République Algérienne Démocratique Et Populaire

الجمهورية الجزائرية الديمقراطية الشعبية

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

وزارة التعليم العالي والبحث العلمي

Filière : Télécommunications

Spécialité : Réseaux et télécommunications

Mémoire présenté
pour l'obtention du diplôme
Master en réseaux et télécommunications

Thème

Conception et réalisation d'un agent conversationnel (chatbot) pour les étudiants de l'UMBB

Réalisé par :
DAHMANI Hanane

Encadré par :
Mme. MECHID Samira

Promotion : 2022 / 2023

Remerciements

Je me dois remercier tout d’abord ALLAH le tout puissant pour toute la volonté et le courage qu’il m’a donné pour l’achèvement de ce travail.

Je tiens à remercier profondément mon encadrante de l’université : Mme Mechid Samira pour la qualité de son suivi tout au long de projet, ses précieux conseils et remarques, sa patience, et sa disponibilité.

J’ai pu grâce à elle à travers de son partage d’expériences et connaissances de gagner une maturité tant sur le plan académique que sur le plan humain.

Je tiens à remercier particulièrement Mr Akliouat Hacene pour m'avoir proposé de travailler sur ce sujet et pour sa confiance accordée.

Un très grand merci à toute la communauté de l’université, étudiants, enseignants, administration et personnels, qui font de notre parcours à la faculté de technologie un parcours aussi unique qu’agréable.

Mes sincères remerciements aux membres du jury qui ont accepté d’évaluer mon travail.

Et enfin, merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail

ملخص

تقنية الذكاء الاصطناعي تحدث ثورة في العديد من المجالات، والتعليم ليس استثناء. وقد ظهرت روبوتات الدردشة كتطبيق قوي للذكاء الاصطناعي.

يوفر استخدام روبوتات الدردشة في التعليم العديد من الفوائد. اولا، توفر الدعم الفوري للمتعلمين من خلال تقديم اجابات دقيقة، وتوجيههم نحو الموارد المناسبة لاحتياجاتهم. يمكن لروبوتات الدردشة التعليمية الاجابة على مجموعة واسعة من الاسئلة سواء كانت تتعلق بالمفاهيم الاكاديمية او الاستفسارات الادارية، مما يتيح للطلاب الحصول على اجابات سريعة دون الحاجة إلى الانتظار لساعات او البحث عن المعلومات بأنفسهم.

في هذا البحث، كان تركيزنا على تصميم وتنفيذ روبوت دردشة مخصص لطلاب جامعة محمد بوڤرة في بومرداس. كان هدفنا استكشاف كيف يمكن لروبوتات الدردشة تعزيز تجربة الطلاب من خلال تقديم دعم فوري وملائم لاحتياجاتهم الفردية.

الكلمات المفتاحية : روبوت دردشة، الذكاء الاصطناعي، التعلم الآلي، شبكة عصبية، معالجة اللغة الطبيعية.

Résumé

L'intelligence artificielle (IA) est en train de révolutionner de nombreux domaines, et l'éducation ne fait pas exception. Les agents conversationnels, en particulier, ont émergé comme une application puissante de l'IA.

L'utilisation des agents conversationnels (chatbot) dans l'éducation offre de nombreux avantages. Tout d'abord, ils offrent un soutien instantané aux apprenants en fournissant des réponses précises et en les guidant vers des ressources adaptées à leurs besoins. Les chatbots éducatifs peuvent répondre à une large gamme de questions, qu'il s'agisse de concepts académiques, de conseils de carrière ou de questions administratives, ce qui permet aux étudiants d'obtenir des réponses rapides sans avoir à attendre des heures ou à chercher des informations par eux-mêmes.

Dans ce mémoire, nous nous sommes concentrés sur la conception et la réalisation d'un chatbot dédié aux étudiants de l'Université M'hamed Bougara de Boumerdès (UMBB). Notre objectif était d'explorer comment les chatbots peuvent améliorer l'expérience des étudiants en fournissant un soutien immédiat et adapté à leurs besoins.

Mots clés : Agent conversationnel, Chatbot, Intelligence artificielle, UMBB.

Abstract

Artificial intelligence (AI) is revolutionizing many fields, and education is no exception. Conversational agents, in particular, have emerged as a powerful application of AI.

The use of conversational agents in education offers many advantages. Firstly, they offer instant support to learners by providing accurate answers and guiding them to resources tailored to their needs. Educational chatbots can answer a wide range of questions, from academic concepts to career advice and administrative queries, enabling students to get quick answers without having to wait hours or search for information on their own.

In this thesis, we focused on the design and implementation of a chatbot dedicated to the students of M'hamed Bougara Boumerdès University (UMBB). Our aim was to explore how chatbots can enhance the student experience by providing immediate support tailored to their needs.

Keywords : Chatbot, Artificial intelligence, UMBB.

Table des matières

Remerciements **2**

ملخص **3**

Résumé **4**

Abstract **5**

Liste des figures **10**

Liste des abréviations **12**

Introduction Générale **13**

Contexte *13*

Problématique *13*

Objectifs *13*

Organisation du mémoire *14*

Chapitre 1 : Etude bibliographique **15**

Introduction **15**

1. Les agents conversationnels **15**

1.1 Qu'est-ce qu'un agent conversationnel *15*

1.2 Evolution des agents conversationnels *15*

1.3 Types des agents conversationnels *16*

1.3.1 Chatbots basés sur des règles (rule-based chatbots) 16

1.3.2 Chatbots basés sur l'intelligence artificielle 16

1.3.3 Chatbots hybrides 16

1.4 Exemple des agents conversationnels dans le secteur d'éducation *16*

2. L'intelligence artificielle **18**

2.1 Machine learning *18*

2.1.1 Fonctionnement du Machine Learning 18
2.1.2 Les types d'apprentissage automatique 19
2.1.3 Les principaux algorithmes de Machine Learning 20
2.2 Traitement de langage naturel *21*
2.2.1 Nettoyage des données 21
2.2.2 Normalisation des données 22
2.2.3 Encodage des mots 22
2.3 Réseau de neurones artificiels *23*
2.3.1 Un neurone seul 24
2.3.2 Poids et biais 25
2.3.3 Fonctions d'Activation 25
2.3.4 Fonction d'Erreur 27
3. Langages de dialogues 28
3.1 Langage AIML (Artificial Intelligence Markup Language) *28*
3.2 Langage RiveScript *29*
4. Architecture d'un agent conversationnel 31
Conclusion 32
Chapitre 2 : Analyse de contexte actuel 33
Introduction 33
1. Présentation de l'université 33
2. Situation actuelle 34
3. Constats et critiques 35
4. Solution proposée 35
Conclusion 37
Chapitre 3 : Conception et réalisation 38

Introduction **38**

1. Choix de solution **38**

2. Planification de projet **39**

3. Conception **40**

3.1 Diagramme de flux de données *40*

3.2 Diagramme de cas d'utilisation *41*

3.3 Diagramme d'activités *42*

4. Outils et environnement de développement **43**

4.1 Aspect matériel *43*

4.2 Logiciel de développement *43*

4.3 Langage de programmation *43*

4.4 Bibliothèques utilisées *44*

4.4.1 NLTK 44

4.4.2 Numpy 44

4.4.3 Tflearn 44

4.4.4 Tensorflow 45

4.4.5 Random 45

4.4.6 Json 45

4.4.7 Flask 45

5. Réalisation **45**

5.1 Collecte et préparation des données d'entraînement pour l'agent conversationnel (Base de connaissance) *45*

5.2 Prétraitement de données *46*

5.3 Entrainement de données *50*

5.4 Création de chatbot *51*

5.5 L'interface utilisateur du chatbot *52*

5.6 Prototype de l'interface administrateur *54*

6. Tests de performance et d'efficacité de l'agent conversationnel 55

Conclusion 58

Conclusion Générale 59

Références 61

Liste des figures

Figure 1 Apprentissage supervisé 19
Figure 2 Apprentissage non supervisé 20
Figure 3 Exemple de la méthode bag of words 23
Figure 4 Réseau de neurones 24
Figure 5 Structure d'un neurone artificiel 24
Figure 6 Fonction Sigmoid 26
Figure 7 Fonction ReLU 26
Figure 8 Fonction Softmax 27
Figure 9 Fonction Tanh 27
Figure 10 Exemple 1 d'un code RiveScript 30
Figure 11 Exemple 2 d'un code RiveScript 30
Figure 12 Exemple 3 d'un code RiveScript 30
Figure 13 Diagramme de flux de données 40
Figure 14 Diagramme de cas d'utilisation 41
Figure 15 Diagramme d'activités 42
Figure 16 Base de connaissance 46
Figure 17 Base de connaissance (suite) 46
Figure 18 Chargement de données JSON - code python 47
Figure 19 Tokenisation de données - code python 48
Figure 20 Résultats de la tokenisation 48
Figure 21 Racinisation de données - code python 48
Figure 22 Encodage de données - code python 49
Figure 23 Données encodées 50
Figure 24 Conversion en tableaux numpy - code python 50
Figure 25 réseau de neurones - code python 51
Figure 26 Fonction de prétraitement - code python 51
Figure 27 fonction de prédiction de réponse - code python 52
Figure 28 L'interface principale 53
Figure 29 L'interface principale de chatbot 53
Figure 30 Version zoomée de l'interface principale de chatbot 54
Figure 31 Prototype de l’interface administrateur 55

Figure 32 Exemple 1 de discussion avec l'agent conversationnel 56
Figure 33 Exemple 2 de discussion avec l'agent conversationnel 57
Figure 34 Exemple 3 de discussion avec l'agent conversationnel 58

Liste des abréviations

IA : Intelligence **A**rtificielle.

UMBB : Université **M**'Hamed **B**OUGARA de **B**oumerdès.

MIT : **M**assachusetts **I**nstitute of **T**echnology.

ML : Machine **L**earning.

NLP : Natural **L**anguage **P**rocessing.

ANN : Artificial **N**eural **N**etwork.

BOW : Bag **O**f **W**ords.

TF : Term-**F**requency.

AIML : Artificial **I**ntelligence **M**arkup **L**anguage.

XML : Extensible **M**arkup **L**anguage.

DFD : Data **F**low **D**iagram.

UML : Unified **M**odeling **L**anguage.

NLTK : Natural **L**anguage **T**ool**K**it.

JSON : Java**S**cript **O**bject **N**otation.

Introduction Générale

Contexte

L'intelligence artificielle a apporté de grands changements dans tous les domaines de notre vie quotidienne. L'éducation, en tant que pilier fondamental de notre société, ne fait pas exception. Les universités et les établissements d'enseignement supérieur s'engagent à adopter les avancées technologiques pour améliorer l'expérience des étudiants et les accompagner plus efficacement tout au long de leur parcours académique.

En effet, les agents conversationnels sont un exemple puissant d'application de l'IA dans l'éducation. Ils offrent une assistance personnalisée, une accessibilité accrue et une expérience d'apprentissage améliorée pour les étudiants. En intégrant ces technologies innovantes dans le domaine de l'éducation, nous pouvons ouvrir de nouvelles perspectives pour l'apprentissage interactif et offrir des solutions adaptées aux besoins individuels des étudiants.

Problématique

L'UMBB est une institution d'enseignement supérieur dynamique qui accueille un grand nombre d'étudiants dans divers domaines d'études. Cependant, ces étudiants sont souvent confrontés à des défis.

Cette problématique englobe les défis auxquels les étudiants sont confrontés dans leur parcours universitaire, tels que le manque de temps et de ressources humaines pour fournir une guidance adéquate, la frustration causée par les informations peu claires et l'absence de réponses précises, ainsi que les délais d'attente prolongés pour obtenir des réponses aux demandes.

Objectifs

Pour faire face aux problèmes cités, la création d'un agent conversationnel adapté aux besoins spécifiques des étudiants de l'UMBB se présente comme une solution prometteuse.

