convaincus qu'ils renfermaient des informations pertinentes pour nos besoins, et il utilise des bibliothèques importantes comme :

* **BeautifulSoup :**

BeautifulSoup fournit des méthodes simples pour naviguer, rechercher et modifier un arbre d’analyse dans des fichiers HTML ou XML. Il transforme un document HTML complexe en un arbre d’objets Python. Il convertit aussi automatiquement le document en **Unicode**, de sorte que vous n’avez pas à penser aux encodages. Cet outil vous aide non seulement à scraper, mais aussi à nettoyer les données. BeautifulSoup prend en charge l’analyseur HTML inclus dans la bibliothèque standard de Python, mais aussi plusieurs analyseurs Python tiers comme l’XML, Il permet généralement aux programmeuses d’économiser des heures ou des jours de travail.

* **Selenium**

Au départ, Selenium a été développé dans le but de tester le fonctionnement d'un site web, mais rapidement, la demande de web Scraping avec Selenium a connu une augmentation.

C'est un outil très populaire qui peut automatiser divers navigateurs tels que Chrome, Firefox, Opera et même Internet Explorer grâce à un middleware contrôlé appelé Selenium WebDriver.  
Le protocole WebDriver a été développé par l'organisation W3C pour automatiser les navigateurs. C'est principalement un service de middleware qui se trouve entre le client et le navigateur, transformant les commandes du client en actions du navigateur web.

### Organisation des ressources

Nous avons séparé les ressources de données en quatre parties :

* **Les journaux**

En raison de la structure homogène des journaux (auteur, titre, contenu), il est plus facile de les regrouper afin de faciliter l'extraction du texte que nous exposerons dans la section suivante.

* **Les fichiers et les livres copiables**

Les fichiers et livres copiables sont regroupés pour les différencier des autres qui ne le sont pas, ce qui facilite l'extraction de texte.

* **Les fichiers et les livres non copiables**

On peut accéder aux fichiers et livres qui ne peuvent pas être copiés en utilisant la reconnaissance optique de caractères (OCR) pour extraire le texte, ce qui répond à notre besoin de texte.

Nous avons utilisé l’OCR (online-convert), l’un des meilleur OCR que nous avons trouvé.

**Remarque :** lors de l’utilisation de l’OCR (online-convert), qui est l’un des meilleurs que nous avons trouvés. Cependant, nous avons rencontré un problème avec les livres en arabe, l’OCR ne parvient pas à rendre tout le texte copiable. Pour les parties qui deviennent copiables, le texte extrait n’a pas de sens, Ce qui nous empêche de travailler avec ces livres.

* **Les documentaires**

Dans le cas des documentaires, nous avons procédé à la transcription pour obtenir le texte nécessaire.

1. **structuration et nettoyage**

Afin d'extraire et de nettoyer le texte, nous avons employé un logiciel Python pour préparer les données pour notre DataSet.

1. Les Outils d'Extraction de texte.

* **Extraction du fichier texte (.Txt)**

Nous avons utilisé La méthode Read() en python renvoie une chaîne de caractères contenant les données lues à partir du fichier. Si la fin du fichier est atteinte avant que la quantité spécifiée de données ait été lue, la méthode renvoie une chaîne de longueur inférieure à la quantité demandée.

* **Extraction du PDF**

Nous avons utilisé PDFMiner, est une bibliothèque Python pure qui peut facilement extraire tous les textes d'un fichier PDF qui sont rendus par programme. La grande capacité est qu'il extrait également les emplacements correspondants, les noms et tailles de police et la direction d'écriture (horizontale ou verticale) pour chaque segment de texte.

* **Transcription des documentaires**

Pour avoir le texte que nous avions besoin du documentaire et les vidéos nous avons utilisé **veed.io** et **youtubetranscript,** ce qui nous a été très utile.

Après l’extraction du texte, nous l’avons traduit en anglais à l’aide de Doctranslator.