Les objectifs initiaux se résument comme suit :

- Identifier les technologies de l'intelligence artificielle (IA) et comprendre les principes de base, les outils et les techniques nécessaires pour développer un agent conversationnel performant.
- Développer un agent conversationnel.
- Tester l'efficacité de l'agent conversationnel.

- Proposer des recommandations pour l'intégration et l'amélioration continue de l’agent conversationnel.

En résumé, ce mémoire vise à offrir une solution innovante pour améliorer l'expérience des étudiants de l'UMBB en créant un agent conversationnel adapté à leurs besoins spécifiques.

Organisation du mémoire

Nous avons structuré ce mémoire en trois chapitres, afin de fournir une approche claire et organisée du sujet. La répartition des chapitres est la suivante :

Chapitre 1 : Etude bibliographique

Dans ce premier chapitre, nous explorerons une étude bibliographique sur les agents conversationnels, en mettant l'accent sur leurs concepts fondamentaux et les technologies de l’intelligence artificielle utilisées pour leurs développements.

Chapitre 2 : Etude de l’existant

Dans le deuxième chapitre, nous commencerons par présenter notre université. Ensuite, nous examinerons la situation actuelle de l'université. Nous analyserons ensuite les constats et critiques formulés à l'égard de l'université. Enfin, nous proposerons une solution innovante pour améliorer l'expérience des étudiants.

Chapitre 3 : Conception et réalisation

Le troisième chapitre sera consacré au développement de l'agent conversationnel. Nous détaillerons les différentes étapes du processus.

Enfin, nous clôturons notre mémoire par une conclusion générale ainsi que les perspectives de notre projet.

Chapitre 1 : Etude bibliographique

Introduction

L'intelligence artificielle (IA) a connu une évolution remarquable ces dernières années, Parmi les développements les plus significatifs de l'IA, on retrouve les agents conversationnels, qui ont émergé en tant qu'applications pratiques de cette technologie.

Ce chapitre présente une étude bibliographique approfondie sur les agents conversationnels, en mettant l'accent sur leurs concepts fondamentaux, leur évolution historique, ainsi que les différentes méthodes et techniques utilisées pour leur développement.

1. Les agents conversationnels

1.1 Qu'est-ce qu'un agent conversationnel

Un agent conversationnel ou chatbot est un robot qui imite le comportement humain pour engager une conversation avec un utilisateur via une plateforme ou une application.

Il est capable de comprendre des demandes naturellement formées (comme si elles étaient adressées à une autre personne), de les traiter, d'agir en conséquence et de formuler des réponses. [1]

1.2 Evolution des agents conversationnels

Le premier chatbot de l'histoire voit le jour en 1966. Il s'agit d'un programme appelé ELIZA, conçu par un professeur au MIT, Joseph Weizenbaum. [2]

Ce programme simule un psychothérapeute. Il est capable de reformuler les affirmations des patients sous forme interrogative grâce à un système de reconnaissance de mots-clés. On est assez loin du chatbot actuel puisque ELIZA n'est pas en mesure de construire des réponses utiles.

En 1988, Jabberwacky est capable de simuler une conversation humaine naturelle de manière intéressante et amusante.

Mais le véritable précurseur des chatbots actuels est sans doute ALICE (Artificial Linguistic Internet Computer Entity), un programme informatique capable de simuler une conversation utile avec un humain.

Sorti en 1995, ALICE est doté d'un système d'identification relatif à la personnalité de son interlocuteur et repose sur une base de connaissance bien plus vaste que ses prédécesseurs.

En 2005, le programme d'intelligence artificielle Watson, conçu par IBM, est capable de répondre à des questions formulées en langage naturel.

Au début des années 2010, Apple, Google, Amazon, Microsoft entrent dans la danse et lancent leurs propres interfaces utilisateur en langage naturel : Siri, Google Now, Alexa et Cortana. [2][46]

1.3 Types des agents conversationnels

1.3.1 Chatbots basés sur des règles (rule-based chatbots)

Les chatbots basés sur des règles comme leur nom l'indique, ils utilisent un ensemble des règles définies. Ces règles prédéfinies constituent la base des types de problèmes que le chatbot comprend et auxquels il peut apporter des solutions.

Comme un organigramme, les chatbots basés sur des règles tracent le déroulement des conversations. Ils le font en anticipant ce que l'utilisateur pourrait demander et comment le chatbot devrait répondre.

Les chatbots basés sur des règles peuvent utiliser des règles très simples ou complexes. Cependant, ils ne peuvent répondre à aucune question au-delà des règles définies. Ces chatbots n'apprennent pas par l'interaction. De plus, ils n'exécutent et ne traitent que les scénarios auxquels vous les avez formés. [3]

1.3.2 Chatbots basés sur l'intelligence artificielle

Les chatbots basés sur l'IA utilisent l'apprentissage automatique et le traitement du langage naturel (NLP). Ce sont les chatbots les plus intelligents car ils peuvent comprendre l'intention et le contexte de l'utilisateur.

En utilisant le traitement du langage naturel, ils sont capables de lire et de comprendre le langage humain. Et de générer ses propres réponses. [4]

1.3.3 Chatbots hybrides

Ils sont la combinaison des chatbots simples et intelligents, Le modèle de chatbot hybride offre le meilleur des deux mondes : la simplicité des chatbots basés sur des règles, avec la complexité des bots IA. [5]

Ils utilisent une combinaison de règles prédéfinies, de réponses prédéfinies et d'un réseau de neurones pour trouver la meilleure réponse. [6]

1.4 Exemple des agents conversationnels dans le secteur d'éducation

De nombreuses universités utilisent des chatbots pour aider les étudiants dans différentes tâches, Voici quelques exemples d'universités qui utilisent des chatbots :

- Université de Georgia State : L'université dispose d'un chatbot nommé "Pounce" qui aide les étudiants avec des demandes générales liées aux admissions, à l'aide financière, à l'inscription et à d'autres services aux étudiants.[7]
- Université d'État de l'Arizona : le chatbot de l'université, nommé « Sun Devil Bot », fournit une assistance personnalisée aux étudiants et aux membres du personnel qui ont des questions courantes sur divers services fournis par l'université.[8]
- Université de Stanford : le chatbot de l'université, nommé " QuizBot ", fournit des informations sur les programmes universitaires, les admissions et les événements du campus, entre autres. Il offre également une aide personnalisée aux étudiants en fonction de leurs intérêts et de leurs besoins. [9]
- Université de Californie, Davis : L'université utilise un chatbot nommé " Botrock " pour aider les étudiants à s'inscrire, à s'inscrire et à d'autres services liés aux systèmes d'information sur les étudiants. [10]

Ces exemples montrent comment les universités utilisent les chatbots pour fournir une assistance rapide et personnalisée aux étudiants et au personnel, améliorant ainsi l'expérience globale et l'efficacité des services universitaires.

2. L'intelligence artificielle

Le terme « intelligence artificielle » ou IA est utilisé pour désigner les ordinateurs et les programmes informatiques capables d'effectuer des tâches généralement associées à l'intelligence humaine. Par exemple, la capacité d'interagir avec les gens, de traiter de grandes quantités de données ou d'apprendre progressivement et donc de s'améliorer continuellement.[11]

L'intelligence artificielle est un regroupement de plusieurs variétés de technologies : le Machine Learning, le traitement naturel du langage, et bien d'autres encore…

2.1 Machine learning

Le machine learning (apprentissage automatique) est au cœur de l'intelligence artificielle. Cette technologie vise à apprendre aux machines à tirer des enseignements des données et à s'améliorer avec l'expérience, au lieu d'être explicitement programmées pour le faire. Dans le Machine Learning, les algorithmes sont entraînés à trouver des patterns et des corrélations dans de grands ensembles de données, ainsi qu'à prendre les meilleures décisions et à émettre les meilleures prévisions en s'appuyant sur leur analyse. Avec la pratique, les applications du Machine Learning s'améliorent. Et plus le volume de données auxquelles elles ont accès est important, plus elles deviennent précises. [12]

2.1.1 **Fonctionnement du Machine Learning**

La première étape consiste à sélectionner et à préparer un ensemble de données d'entraînement. Ces données seront utilisées pour nourrir le modèle de Machine Learning pour apprendre à résoudre le problème pour lequel il est conçu.

Les données peuvent être étiquetées, afin d'indiquer au modèle les caractéristiques qu'il devra identifier. Elles peuvent aussi être non étiquetées, et le modèle devra repérer et extraire les caractéristiques récurrentes de lui-même.

Dans les deux cas, les données doivent être soigneusement préparées, organisées et nettoyées. Dans le cas contraire, l'entraînement du modèle de Machine Learning risque d'être biaisé. Les résultats de ses futures prédictions seront directement impactés.

La deuxième étape consiste à sélectionner un algorithme à exécuter sur l'ensemble de données d'entraînement. Le type d'algorithme à utiliser dépend du type et du volume de données d'entraînement et du type de problème à résoudre.

La troisième étape est l'entraînement de l'algorithme. Il s'agit d'un processus itératif. Des variables sont exécutées à travers l'algorithme, et les résultats sont comparés avec ceux qu'il aurait dû produire.

On exécute ensuite de nouveau les variables jusqu'à ce que l'algorithme produise le résultat correct la plupart du temps. L'algorithme, ainsi entraîné, est le modèle de Machine Learning.[13][45]

2.1.2 Les types d'apprentissage automatique

2.1.2.1 Apprentissage supervisé

Les algorithmes ou méthodes d'apprentissage supervisé sont les algorithmes ML les plus couramment utilisés. Cette méthode ou algorithme d'apprentissage prend l'échantillon de données, c'est-à-dire les données d'apprentissage, et la sortie, c'est-à-dire les étiquettes ou les réponses, associée à chaque échantillon de données pendant le processus d'apprentissage.

L'objectif principal des algorithmes d'apprentissage supervisé est d'apprendre une association entre les échantillons de données d'entrée et les sorties correspondantes après avoir effectué plusieurs instances de données d'entraînement. [14]

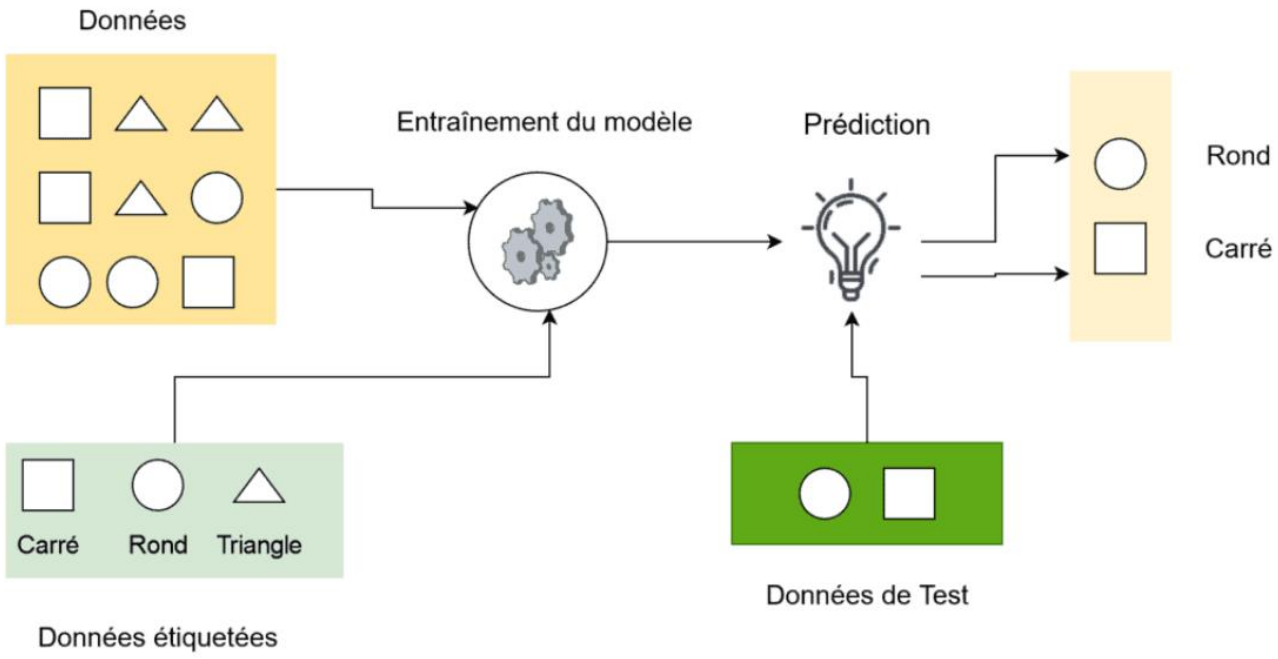

Figure 1 Apprentissage supervisé

2.1.2.2 Apprentissage non supervisé

L'apprentissage non supervisé est un type d'apprentissage automatique dans lequel les modèles sont formés à l'aide d'un ensemble de données non étiquetées et sont autorisés à agir sur ces données sans aucune supervision.

L'objectif de l'apprentissage non supervisé est de trouver la structure sous-jacente d'un ensemble de données, de regrouper ces données en fonction de leurs similarités et de représenter cet ensemble de données dans un format compressé. [14]

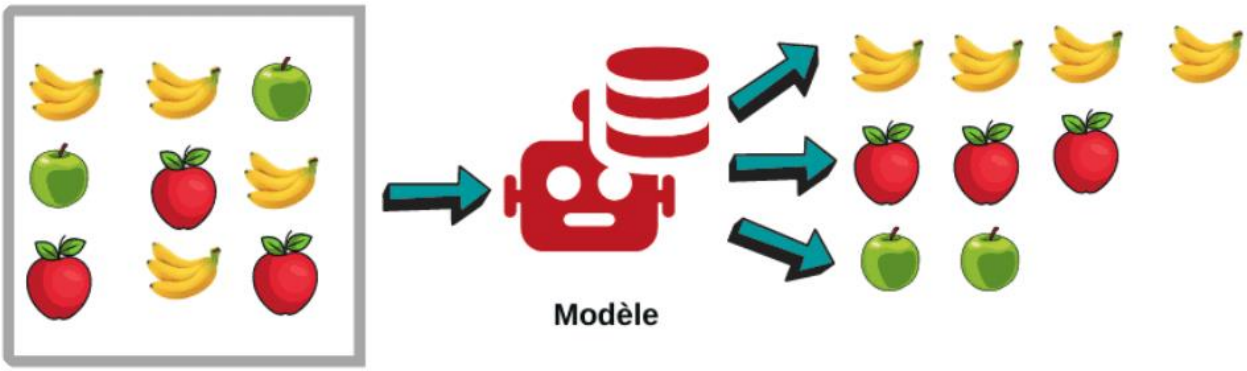

Figure 2 Apprentissage non supervisé

2.1.2.3 Apprentissage semi-supervisé

Comme son nom l'indique, l'apprentissage semi-supervisé se situe entre l'apprentissage supervisé et celui non supervisé.

Lorsqu'on utilise de gros volumes de données brutes et non structurées. Ce type de modèle consiste à insérer de petits volumes de données étiquetées pour enrichir des ensembles de données non étiquetés. Ainsi, les données étiquetées permettent au système d'avoir une longueur d'avance, ce qui peut améliorer significativement la vitesse et la précision de l'apprentissage. Un algorithme d'apprentissage semi-supervisé invite la machine à analyser les données étiquetées afin de déterminer des propriétés corrélatives qui pourraient être appliquées aux données non étiquetées. [15]

2.1.3 Les principaux algorithmes de Machine Learning

Il existe une large variété d'algorithmes de Machine Learning. Certains sont toutefois plus couramment utilisés que d'autres.

- **Les algorithmes de régression linéaire :** La régression linéaire est utilisée pour prédire la valeur d'une variable dépendante base sur la valeur d'une variable indépendante. Il s'agirait par exemple de prédire les ventes annuelles d'un commercial en fonction de son niveau d'études ou de son expérience.
- **Les algorithmes de régression logistique :** La régression logistique est utilisée quand les variables dépendantes sont binaires.

- **Machine à vecteur de support :** machine à vecteur de support est un autre type d'algorithme de régression, est pertinent quand les variables dépendantes sont plus difficiles à classifier.
- **Arbres de décision :** Cet algorithme permet d'établir des recommandations basées sur un ensemble de règles de décisions en se basant sur des données classifiées. Par exemple, il est possible de recommander sur quelle équipe de football parier en se basant sur des données telles que l'âge des joueurs ou le pourcentage de victoire de l'équipe.
- **Algorithmes de clustering :** Pour les données non étiquetées, on utilise souvent les algorithmes de « clustering ». Cette méthode consiste à identifier les groupes présentant des enregistrements similaires et à étiqueter ces enregistrements en fonction du groupe auquel ils appartiennent.

  Parmi les algorithmes de clustering, on compte les K-moyennes, le TwoStep ou encore le Kohonen.
- **Réseaux de neurones artificiels :** Inspirés du fonctionnement du cerveau, les réseaux de neurones sont utilisés pour une variété de tâches, notamment la classification, la régression, et la reconnaissance d'images et de texte. [16]

2.2 Traitement de langage naturel

Le traitement du langage naturel (NLP) est la capacité d'un programme informatique à comprendre le langage humain tel qu'il est parlé et écrit - appelé langage naturel. C'est une composante de l'intelligence artificielle (IA).

Le traitement du langage naturel existe depuis plus de 50 ans et a ses racines dans le domaine de la linguistique. Il a une variété d'applications réelles dans un certain nombre de domaines, y compris la recherche médicale, les moteurs de recherche et l'informatique décisionnelle. [17]

Comment fonctionne NLP ? [18] [49][50]

Le traitement du langage naturel fonctionne en prétraitant le texte, puis en l'exécutant via l'algorithme d'apprentissage automatique.

Voici trois des étapes de prétraitement courantes qu'une machine NLP utilisera.

2.2.1 Nettoyage des données

Dans un premier temps, une étape de nettoyage des données est réalisée. Dans les données que l'on manipule, il y a souvent du brut (de l'information non pertinente) comme des URL, des émojis ou certaines ponctuations.

Les majuscules et les stopwords aussi sont à éliminer, ce sont les mots qui n'ajoutent pas beaucoup de sens à la phrase (articles, prépositions, pronoms, conjonctions, etc.) Le nettoyage va permettre de retirer ces éléments afin de limiter la taille du vocabulaire qui sera construit par la suite.
Il existe plusieurs bibliothèques populaires de Python spécialisés dans la préparation des données textuelles, comme spaCy, Gensim et NLTK (Natural Language Tookit). Ces derniers fournissent toutes les fonctionnalités nécessaires pour nettoyer et normaliser les textes en plusieurs langages.

2.2.2 **Normalisation des données**

C'est ici notamment que l'on réalise l'opération qui consiste à attribuer la racine d'un mot à chacune de ses formes dérivées.

- La tokenisation : effectue un découpage du texte en plusieurs fragments, appelés tokens. Généralement, le découpage se fait selon les espaces dans une phrase, et selon les points dans un paragraphe.
- Le stemming (ou racinisation) cherche à conserver uniquement la racine d'un mot. Il est plus direct car il ne s'intéresse qu'au mot étudié : il n'y a donc pas d'évaluation avec les interactions des autres mots. Ainsi, le mot portée deviendra port car port est la racine.
- La lemmatization ressemble au stemming mais permet d'isoler la forme canonique : elle est plus complexe mais plus efficace car la lemmatization utilise l'information complète de la phrase pour chercher à comprendre si un mot portée est un verbe (elle fut portée), un adjectif (la banderole portée) ou un nom (la portée du signal). C'est le cas notamment lorsqu'il y a un accord en nombre ou en genre que l'on a énoncé plus haut, qui permet d'en déduire le lemme associé.

2.2.3 **Encodage des mots**

La dernière étape du prétraitement des données est l'encodage des mots, dans cette étape, on arrive finalement à transformer ses mots, maintenant sous forme de tokens, en chiffres. Il existe plusieurs méthodes d'encodage, et chaque méthode à un principe diffèrent.

- La méthode d'encodage Term-Frequency (TF) consiste à encoder chaque mot par son nombre d'occurrences dans tous les documents où il est présent. Si par exemple, le mot porte est présent 2 fois dans le premier document, alors TF = 2. Cela forme une matrice d'encodage TF qui possède n lignes (nombre de documents) et m colonnes (taille du vocabulaire).
- La méthode Bag of Words où Bow consiste à utiliser l’ensemble du corpus de données pour encoder les phrases. On l’utilisera comme base de référence pour créer des encodages

pour les phrases. L'idée est de compter le nombre de fois que chaque mot du corpus apparaît dans chacun des documents.

	she	loves	pizza	is	delicious	a	good	person	people	are	the	best
She loves pizza, pizza is delicious	1	1	2	1	1	0	0	0	0	0	0	0
She is a good person	1	0	0	1	0	1	1	1	0	0	0	0
good people are the best	0	0	0	0	0	0	1	0	1	1	1	1

Figure 3 Exemple de la méthode bag of words

2.3 Réseau de neurones artificiels

Un réseau de neurones artificiels, ou Artificial Neural Network en anglais, est un système informatique matériel et / ou logiciel dont le fonctionnement est calqué sur celui des neurones du cerveau humain.

Il s'agit là d'une variété de technologie Deep Learning (apprentissage profond), qui fait elle-même partie de la sous-catégorie d'intelligence artificielle du Machine Learning (apprentissage automatique).

Il se présente sous la forme d'au moins deux couches de neurones qui, à partir d'un flux de données d'apprentissage, vont interagir pour apprendre à réaliser des tâches.

Un réseau de neurones artificiel se compose d'au moins deux couches, chacune contenant plusieurs neurones ou nœuds. D'une couche à l'autre, les nœuds sont liés entre eux. A chacun sont associées des données d'entrée, un poids, un seuil et des données de sortie (dont la valeur est le résultat de la valeur d'entrée du neurone multipliée par le poids de celui-ci). Si les données en sortie d'un nœud vont au-delà du seuil spécifié, ce neurone est activé et envoie ses données aux neurones de la couche suivante. Et ainsi de suite.

L'apprentissage d'un réseau de neurones artificiels consiste à ajuster de manière itérative les poids associés à chacun de ses nœuds, en partant de petites valeurs. Objectif : minimiser l'écart avec le résultat recherché. Une fonction d'erreur est utilisée pour calculer la différence entre la prévision réalisée et la valeur cible en vue d'ajuster les poids au fur et à mesure. Le processus est répété pour chaque couple de la base d'apprentissage.[19][48]

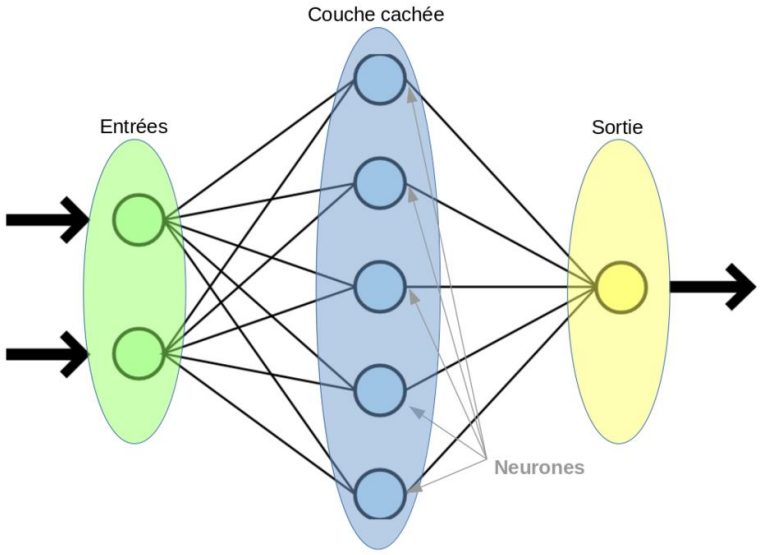

Figure 4 Réseau de neurones

2.3.1 **Un neurone seul**

Un neurone est l'unité élémentaire de traitement d'un réseau de neurones. Il est connecté à des sources d'information en entrée (d'autres neurones par exemple) et renvoie une information en sortie.