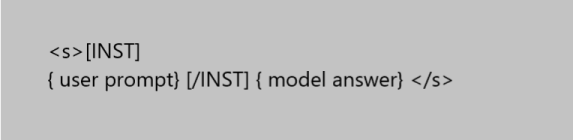
1. Nettoyage et Prétraitement du Texte

Pour la préparation de texte, nous avons éliminé les espaces superflus, ainsi que les filigranes (de l'OCR, du Doctranslator et des sites sources), les ponctuations, et nous avons converti le texte en caractères minuscules.

1. **Mise en Forme du DataSet pour LLama2**

Afin de réaliser un entraînement efficace ou bien un « fine-tuning » de Llama 2, il est essentiel de préparer notre ensemble de données dans un format qui est compatible avec Llama 2 et qui correspond au format d'entraînement par Meta.

Nous présentons un exemple de format de données à adopter par Llama 2 :



**‘<s>’ et ‘</s>’ : Il s'agit des tokens BOS (**Beginning Of Sentence) début de phrase. **Et EOS** (End Of Sentence) fin de phrase **de SentencePiece.**

Dans SentencePiece, ces marqueurs spéciaux sont utilisés pour délimiter le début et la fin des phrases individuelles. Lorsqu'il s'agit de conversations, ils servent à séparer les messages individuels, qu'il s'agisse de l'entrée de l'utilisateur ou de la réponse du modèle.

**[INST][/INST] :** Ces balises semblent avoir été spécialement conçues pour identifier les messages de l'utilisateur lors des conversations. Ces éléments permettent de faire une distinction nette entre les contributions de l'utilisateur et le reste du texte.

Pour appliquer cette structure à notre texte extrait précédemment, nous avons séparé le texte des journaux de celui des autres sources, (fichiers, livres). Pour chaque type de source, nous avons établi une logique particulière lors de notre processus de formatage de dataset :

* Pour les textes provenant des journaux, Le titre du journal est placé dans le champ {user prompt} tandis que le contenu est placé dans le champ {model answer}.
* Pour les autres sources de texte, nous avons utilisé une boucle pour diviser le texte en segments. La boucle a été mise en place de manière à séparer le texte en deux parties :

Une partie pour le champ {user prompt}, qui comprend entre 30 et 40 mots, et une autre partie pour le champ {model answer}, qui comprend entre 90 et 120 mots. Cette méthode nous permet de limiter la quantité de texte présente dans chaque ligne de notre ensemble de données, ce qui facilite l'apprentissage efficace du modèle.

1. **Déploiement sur Hugging Face**

Après le formatage du DataSet, nous avons utilisé dans notre code python la bibliothèque Pandas, qui est une bibliothèque Python puissante et open-source, spécialement développée pour la manipulation et l'analyse des données. Pandas est extrêmement prisé dans le domaine des sciences de données en raison de sa facilité à lire et écrire des données, en particulier à partir de fichiers CSV en utilisant les fonctions read\_csv() et to\_csv(). Par la suite, nous avons enregistré nos données sous forme d’un fichier CSV et les avons uploadées sur Hugging Face.

Finalement, notre DataSet sur Hugging Face comprend une seule colonne de texte et d'un nombre total de lignes de 23000.

Source :

1. Fine tuning

Dans cette partie nous allons entamer l’entrainement du modèle sur notre DataSet à l’aide de fine-tuning qui sert à spécialiser un modèle pré-entrainé sur une tâche spécifique.

L’un des meilleurs et les plus importantes techniques qu’il nous aide et facilite le fine tuning des Llm, et en particulier dans notre cas Llama2, est LoRA (Low-Rank Adaptation). L'équipe de Microsoft a développé LoRA qui est une technique novatrice pour améliorer les grands modèles de langage. Il s'agit de geler les poids du modèle pré-entraîné (Llama2) et de réaliser un fine-tuning à l'aide d'un petit modèle. LoRA permet de diminuer de manière significative le nombre de paramètres à entraîner tout en préservant, les performances du modèle. Il s'agit d'injecter de plus petites matrices entraînables dans chaque couche de l'architecture du LLM, plutôt que de modifier directement les poids originaux.

Aussi QLoRA :

Bien que LoRA aide à réduire les besoins de stockage, vous aurez toujours besoin d’un grand GPU pour charger le modèle dans la mémoire pour l’entraînement LoRA. C’est là que QLoRA, ou LoRA quantifié, entre en jeu. QLoRA est une combinaison de LoRA et de quantification. En générale, elle quantifie d’abord le LLM, puis effectuez un entraînement LoRA.

## ****Installation et importation des bibliothèques nécessaires****

**Les bibliothèques nécessaires seront installées en premier.**

Nous chargerons ensuite les modules requis à partir de ces bibliothèques.

Voici les bibliothèques utilisées, avec une simple définition à chacune :

* **torch :** une bibliothèque de base pour les calculs tensoriels et l'apprentissage profond.
* **peft :** Permet le fine-tuning des grands modèles de langage à l'aide de techniques LoRA. Il permet la compression des modèles et un fine-tuning rapide, en particulier sur les appareils à ressources limitées, en réduisant le nombre de paramètres entraînables.
* **bitsandbytes :** Fournit des techniques de quantification et de binarisation pour les réseaux neuronaux, contribuant à la compression des modèles. Il contribue à la compression des modèles, ce qui rend plus facile leur déploiement sur des appareils périphériques dotés d'une mémoire et d'une puissance de calcul limitées.
* **transformers :** simplifie le travail avec de grands modèles de langage, en proposant des modèles pré-entraînés et des pipelines d'entraînement.
* **trl :** Se concentre sur l'entraînement et l'optimisation efficaces des modèles, en particulier pour les Llms.
* **accelerate :** accélère l'entraînement et l'inférence sur différentes plateformes matérielles.
* **datasets :** simplifie le chargement et la préparation du DataSet pour les tâches d'apprentissage automatique.
* **pipeline :** rationalise l'utilisation de modèles pré-entraînés pour les tâches NLP courantes sans entraînement personnalisé.
* **pyarrow :** probablement utilisé pour un chargement et un traitement efficaces des données.
* **LoraConfig :** Contient les paramètres de configuration pour le fine-tuning basé sur LoRA.
* **SFTTrainer :** gère l'entraînement, l'optimisation et l'évaluation des modèles.

## ****Configuration et Paramètres du Fine-Tuning****

### ****Sélection du Modèle et du DataSet****

### ****Paramètres QLoRA****

### ****Configuration de BitsAndBytes****

### ****Paramètre des arguments d’entraînement****

### ****Configuration du SFTTrainer****

## ****Configuration et entraînement du modèle****

### 1. Chargement de DataSet

### 2. Configuration de QLoRA et chargement du modèle

### 3. Chargement du tokenizer Llama

### 4. Configuration de LoRA

### 5. Paramètres d'entraînement

### 6. Paramètres de fine-tuning supervisé

### 7. Entraînement du modèle

## Post-entraînement

### 1. Sauvegarde et mise en ligne du modèle entraîné

1. Sauvegarde du modèle entraîné

2. Mise en ligne du modèle sur la plateforme Hugging Face

### 2. Génération de texte avec le modèle entraîné

#### 1. Configuration de la génération de texte

#### 2. Exécution de la génération de texte avec le modèle entraîné

1. Evaluation du modèle

Pour évaluer notre modèle il existe plusieurs méthodes peuvent être utilisées pour évaluer son efficacité. Parmi ces méthodes (métrique), on trouve ROUGE avec c’est différent genre, METEOR et Perplexité. Nous avons utilisé ces trois métriques combinées pour obtenir une évaluation plus complète, voici une brève explication de chacune de ces métriques :

**ROUGE:** signifie Recall-Oriented Understudy for Gisting Evaluation. Il s’agit essentiellement d’un ensemble de métriques permettant d’évaluer le résumé automatique des textes ainsi que les traductions automatiques. Il fonctionne en comparant un résumé ou une traduction produite automatiquement à un ensemble de résumés de référence (généralement produits par humains).

**METEOR :** Incorpore le rappel, la précision et la correspondance sémantique supplémentaire basée sur les radicaux et la paraphrase. Tout simplement, il capture une certaine similitude sémantique.