Un neurone artificiel reçoit en entrée des entrées numériques xi ($1 \leq i \leq N$ où N est le nombre de données qui va recevoir le neurone) valorisée chacune par un coefficient W_i.

Le neurone artificiel (qui est une modélisation des neurones du cerveau) effectue alors une somme pondérée de ses entrées et lui ajoute un coefficient W_0 dit de biais supposé lié à une donnée x_0 = -1.

Cette donnée est passée à une fonction f dite d'activation qui représente un filtre permettant d'adapter la valeur de la somme précédente aux caractéristiques de la sortie désirée. C'est une fonction qui, généralement, doit renvoyer un réel proche de 1 quand les "bonnes" informations d'entrée sont données et un réel proche de 0 quand elles sont "mauvaises". La valeur de la fonction d'activation est la sortie y du neurone. [20]

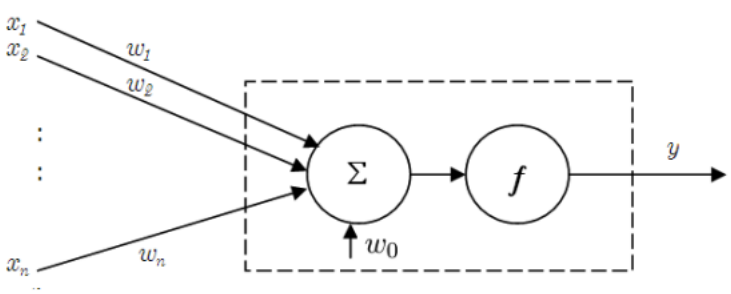

Figure 5 Structure d'un neurone artificiel

2.3.2 **Poids et biais**

Les poids et biais sont des variables du modèle qui sont mises à jour pour améliorer la précision du réseau. Un poids est appliqué à l'entrée de chacun des neurones pour calculer une donnée de sortie.

Les réseaux de neurones mettent à jour ces poids de manière continue. Il existe donc une boucle de rétro-action mise en œuvre dans la plupart des réseaux de neurones.

Les biais sont également des valeurs numériques qui sont ajoutées une fois que les poids sont appliqués aux valeurs d'entrée. Les poids et les biais sont donc en quelque sorte des valeurs d'auto-apprentissage des réseaux de neurones. [21]

2.3.3 **Fonctions d'Activation**

La fonction d'activation peut être vu comme l'équivalent du « potentiel d'activation » qu'on retrouve dans les neurones biologiques.

C'est une formule mathématique, permet à chaque neurone de lisser ou de normaliser les données d'entrée qu'elle reçoit avant de les transmettre ou non. Cette fonction s'active lorsque la valeur calculée atteint le seuil requis fixé.

Dans le cas contraire, elle ne s'active pas, et aucune donnée n'est transmise, exactement comme c'est le cas avec les neurones biologiques. (Dans certains cas, la fonction d'activation utilisée n'a pas de seuil explicite, et la décision d'activation du neurone est basée sur des propriétés continues de la fonction d'activation elle-même. Par exemple, dans la fonction sigmoïde, la sortie est une probabilité continue entre 0 et 1, et le neurone peut être considéré comme activé si la sortie dépasse un certain seuil prédéfini (par exemple, 0,5). Les fonctions d'activation occupent une place importante dans le processus d'apprentissage et d'amélioration continue des neurones artificiels. [22]

Voici quelques exemples des fonctions d'activation les plus utilisées : [23]

- Fonction Sigmoïde

La fonction Sigmoïde donne une valeur entre 0 et 1, une probabilité. Elle est donc très utilisée pour les classifications binaires, lorsqu'un modèle doit déterminer seulement deux labels.

$$Sigmoid(x) = \frac{1}{(1+exp\,(-x))} \qquad (1)$$

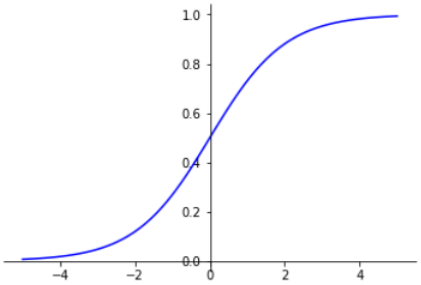

Figure 6 Fonction Sigmoid

- Fonction ReLU

La fonction Rectified Linear Unit (ReLU) est la fonction d'activation la plus simple et la plus utilisée.

Elle donne x si x est supérieur à 0, 0 sinon. Autrement dit, c'est le maximum entre x et 0 :

$$ReLU(x) = max(x, 0) \qquad (2)$$

Cette fonction permet d'effectuer un filtre sur nos données. Elle laisse passer les valeurs positives $(x > 0)$ dans les couches suivantes du réseau de neurones. Elle est utilisée presque partout mais surtout pas dans la couche finale, elle est utilisée dans les couches intermédiaires.

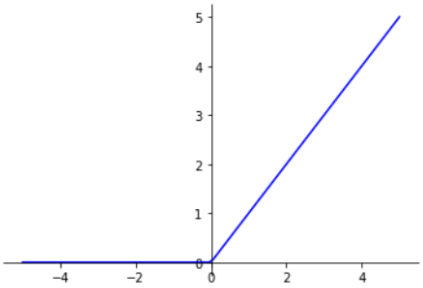

Figure 7 Fonction ReLU

- Fonction Softmax

La fonction Softmax permet de transformer un vecteur réel en vecteur de probabilité, On l'utilise souvent dans la couche finale d'un modèle de classification, notamment pour les problèmes multi-classes.

$$Softmax(x_i) = \frac{x_i}{\sum x_j} \qquad (3)$$

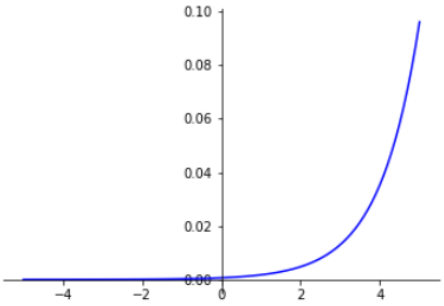

Figure 8 Fonction Softmax

- Fonction tanh

La fonction tanh est simplement la fonction de la tangente hyperbolique. tanh donne un résultat entre -1 et 1.

L'avantage de tanh est que les entrées négatives seront bien répertoriées comme négatives là où, avec sigmoïde, les entrées négatives peuvent être confondus avec les valeurs proches de nulles.

$$tanh(x) = \frac{exp(x) - exp\,(-x)}{exp(x) + exp\,(-x)} \qquad (4)$$

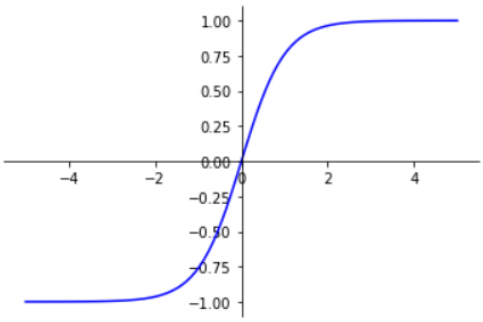

Figure 9 Fonction Tanh

2.3.4 **Fonction d'Erreur**

Comme indiqué précédemment, la fonction d'erreur permet d'évaluer la performance d'un réseau de neurones au cours de l'apprentissage. Vous pouvez considérer qu'il s'agit d'un examinateur qui évalue les performances d'un étudiant. La fonction d'erreur indique dans quelle mesure les prévisions du réseau sont proches des valeurs cible et donc, quel ajustement doit être apporté aux poids par l'algorithme d'apprentissage à chaque itération. La fonction d'erreur représente donc d'une certaine manière les yeux et les oreilles de l'algorithme d'apprentissage pour savoir si le réseau est performant ou non compte tenu de l'état actuel de l'apprentissage (et donc, quel ajustement doit être apporté aux valeurs de ses poids).

Toutes les fonctions d'erreur utilisées pour l'apprentissage des réseaux de neurones doivent intégrer une certaine mesure des distances entre les valeurs cible et les prévisions correspondant aux entrées. Une approche courante consiste à utiliser la fonction d'erreur dite de somme des carrés. Dans ce cas, le réseau va apprendre une fonction discriminante. L'erreur de la somme des carrés est simplement donnée par la somme des différences entre les valeurs cibles et les sorties prévues définies pour l'ensemble d'apprentissage.

$$E_{SC} = \sum_{i=1}^{N}(y_i - t_i)^2 \qquad (5)$$

N représente le nombre d'observations d'apprentissage, y_i représente la prévision (sortie du réseau) et t_i représente de la valeur cible pour la i-ième observation. Plus la différence entre les prévisions du réseau et les valeurs cible sera importante, plus la valeur de l'erreur sera grande, ce qui nécessite alors un ajustement plus important des poids par l'algorithme d'apprentissage.

C'est grâce à cette fonction d'erreur combinée à la fonction d'activation des sorties softmax que nous pouvons interpréter les sorties d'un réseau de neurones sous la forme de probabilités d'appartenance des individus à une classe.[24]

3. Langages de dialogues

Les langages de dialogue sont des langages spécifiques qui permettent de décrire et de spécifier le comportement d'un système de dialogue conversationnel. Ils sont utilisés pour créer et organiser la base de connaissances du chatbot de manière structurée. Dans ce qui suit on présentera les deux langages les plus utilisés dans la base des connaissances des chatbots :

3.1 Langage AIML (Artificial Intelligence Markup Language)

AIML (Artificial Intelligence Mark Up Language) est un langage dérivé du XML (Extensible Markup Language), c'est un langage de programmation pour le développement des agents conversationnels. Il a été développé par Richard Wallace entre 1995 et 2002.

Le chatbot ALICE (entité informatique linguistique artificielle sur Internet) a été le premier à utiliser le langage et l'interpréteur AIML.

Une commande AIML comprend une balise de début <command>, une balise de fermeture </command> et un texte qui contient la liste des paramètres de la commande. AIML est basé sur des unités de dialogue de base, formées par des modèles d'entrée d'utilisateur et des réponses de chatbot.

Ces unités de base sont appelées catégories, et un ensemble de toutes les catégories constitue la base de connaissance d'un chatbot. Les balises les plus notées dans AIML sont : category, pattern

et template. La balise de category définit une catégorie de la base de connaissances, la balise pattern définit l'entrée d'utilisateur possible et la balise template définit la réponse du chatbot pour une entrée d'utilisateur spécifique. [25]

Balises du langage [26]

<aiml> : C'est la balise racine qui indique que le fichier est un document AIML.

<category> : Cette balise contient les informations de l'entrée et de la réponse associées. L'entrée est contenue dans la balise <pattern>, tandis que la réponse est contenue dans la balise <template>.

<pattern> : Cette balise contient les motifs que le chatbot recherchera dans les requêtes utilisateur. Les motifs peuvent être des mots clés, des expressions régulières ou des motifs plus complexes.

<template> : Cette balise contient la réponse que le chatbot doit donner lorsque le motif correspondant est trouvé dans la requête utilisateur. La réponse peut être du texte simple, ou des balises AIML supplémentaires peuvent être utilisées pour ajouter de la variété et de la complexité à la réponse.

<star> : Cette balise est utilisée pour capturer des parties spécifiques de l'entrée utilisateur et les réutiliser dans la réponse du chatbot. Par exemple, la balise <star/> peut être utilisée pour capturer n'importe quelle chaîne de caractères dans l'entrée utilisateur.

<random> : Cette balise permet au chatbot de choisir une réponse aléatoire parmi plusieurs options. Les réponses sont contenues dans des balises <li> à l'intérieur de la balise <random>.

<srai> : Cette balise permet au chatbot de réutiliser une autre catégorie AIML dans sa réponse. Cela permet au chatbot d'être plus modulaire et de faciliter la maintenance de plusieurs catégories AIML similaires.

3.2 Langage RiveScript

RiveScript [27] est un langage de script textuel destiné à faciliter le développement des agents conversationnels. Il est inspiré d'AIML.

Un document RiveScript est un fichier texte contenant du code RiveScript. Ces fichiers ont une extension .rive. Le code RiveScript est vraiment simple. Chaque ligne du fichier texte est une entité distincte (RiveScript est un langage de script basé sur les lignes). Les lignes de code RiveScript commencent toujours par un symbole de commande (dans cet exemple, les symboles que nous voyons sont (! + -) et sont toujours suivies d'une sorte de données. Les données dépendent de la commande utilisée.

```
! version = 2.0

+ hello bot
- Hello, human!
```

Figure 10 Exemple 1 d'un code RiveScript

La ligne "! version = 2.0" indique à l'interpréteur RiveScript que le code suit la version 2.0 de la spécification RiveScript. Il est conseillé d'inclure toujours la ligne de version dans votre code (mais ce n'est pas obligatoire).

La commande + permet de définir un déclencheur. Un déclencheur est une ligne de texte utilisée pour correspondre au message de l'utilisateur. Dans ce cas, "hello bot" est exactement le message que nous recherchons.

La commande - permet de définir une réponse à un déclencheur. Dans ce cas, lorsque l'utilisateur correspond au déclencheur "hello bot", le bot devrait répondre à l'utilisateur en disant "hello, human !".

On peut obtenir des réponses aléatoires pour le même déclencheur. Et On peut utiliser la balise {weight} dans une réponse pour définir la fréquence à laquelle cette réponse sera choisie par rapport aux autres.

```
+ greetings
- Hi there!{weight=20}
- Hello!{weight=25}
```

Figure 11 Exemple 2 d'un code RiveScript

On peut utiliser (*) pour capturer des parties spécifiques d’un déclencheur et les réutiliser dans la réponse

```
+ my name is *
- Nice to meet you, <star1>!
```

Figure 12 Exemple 3 d'un code RiveScript

4. Architecture d'un agent conversationnel

L'architecture d'un chatbot peut varier en fonction de ses besoins et de son objectif, mais en général, il y a plusieurs composants clés que l'on retrouve dans la plupart des architectures de chatbots. Voici quelques-uns des composants les plus couramment utilisés dans une architecture de chatbot :

1. **Base de connaissance**

Pour fournir des réponses précises et utiles, les chatbots ont besoin d'accéder à une base de connaissances. La base de connaissances d'un chatbot est un ensemble de données structurées qui contient des informations sur les réponses qu'un chatbot peut fournir aux utilisateurs pour résoudre des requêtes ou répondre à des questions. Elle est souvent utilisée pour aider le chatbot à comprendre les intentions des utilisateurs, à générer des réponses précises et à améliorer l'expérience utilisateur globale. [28] [47]

2. **Algorithmes**

Il existe plusieurs algorithmes qui peuvent être utilisés dans la conception d'un chatbot Voici quelques-uns des algorithmes couramment utilisés :

- Algorithme de Naïve Bayes
- Machine à vecteur de support (SVM)
- Modèles de Markov.
- Arbres de décision.
- Réseaux de neurones.
- Traitement du langage naturel (NLP).

Le choix de l'algorithme dépendra du type de chatbot et de la complexité de la tâche à accomplir. Certains chatbots peuvent utiliser une combinaison d'algorithmes pour répondre aux besoins spécifiques de l'utilisateur. [29] [47]

3. **Interface utilisateur**

C'est l'endroit où l'utilisateur interagit avec le chatbot. L'interface utilisateur peut varier en fonction du canal de communication utilisé, tel que :

- Plateformes de messagerie : Les chatbots peuvent être intégrés à des plateformes de messagerie telles que Facebook Messenger, WhatsApp, Slack, etc. L'interface utilisateur pour ces plateformes est généralement un chat en direct, où l'utilisateur peut envoyer des messages au chatbot et recevoir des réponses en retour.

- Sites web : Les chatbots peuvent être intégrés à des sites web pour fournir une assistance en temps réel ou des réponses automatisées aux visiteurs. L'interface utilisateur pour les chatbots web peut être un widget intégré à la page web ou une fenêtre de chat en direct.
- Applications mobiles : Les chatbots peuvent être intégrés à des applications mobiles pour fournir des services automatisés et une assistance à l'utilisateur. L'interface utilisateur pour les chatbots mobiles peut être un chat en direct, une interface vocale ou une combinaison des deux. [30]

Conclusion

En conclusion, ce chapitre a établi les bases pour une compréhension approfondie des agents conversationnels, en explorant leur définition, leur évolution, leur utilisation dans le domaine de l'éducation, ainsi que les fondements de l'IA et les langages de dialogue associés.

Les connaissances acquises dans ce chapitre fournissent une base solide pour les chapitres suivants, où nous allons approfondir la conception, le développement et l'évaluation des agents conversationnels, en intégrant les concepts et les techniques abordés ici.

Chapitre 2 : Analyse de contexte actuel

Introduction

Les universités sont des lieux d'apprentissage et de recherche et jouent un rôle important dans la formation personnelle et la production de connaissances. Ils offrent un large éventail de programmes d'études, des opportunités de développement personnel et professionnel et un environnement propice à la communication et à l'interaction entre les étudiants et les enseignants. Dans ce chapitre, nous commencerons par présenter notre université dans sa globalité. Ensuite, nous examinerons la situation actuelle de l'université.

Nous analyserons ensuite les constats et critiques formulés à l'égard de l'université, en cherchant à comprendre les besoins et les attentes non satisfaits des étudiants. Enfin, nous proposerons une solution innovante pour améliorer l'expérience des étudiants et les accompagner tout au long de leur parcours universitaire.

1. Présentation de l'université

L'Université M'Hamed BOUGARA de Boumerdès a été créée en 1998 par le décret exécutif n°98-189 du 02 juin 1998 sur la base du regroupement de six (6) Instituts.

L'Université M'Hamed BOUGARA de Boumerdès (UMBB) comporte cinq Facultés et un Institut :

- Faculté des Sciences : La faculté des sciences comprend les six (06) départements suivants :

  Mathématiques, Physique, Chimie, Informatique, Biologie, Langues Etrangères, Cellule de Suivi (Sciences et Techniques des Activités Physiques et Sportives), Cellule Sciences Agronomiques, Cellule Infotronique
- Faculté de Technologie : La Faculté de Technologie comprend les huit (08) départements suivants : Génie des Procédés Industriels, Technologie Alimentaire, Génie de l'Environnement, Génie Mécanique, Energétique, Maintenance Industrielle, Génie des Matériaux, Génie Civil
- Faculté des Hydrocarbures et de la Chimie : La faculté des Hydrocarbures et de la Chimie comprend les six (06) départements suivants : Génie Parasismique, Géophysique et Phénomènes Aléatoires, Gisements Miniers et Pétroliers, Transport et Equipements des Hydrocarbures, Génie des Procédés Chimiques et Pharmaceutiques, Automatisation des

Procédés et Electrification, Economie et Commercialisation des Hydrocarbures, Transport et Equipements des Hydrocarbures.

- Faculté des Sciences Economiques, Commerciales et des Sciences de Gestion : Faculté des Sciences Economiques, Commerciales et des Sciences de Gestion comprend les trois (03) départements suivants : Sciences de Gestion, Sciences Commerciales, Sciences Economiques
- Faculté de Droit et Sciences Politiques : La faculté de droit comprend les trois (03) départements suivants : Droit Public, Droit Privé, Sciences Politiques, Cellule Langue et Littérature Arabes
- Faculté des Lettres et des Langues : La faculté des Lettres et des Langues comprend les trois (03) départements suivants : Lettres Arabes, Français, Anglais.
- Institut de Génie Electrique et Electronique : Institut de Génie Electrique et Electronique comprend les trois (03) départements suivants : Enseignement de Base, Electronique, Automatique et Electrotechnique. [31]

2. Situation actuelle

Pour guider les étudiants et pour répondre à leurs besoins en termes de renseignements, l'université actuellement dispose d'un site web pour la diffusion des informations.

La cellule de communication pilote le site web de l'UMBB, elle est chargée de :

- Contribution à l'actualisation et l'animation des contenus du site web notamment en rédigeant des actualités ou tout autre type de présentation
- Veille à la mise à jour régulière des informations des pages centrales du site Internet.
- Diffusion des informations via divers médias numériques et les réseaux sociaux
- Newsletter : Réalisation et diffusion de la newsletter à l'ensemble des personnels, elle est le principal moyen de communication interne.
- Mailing : Répondre aux besoins d'information par mail.
- Réseaux sociaux : Les réseaux sociaux sont un outil de communication stratégique, la présence de l'université sur certains réseaux sociaux permet de fait connaitre l'université, développer son image et d'être en contact continu non seulement avec de futurs étudiants ou étudiants déjà inscrits mais aussi avec toute autre personne externe. [32]

Les étudiants également peuvent se rapprocher vers l'administration en présentiel, pour soulever leurs questions et préoccupations ainsi que demander des renseignements.

3. Constats et critiques

- Les conseillers pédagogiques ne peuvent pas consacrer suffisamment de temps à guider les étudiants en raison de l'augmentation du nombre d'étudiants et du manque de temps. Donc c'est tout à fait évident que les étudiants passent de longues heures non seulement à scruter le site Web de l'université mais aussi dans la lecture du prospectus. Les étudiants s'énervent lorsqu'ils trouvent que le contenu des sites Web et des prospectus n'est pas concis et limpide. Leur colère s'ajoute, quand après avoir passé de longues heures à l'université site Web ou prospectus, leurs doutes ou questions n'ont pas été dissipés.
- Parfois les représentants humains fournissent des réponses vagues ou ne disent simplement : "Je ne sais pas, vérifiez auprès de quelqu'un d'autre."
- Avec des problèmes particuliers, les étudiants ont souvent besoin d'aide et ne savent pas à qui s'adresser
- Un autre problème courant lors des traitements des demandes par email est que les étudiants doivent attendre plusieurs jours pour avoir une réponse à leur demande. Ceci est principalement dû au nombre limité de personnes dans l'équipe universitaire par rapport au nombre de messages entrants

4. Solution proposée

Pour faire face aux problèmes cités dans le constat, nous proposons la mise en place d'un agent conversationnel qui va garantir ce qui suit :

- Informations correctes

  Trouver des informations sur l'inscription à l'université peut prendre énormément de temps pour les candidats. Il arrive régulièrement que certaines informations importantes (ex. délai pour rendre son dossier d'inscription, dates de la prochaine session d'inscription…), ne soient pas renseignées sur le site de l'université, ce qui rend le processus d'admission plus difficile (et pire, cela dissuade les candidats de soumettre la demande d'admission).

  Un chatbot permet de renseigner le candidat avec une information correcte ou de le diriger vers la page correspondant à sa demande.
- Réponses immédiates 24/7

  Les délais de réponses des équipes universitaires peuvent être assez long. Un chatbot est disponible 24/7, les candidats et étudiants obtiennent une réponse dans l'immédiat.
- Coûts de main-d'œuvre réduits

- Pour préparer les admissions et rentrées universitaires, beaucoup d'écoles & universités embauchent du personnel. Le chatbot peut gérer environ 70% des questions des candidats et élèves. Les interrogations qui ne peuvent pas être solutionnées par le bot pourront être envoyées directement aux équipes universitaires avec l'historique de la conversation afin d'avoir tout le contexte et les échanges entre le bot et le candidat ou étudiant. Ainsi, le traitement de la demande pourra être plus rapide.
- Il est aussi possible de prévoir une reprise en main en direct pour répondre en direct à l'étudiant tout en ayant accès à l'historique de la conversation
- Un chatbot peut répondre à plusieurs étudiants & candidats en même temps.
- Simplifier le processus d'inscription

  Les chatbots éducatifs peuvent également faciliter l'inscription. Peuvent répondre aux questions les plus courantes que le nouveau groupe d'étudiants pose chaque année. Ils peuvent fournir des informations sur le campus universitaire, les installations, les bourses et avantages, et le processus d'inscription.