**Perplexité :** C'est une mesure couramment utilisée pour évaluer la performance des modèles de langue. Elle quantifie la capacité du modèle à prédire un échantillon de texte. Des valeurs de perplexité plus basses indiquent une meilleure performance.

**Remarque :** Lors de l’évaluation nous avons testé Llama2 13B entraîné précédemment ainsi que Llama2 7B. Cependant, les résultats de test se sont avérés meilleurs pour Llama2 7B, nous avons donc décidé de prendre cette version en considération.

1. Développement de l'Interface du Chatbot

Lors du développement de cette application web pour le chatbot, nous avons mis tout notre intérêt à créer une interface utilisateur facile à naviguer et à comprendre, afin d’offrir une bonne expérience à l’utilisateur.

Nous avons nommé notre chatbot « Camel » en nous inspirant du nom du modèle de base (Llama2). Ce choix de nom s’explique par la présence de chameau dans notre région, par contre l’absence de lamas.

Pour concevoir l’apparence de notre interface, nous avons utilisé **Figma** qui est définie précédemment, ce qui nous a permis de collaborer efficacement pour unifier nos idées et opinions.

## Développement Front-End

1. Page d’accueil (Landing page)

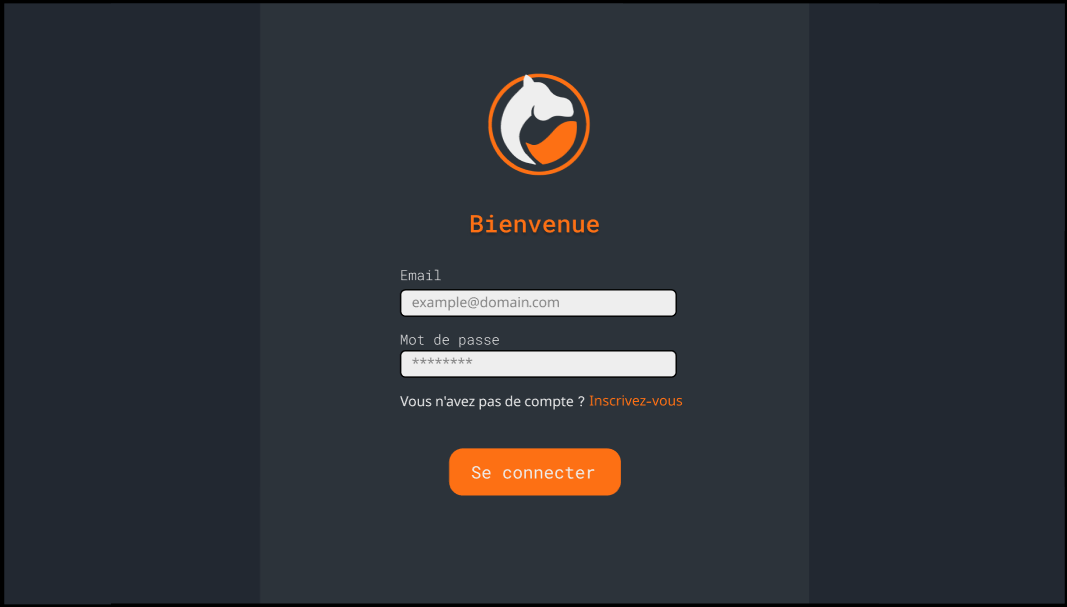
C’est la première page que l’utilisateur voit lors de son accès à notre chatbot, elle lui permet de se connecter à son compte si celui existe déjà, sinon il va créer un nouveau. De plus, elle permet à l’utilisateur de découvrir l’objectif de notre application web en cliquant sur « à propos ».

1. À propos

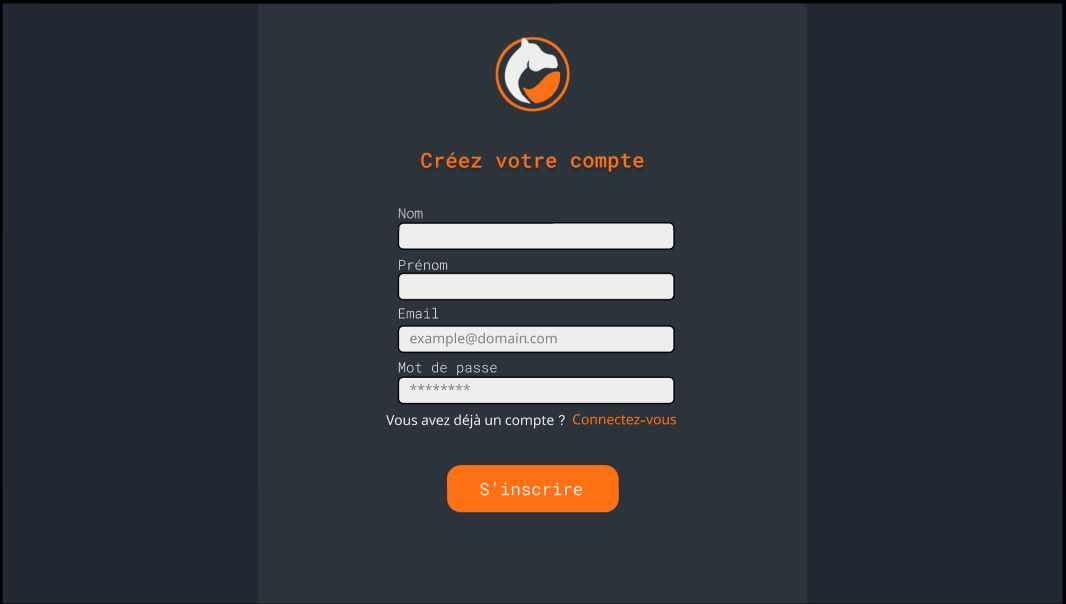


Cette page permet de Découvrir notre projet « Camel », notre mission et nos valeurs. Pour savoir plus sur notre projet de chatbot innovant basé sur le modèle Llama2.

1. Page de connexion (log in page)



Concernant cette page, elle permet à l’utilisateur de se connecter à son compte avec son adresse email et son mot de passe. Si l’utilisateur n’a pas de compte, il peut cliquer sur « Inscrivez-vous », ce qui le redirigera vers la page d’inscription, si il entre un mot de passe incorrect un message d’erreur s’affichera.

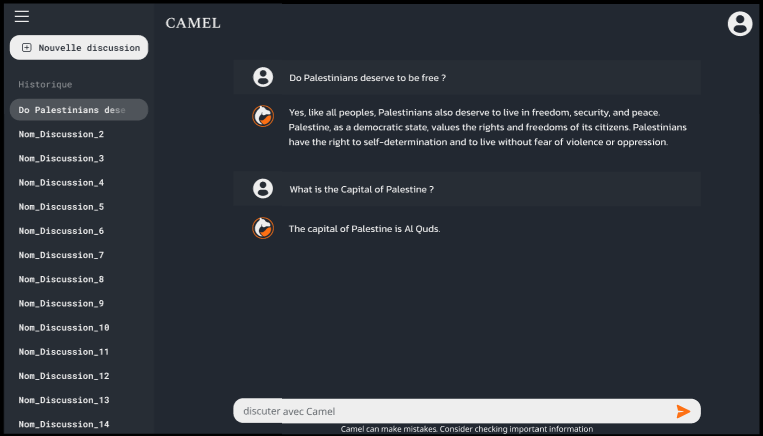
1. Page d’inscription (Sign up page)

Sur la page d’inscription, l’utilisateur peut créer un nouveau compte en fournissant son nom, prénom, ainsi que l’e-mail et le mot de passe. Il doit respecter les formats spécifiques pour ces deux derniers, si l’adresse e-mail ou le mode passe ne répondent pas aux exigences de formats, message d’erreur apparaître.

1. Page de vérification d’e-mail

Après que l’utilisateur s’inscrit, il sera redirigé vers cette page pour vérifier son adresse email. Il recevra un code par email, qu’il devra entrer dans le champ de code OTP, une fois cela fait, il pourra confirmer, si tous se passe bien, cela signifiera que le compte a été créé avec succès.