L'utilisation des chatbots peut aider notre université à :

- Fournir un temps de réponse plus rapide : les milléniaux veulent des réponses rapides, et le déploiement d'un chatbot peut vous aider à vous assurer que leurs questions simples obtiennent une réponse en quelqes minutes.
- Fournir une assistance 24h/24 et 7j/7 : avec un chatbot, vous pouvez répondre aux questions et aux problèmes de vos candidats et étudiants à tout moment, 24h/24. Cela peut vous aider à améliorer leurs expériences et à les maintenir engagés.
- Réduire les coûts : Un chatbot peut gérer plusieurs demandes sans compromettre la qualité des interactions. Le fait qu'un bot interagisse avec vos candidats et vos étudiants 24 heures sur 24, 7 jours sur 7 peut vous aider à réduire le coût d'embauche d'un certain nombre de professionnels du service d'assistance.
- Les nouveaux étudiants peuvent se sentir perdus lorsqu'ils rejoignent l'université. Un chatbot peut servir de guide virtuel en fournissant des informations sur les différents départements, les services offerts par l'université, les ressources disponibles, les installations sur le campus, etc. Cela permet aux étudiants de s'orienter plus facilement et de trouver rapidement les réponses à leurs questions.

Conclusion

En conclusion, ce chapitre se concentre sur l'université en tant qu'institution d'enseignement supérieur et explore différents aspects clés tels que sa présentation, sa situation actuelle, les constats et critiques qui lui sont associés, ainsi que la solution proposée pour améliorer l'expérience des étudiants : création d'un chatbot.

En explorant les avantages et les applications potentielles d'un chatbot à l'université, nous mettrons en évidence comment cette technologie peut faciliter l'accès à l'information, offrir un soutien personnalisé et favoriser l'engagement des étudiants.

Chapitre 3 : Conception et réalisation

Introduction

Ce chapitre se concentre sur la conception et la réalisation d'un agent conversationnel, offrant un aperçu détaillé des étapes nécessaires pour créer un chatbot fonctionnel et interactif. Ce chapitre explore les aspects pratiques de la conception et de la réalisation d'un chatbot, en se basant sur les connaissances acquises dans les chapitres précédents.

Au cours de ces étapes, nous examinerons les différentes techniques, outils et méthodologies nécessaires pour créer un chatbot performant. De la collecte des données à la mise en place de l'interface utilisateur, chaque étape est cruciale pour garantir un chatbot efficace et adapté aux besoins des étudiants.

1. Choix de solution

Le chatbot, qu'il soit basé sur des règles ou basé sur l'intelligence artificielle (IA), Les deux approches ont leurs avantages et leurs limitations, dans le tableau de comparaison ci-dessous, nous examinons de plus près les principales caractéristiques et différences entre ces deux approches.

Chatbot basé sur des règles	**Chatbot basé sur l'intelligence artificielle**
Adapté pour un domaine spécifique et bien défini.	Peut s'adapter à différents domaines et apprendre de nouvelles informations.
Niveau de contrôle élevé car les réponses sont prédéfinies.	Moins de contrôle car il génère ses propres réponses.
Plus rapide à mettre en place.	Peut nécessiter plus de temps.
Moins flexible, ne peut pas apprendre de nouvelles informations.	Plus flexible, peut apprendre et s'adapter à de nouvelles situations.
Peut fournir des réponses précises dans les limites des règles prédéfinies.	Peut fournir des réponses plus personnalisées et adaptées aux besoins spécifiques des utilisateurs.
Plus faciles à mettre en œuvre et nécessitent moins de ressources.	Plus complexes à mettre en œuvre et nécessitent plus de ressources

	informatiques pour l'entrainement et l'inférence.

Tableau 1 Comparaison des chatbots basé sur des règles et des chatbots basé sur intelligence artificielle

Après avoir étudié les différentes approches disponibles, il a été décidé d'opter pour un chatbot hybride, combinant à la fois des éléments d'un chatbot basé sur des règles et des fonctionnalités d'un chatbot intelligent. Cette décision a été prise en considérant les avantages de cette approche qui répond aux problématiques essentielles de ce projet et elle permet la réalisation d'une solution sous les contraintes de ressources et durée du développement.

2. Planification de projet [46]

Nous avons planifié notre projet selon les étapes suivantes :

1) Collecte de données : Cette étape consiste à rassembler toutes les informations nécessaires pour le développement du chatbot.
2) Conception de l'architecture : Une fois les données collectées, il est essentiel de concevoir une architecture solide pour le chatbot.
3) Développement d'un modèle backend : Le modèle backend est responsable de l'analyse des messages entrants, de l'extraction des intentions et des entités, ainsi que de la génération de réponses appropriées.
4) Développement d'un modèle frontend : Une fois le modèle backend développé, il est nécessaire de concevoir et de développer l'interface utilisateur du chatbot. Cela inclut la création d'une interface web, pour interagir avec le chatbot.
5) Tests d'efficacité : cette étape consiste à effectuer des tests pour évaluer l'efficacité et la convivialité du chatbot.

3. Conception

Dans cette section, nous avons sélectionné quelques diagrammes pour présenter de manière visuelle et structurée notre conception de chatbot.

3.1 Diagramme de flux de données

Un diagramme de flux de données (DFD) est une représentation graphique qui illustre le flux d'informations dans un système. Il est utilisé pour modéliser visuellement les processus, les données et les interactions entre eux.

Dans ce diagramme, les symboles ou notations utilisés sont les rectangles, les cercles, les flèches, les lignes, etc.

L'interaction commence lorsque l'utilisateur pose une question. Cette requête est transmise au chatbot via son interface utilisateur graphique. La requête est envoyée à l'unité de traitement du langage naturel. Les données traitées ensuite transmises à l'unité des algorithmes d'apprentissage automatique. Cette unité utilise un modèle de réseau de neurones pour rechercher la réponse appropriée dans la base de connaissances. Une fois la réponse sélectionnée, elle est renvoyée à l'utilisateur.

Le diagramme montre également la présence d'un administrateur qui a la capacité de mettre à jour la base de connaissances.

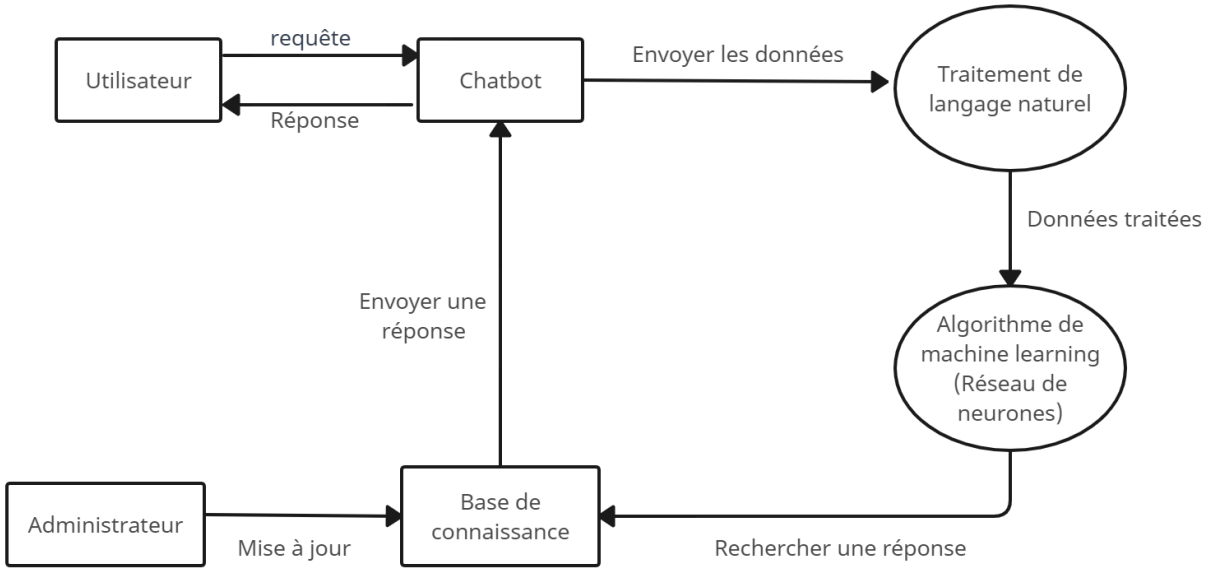

Figure 13 Diagramme de flux de données

3.2 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un type de diagramme UML (Unified Modeling Language), c'est une représentation graphique utilisée pour décrire les interactions entre les acteurs (utilisateurs ou systèmes externes) et un système donné. Il met l'accent sur les fonctionnalités du système du point de vue de l'utilisateur.

Ce diagramme représente les interactions essentielles entre les acteurs (utilisateur et administrateur) et le chatbot.

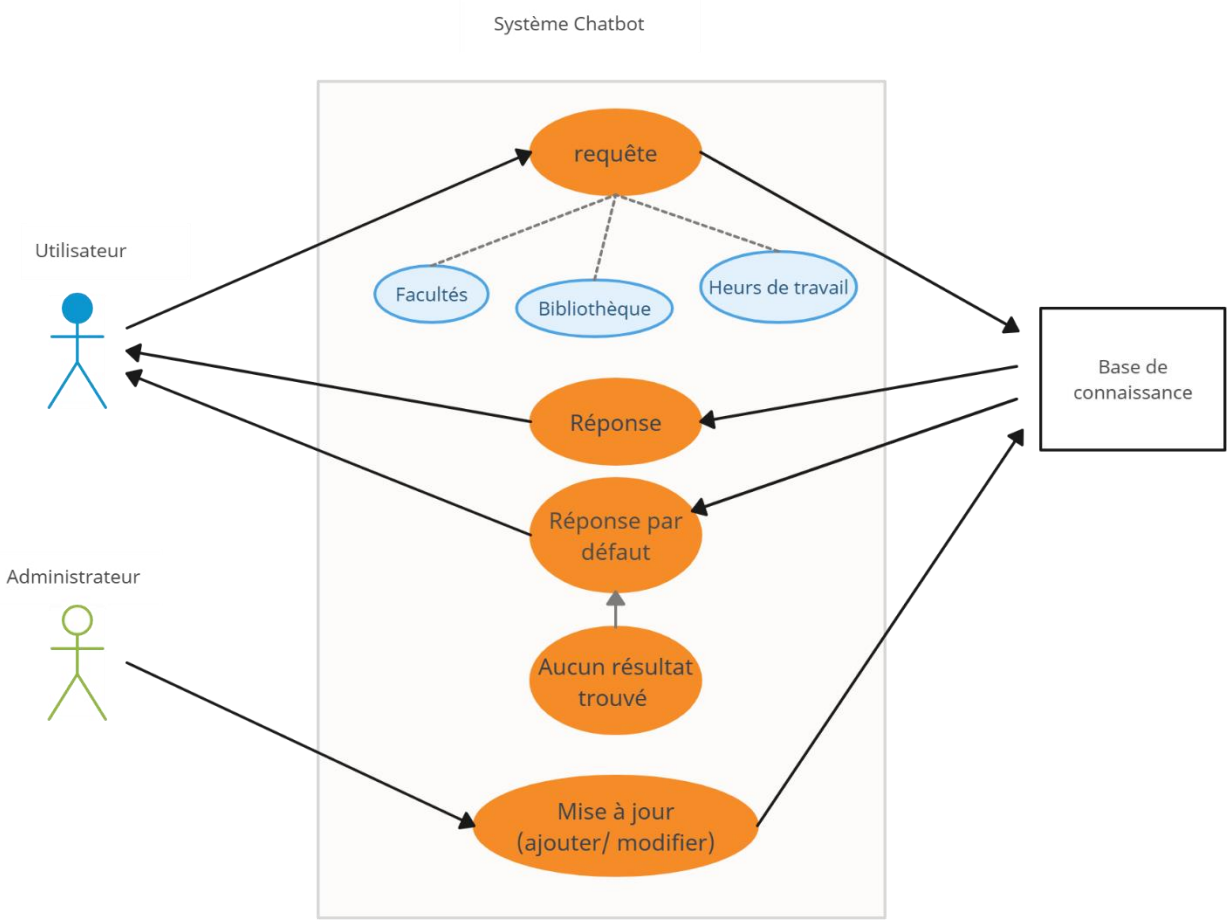

Figure 14 Diagramme de cas d'utilisation

3.3 Diagramme d'activités

Un diagramme d'activités est un autre type de diagramme UML, il est utilisé pour représenter le flux de contrôle et le déroulement des activités dans un système. Il est utilisé pour modéliser les étapes séquentielles, parallèles et conditionnelles d'un ensemble d'activités.

- Noeud de début : Le processus démarre lorsque l'utilisateur envoie une requête au chatbot.
- Entrée de l'utilisateur : Le chatbot reçoit la requête de l'utilisateur et la traite.
- Rechercher dans la base de connaissances : Le chatbot consulte sa base de connaissances pour trouver une réponse.
- Réponse correspondante : Si une réponse correspondante est trouvée, le chatbot génère la réponse appropriée.
- Réponse par défaut : Si aucune réponse correspondante n'est trouvée, le chatbot génère une réponse par défaut.
- Résultat : le chatbot fournit le résultat final à l'utilisateur.
- Noeud de fin : Le processus se termine.

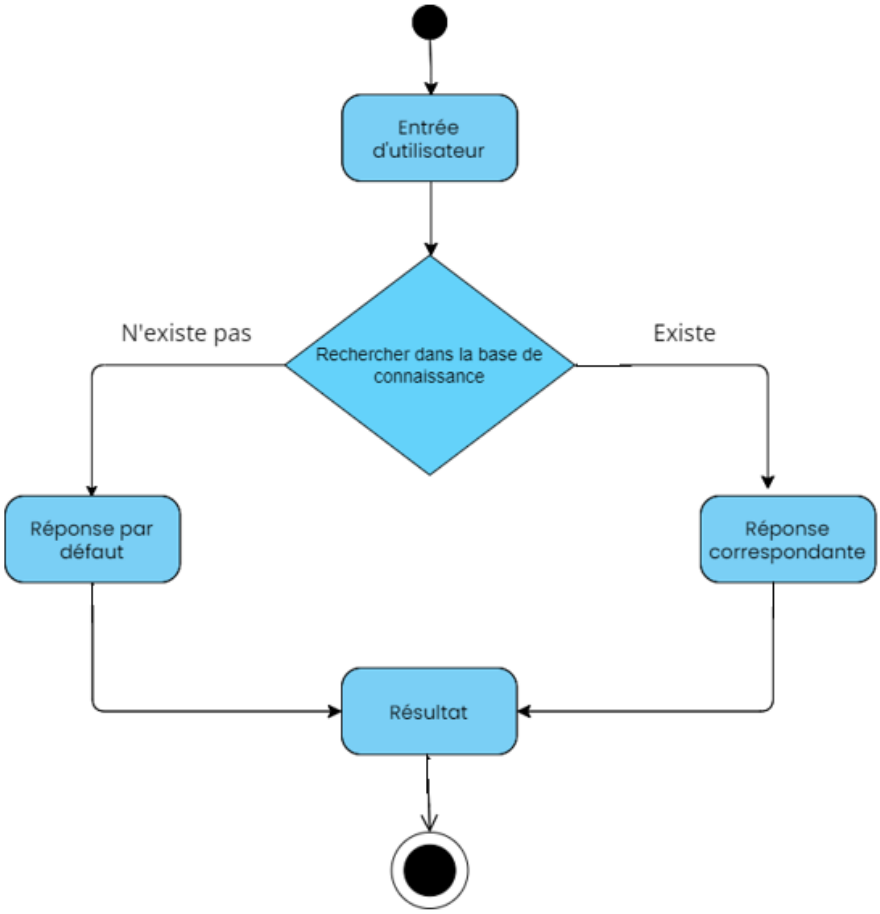

Figure 15 Diagramme d'activités

4. Outils et environnement de développement

Parler de l'implémentation revient à détailler l'aspect matériel, l'environnement de développement et les différents outils qui ont été utilisé pour réaliser l'application.

4.1 Aspect matériel

Notre projet a été développé sur un pc :

Processeur : Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz 1.80 GHz

RAM : 16,0 Go

Disque dure : 214 GO

Système d'exploitation : Windows 10

4.2 Logiciel de développement

Le chatbot a été implémentée sous l'environnement PyCharm (version 2023.1.1). PyCharm est un environnement de développement intégré utilisé pour programmer en Python.

PyCharme comporte de nombreux avantages. Son éditeur de code intelligent aide à écrire du code de haute qualité. Ses différents codes couleur pour les mots-clés, les classes et les fonctions augmentent la lisibilité et la compréhensibilité du code. Ceci simplifie aussi la détection des erreurs. [33]

4.3 Langage de programmation

L'application réalisée a été implémentée avec le langage Python version 3.11.2

Pourquoi Python ?

Python est un langage de programmation polyvalent qui a été utilisé pour tout, des sites Web et des applications mobiles aux applications de bureau et aux jeux. Il est également populaire dans la science des données, un domaine où il a gagné en importance.

Python a été conçu pour être un langage facile à utiliser et à lire, Python est un langage populaire pour l'apprentissage automatique. Python dispose d'un grand nombre de bibliothèques d'apprentissage automatique, et il est facile à apprendre et à utiliser. Il est également largement utilisé dans l'industrie, avec ses nombreuses applications, notamment la science des données et le calcul scientifique. Il est également utilisé dans le développement web, ce qui en fait une option intéressante pour les débutants qui souhaitent se lancer rapidement dans l'intelligence artificielle.[34]

JSON

JSON (JavaScript Object Notation) est un format de fichier textuel conçu pour l'échange de données. Il représente des données structurées basées sur un sous-ensemble du langage de programmation JavaScript. JSON apparaît dans des fichiers portant l'extension. json ou entre guillemets sous forme de chaînes de caractères ou d'objets affectés à une variable dans d'autres formats de fichiers.

JSON est une alternative simple et légère au langage de balisage extensif (XML). Comme la structure JSON est basée sur celle de JavaScript, ils partagent un certain nombre de similitudes.

Voici les principaux éléments de la syntaxe JSON :

- Les données sont présentées sous forme de paires clé/valeur.
- Les éléments de données sont séparés par des virgules.
- Les crochets {} désignent les objets.
- Les crochets [] désignent des tableaux. [35]

4.4 Bibliothèques utilisées

4.4.1 NLTK

Le NLTK, ou Natural Language Toolkit, est une suite de bibliothèques logicielles et de programmes. Elle est conçue pour le traitement naturel symbolique et statistique du langage anglais en langage Python. C'est l'une des bibliothèques de traitement naturel du langage les plus puissantes.

Cette suite d'outils rassemble les algorithmes les plus communs du traitement naturel du langage comme La tokenization, La méthode du Stemming, La technique de Lemmatisation. [36]

4.4.2 Numpy

NumPy (Numerical Python) est une bibliothèque de python qui comporte des fonctions permettant de manipuler des matrices ou tableaux multidimensionnels.

Les tableaux NumPy utilisent moins de mémoire et d'espace de stockage, ce qui le rend plus avantageux que les tableaux traditionnels de python. [37]

4.4.3 Tflearn

TFlearn est une bibliothèque d'apprentissage en profondeur modulaire et transparente construite sur Tensorflow. Il a été conçu pour fournir une API de niveau supérieur à TensorFlow afin de faciliter et d'accélérer les expérimentations, tout en restant totalement transparent et compatible avec elle. [38]

4.4.4 **Tensorflow**

TensorFlow est une bibliothèque de Machine Learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissage expérimentales et de les transformer en logiciels.

Cette bibliothèque permet notamment d'entraîner et d'exécuter des réseaux de neurones pour la classification de chiffres écrits à la main, la reconnaissance d'image, les plongements de mots, les réseaux de neurones récurrents, les modèles sequence-to-sequence pour la traduction automatique, ou encore le traitement naturel du langage. [39]

4.4.5 **Random**

random est un module Python regroupant plusieurs fonctions permettant de travailler avec des valeurs aléatoires. [40]

4.4.6 **Json**

Pour formater des données Python en JSON et inversement, On utilise le module Python standard json. [41]

4.4.7 **Flask**

Flask est un petit framework web Python léger, qui fournit des outils et des fonctionnalités utiles qui facilitent la création d'applications web en Python. [42]

5. Réalisation

5.1 Collecte et préparation des données d'entraînement pour l'agent conversationnel (Base de connaissance)

La tâche principale dans la création d'un chatbot est la construction de la base de connaissances, qui peut être interprétée comme le cerveau du chatbot. Elle est constituée d'un système question réponse stocké dans un fichier json qui contient des mots, des chiffres et des symboles pour ressembler à la structure utilisée dans les conversations humaines.

La base de connaissace de chatbot pour l'entraînement du réseau de neurones été créés dans un fichier séparé. Le modèle a utilisé les données du fichier intents.json , qui contient :

"tag" : un raccourci pour le sujet de la conversation.

"patterns" : l'entrée de l'utilisateur, plusieurs alternatives possibles pour ce sujet.

"réponses" : les réponses de chatbot.

"contexte" : un champ qui met en corrélation une entrée avec le champ "tag" d'une autre entrée, pour des interactions multiples avec l'utilisateur. [43][44]