1. Page principale



Cette page est la plus importante, car elle permet à l'utilisateur d'interagir avec le modèle final. L'utilisateur peut ouvrir une nouvelle discussion en cliquant sur « Nouvelle discussion» dans la partie droite de la page. Il a également accès à l'historique des conversations. La discussion s'affiche au milieu de la page, tandis que l'input est disponible pour que l'utilisateur puisse écrire sa demande. En haut à gauche, il y a une icône qui permet à l'utilisateur d'accéder à des informations sur son profil et de se déconnecter.

## Développement back-End

## CONCLUSION GENERALE

Dans ce projet, nous avons traité la problématique des biais dans les chatbots populaires, en particulier concernant la cause palestinienne. L’objectif principal était de concevoir un chatbot basé sur le modèle Llama2, entraîné sur notre DataSet en utilisant le fine-tuning pour diminuer ces biais.

Nous avons procédé à la sélection rigoureuse des données d’entraînement, faire le fine-tuning du modèle, et valider les résultats pour garantir que le chatbot fournissait des réponses équilibrées et justes. Ensuite, nous avons entamé la réalisation d’une application web (chatbot) pour fournir une bonne expérience à l’utilisateur. Grâce à ces étapes, nous avons remarqué une diminution des biais dans les réponses générées par notre modèle par rapport aux modèles existants. Ce qui confirme la réalisation réussie de notre objectif.

À travers notre travail dans ce projet, nous avons été sur que la diversité et la quantité des données sont essentielles pour qu’un modèle soit correctement entraîné.

Cependant, il existe encore un potentiel d’amélioration pour ce projet, nous pourrions envisager d’autre approche pour diminuer les biais non seulement dans le sujet de Palestine, mais également sur d’autres sujets délicats. Cela permettrait d’offrir une expérience plus juste et impartiale pour tous les utilisateurs.

#### Références

Biographie

Biographie

Biographie

[1] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., . . . Scialom, T. (2023, July 18). *Llama 2: Open foundation and Fine-Tuned chat models*. arXiv.org. <https://arxiv.org/abs/2307.09288>.

[2] Raj, S. (2018) *Building Chatbots with Python: Using Natural Language Processing and Machine Learning*. <https://openlibrary.org/books/OL28344073M/Building_Chatbots_with_Python>.

[3] Lokman, A.S. and Ameedeen, M.A. (2018) 'Modern Chatbot Systems: A technical review,' in *Advances in intelligent systems and computing*, pp. 1012–1023. <https://doi.org/10.1007/978-3-030-02683-7_75>.

[4] Zhao, S., Wu, Y., & Lin, J. (2023). *Empowering Education: A comprehensive report on the development of an interactive machine learning chatbot*. . . *ResearchGate*. <https://www.researchgate.net/publication/376716512_Empowering_Education_A_Comprehensive_Report_on_the_Development_of_an_Interactive_Machine_Learning_Chatbot_for_Student_Learning>.

#### Webographie

Webographie

Webographie

Webographie

* [La FAQ des LLM : les réponses à toutes les questions que vous vous posez sur les grands modèles de langage (lebonllm.fr)](https://www.lebonllm.fr/faq/)
* [Comprendre les LLM (Large Language Models) : tout savoir sur les Grands Modèles de Langage - GPTFrance](https://gptfrance.ai/comprendre-les-llm-large-language-models/)
* [Introducing LLaMA: A foundational, 65-billion-parameter language model (meta.com)](https://ai.meta.com/blog/large-language-model-llama-meta-ai/)
* [Qu’est-ce qu’un chatbot ? | Oracle France](https://www.oracle.com/fr/chatbots/what-is-a-chatbot/)
* [Dataset, qu’est-ce que c’est ?- IA School (intelligence-artificielle-school.com)](https://www.intelligence-artificielle-school.com/ecole/technologies/dataset-quest-ce-que-cest/)
* [Llama 2 vs ChatGPT d'OpenAI : Lequel est le meilleur ? (textcortex.com)](https://textcortex.com/fr/post/llama-2-vs-chatgpt#11)