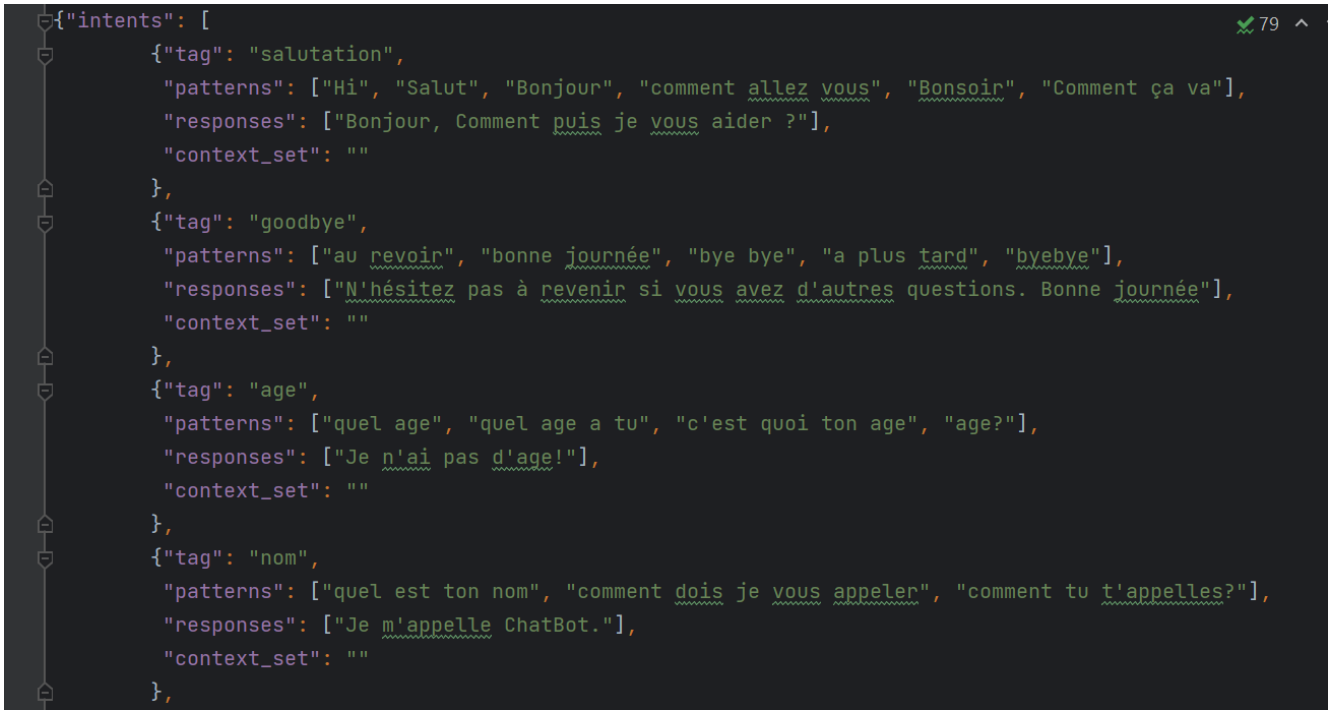

Figure 16 Base de connaissance

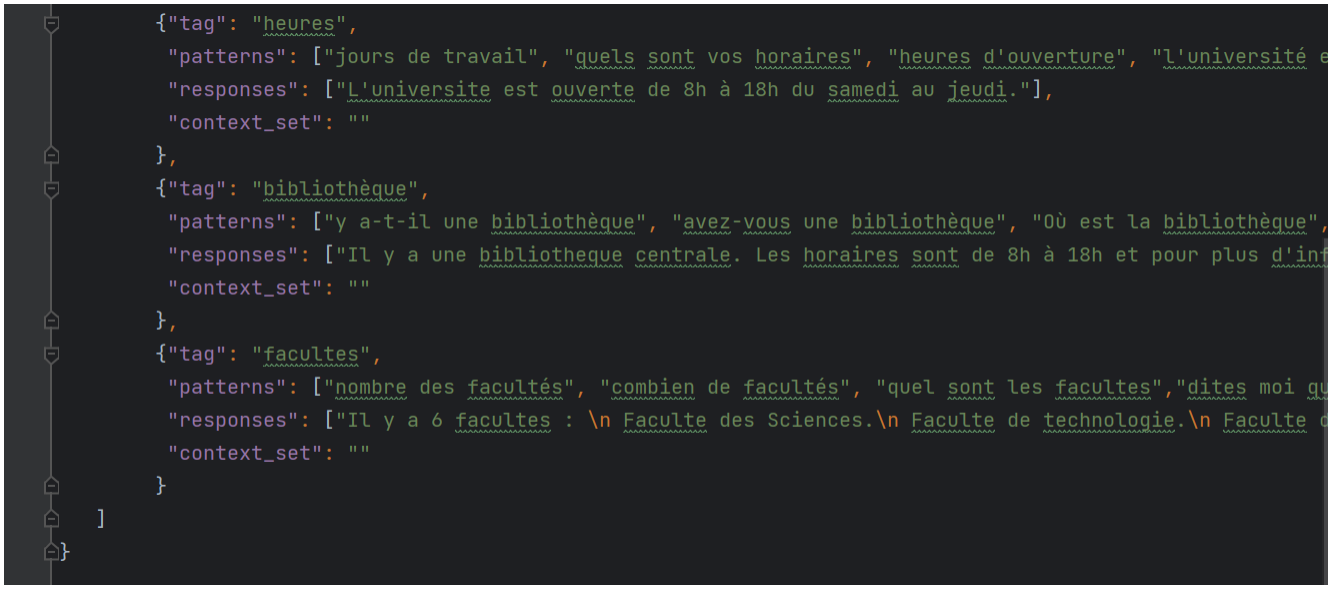

Figure 17 Base de connaissance (suite)

5.2 Prétraitement de données

Afin de créer nos données d'entraînement, nous devons d'abord effectuer certaines opérations sur nos données telles que :

a) Chargement de nos données JSON

b) Créer un vocabulaire de tous les mots utilisés dans les patterns.

c) Créer une liste des classes, Il s'agit simplement des tags de chaque intention.

d) Créer une liste de tous les patterns dans le fichier des intentions.

e) Créer une liste de tous les tags associés à chaque pattern dans le fichier intents.

```
import nltk
from nltk.stem.lancaster import LancasterStemmer
stemmer = LancasterStemmer()

import numpy
import tflearn
import tensorflow
import random



import json
with open('intents.json') as file:
    data = json.load(file)
```

Figure 18 Chargement de données JSON - code python

```python
words = []
labels = []
docs_x = []
docs_y = []

for intent in data['intents']:
    for pattern in intent['patterns']:
        wrds = nltk.word_tokenize(pattern)
        words.extend(wrds)
        docs_x.append(wrds)
        docs_y.append(intent["tag"])

    if intent['tag'] not in labels:
        labels.append(intent['tag'])


print(f'words = {words}')
print(f'labels = {labels}')
print(f'doc_x = {docs_x}')
print(f'doc_y = {docs_y}')
```

Figure 19 Tokenisation de données - code python

La liste words contient tous les mots après la tokenisation (Division du texte en mots) des patterns, la liste labels contient les tous tags, la liste doc_x contient tous les patterns tokenisé et la liste doc_y contient les tags corresponds à chaque pattern, Voici à quoi ressemble chaque liste :

```
words = ['Hi', 'Salut', 'Bonjour', 'comment', 'allez', 'vous', 'Bonsoir', 'Comment', 'ca', 'va', 'au', 'revoir', 'bonne', 'journÃ©e', 'bye', 'by
labels = ['greeting', 'goodbye', 'age', 'name', 'facultes']
doc_x = [['Hi'], ['Salut'], ['Bonjour'], ['comment', 'allez', 'vous'], ['Bonsoir'], ['Comment', 'ca', 'va'], ['au', 'revoir'], ['bonne', 'journÃ
doc_y = ['greeting', 'greeting', 'greeting', 'greeting', 'greeting', 'greeting', 'goodbye', 'goodbye', 'goodbye', 'goodbye', 'age', 'age', 'age'
```

Figure 20 Résultats de la tokenisation

Les mots extraits sont normalisés en les convertissant en lettres minuscules et en appliquant la racinisation (stemming). Les mots uniques sont ensuite triés et stockés dans la liste words.

```python
words = [stemmer.stem(w.lower()) for w in words if w != "?"]
words = sorted(list(set(words)))

labels = sorted(labels)
```

Figure 21 Racinisation de données - code python

Cependant, les réseaux de neurones s'attendent à recevoir des valeurs numériques, et non des mots. Nous devons donc d'abord traiter nos données pour qu'un réseau de neurones puisse lire ce que nous faisons.

Pour l'encodage des mots on va utiliser la méthode bag of words.

Ce code prend la liste (docs_x) et leurs étiquettes correspondantes (docs_y), puis crée des sacs de mots (représentations binaires) pour chaque document. Les sacs de mots et les sorties attendues sont ensuite stockés dans les listes "training" et "output" respectivement.

```
training = []
output = []

out_empty = [0 for _ in range(len(labels))]

for x, doc in enumerate(docs_x):
    bag = []

    wrds = [stemmer.stem(w.lower()) for w in doc]

    for w in words:
        if w in wrds:
            bag.append(1)
        else:
            bag.append(0)

    output_row = out_empty[:]
    output_row[labels.index(docs_y[x])] = 1

    training.append(bag)
    output.append(output_row)
```

Figure 22 Encodage de données - code python

Voici à quoi ressemble chaque pattern après l'encodage des mots avec la méthode bag of words :

```
doc = ['Bonsoir']
bag = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
output_row = [0, 0, 0, 1, 0]
doc = ['Comment', 'ca', 'va']
bag = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
output_row = [0, 0, 0, 1, 0]
doc = ['au', 'revoir']
bag = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
output_row = [0, 0, 1, 0, 0]
doc = ['bonne', 'journee']
bag = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
output_row = [0, 0, 1, 0, 0]
doc = ['bye', 'bye']
bag = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
output_row = [0, 0, 1, 0, 0]
doc = ['a', 'plus', 'tard']
bag = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
output_row = [0, 0, 1, 0, 0]
```

Figure 23 Données encodées

Maintenant, on utilise la bibliothèque NumPy pour convertir les listes "training" et "output" en tableaux NumPy. La conversion des données en tableaux NumPy facilite l'intégration des données dans les bibliothèques TensorFlow et tflearn qu'on va utiliser pour l'entraînement.

```
training = numpy.array(training)
output = numpy.array(output)
```

Figure 24 Conversion en tableaux numpy - code python

5.3 Entrainement de données

Après avoir converti nos données au format numérique, nous pouvons maintenant construire un modèle de réseau de neurones.

L'architecture du réseau de neurones comporte 4 couches, une couche d'entrée avec un nombre de neurones égale au nombre de tous les mots, deux couches cachées, chaque couche est définie avec un nombre de neurones égale à 8 et une fonction d'activation ReLU, et une couche de sortie avec un nombre de neurones égal au nombre des classes. Chaque couche cachée est entièrement connectée à la couche précédente. La fonction d'activation "softmax" est utilisée dans la couche de sortie pour obtenir des probabilités pour chaque classe de sortie.

Le modèle est réglé pour minimiser l'erreur. Où la ligne (tflearn.regression(net)) spécifie comment l'erreur est calculée et comment les poids du réseau sont ajustés pendant l'apprentissage.

```
tensorflow.compat.v1.reset_default_graph()

net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 8, activation="relu")
net = tflearn.fully_connected(net, 8, activation="relu")
net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
net = tflearn.regression(net)

model = tflearn.DNN(net)


model.fit(training, output, n_epoch=1000, batch_size=8, show_metric=True)
model.save("model.tflearn")
```

Figure 25 réseau de neurones - code python

5.4 Création de chatbot

Nous avons entraîné notre modèle de réseau de neurones, nous devons maintenant créer les fonctions réelles qui nous permettraient d'utiliser notre modèle.

On a créé une fonction (bag_of_words) qui prend une chaîne de caractères s, et renvoie le sac de mots sous forme d'un tableau numpy, cette fonction divise la chaîne de caractères s en mots individuel (tokenisation) s_words, applique la racinisation (stemming) et la mise en minuscule à chaque mot, et les convertir en sac de mots

```
def bag_of_words(s, words):
    bag = [0 for _ in range(len(words))]

    s_words = nltk.word_tokenize(s)
    s_words = [stemmer.stem(word.lower()) for word in s_words]

    for se in s_words:
        for i, w in enumerate(words):
            if w == se:
                bag[i] = 1

    return numpy.array(bag)
```

Figure 26 Fonction de prétraitement - code python

La fonction chat utilise le modèle entraîné pour prédire une réponse en utilisant la fonction predict sur l'entrée convertie en sac de mots à l'aide de la fonction bag_of_words. La fonction predict renvoie une liste de probabilités pour chaque classe (tag) possible, trouve la classe ayant la

probabilité la plus élevée et renvoie une réponse aléatoire parmi les réponses associées à l'intention correspondante.

```python
def chat(inp):

        results = model.predict([bag_of_words(inp, words)])
        results_index = numpy.argmax(results)
        tag = labels[results_index]
        proba = results[0][results_index]
        if proba < 0.5 :
            return "Je suis désolé, je ne comprends pas votre question. Pourriez-vous la reformuler ?"

        for tg in data["intents"]:
            if tg['tag'] == tag:
                responses = tg['responses']

        return random.choice(responses)
```

Figure 27 fonction de prédiction de réponse - code python

5.5 L'interface utilisateur du chatbot

On a créé l'interface de l'utilisateur avec le framework Flask en utilisant une combinaison de HTML, CSS et JavaScript pour permettre aux étudiants de communiquer avec le chatbot. Flask offre une structure flexible pour gérer les requêtes HTTP et afficher des pages HTML dynamiques. Pour interagir avec le chatbot, un bouton discret en bas à droite de la page, qui permet aux utilisateurs de déclencher l'apparition de l'interface du chatbot lorsqu'ils cliquent dessus.

Lorsqu'un utilisateur visite la page web, le bouton du chatbot est visible en permanence, mais sans être envahissant. Cela permet aux utilisateurs de naviguer sur le site sans être perturbés par l'interface du chatbot. Cependant, lorsqu'ils souhaitent interagir avec le chatbot, ils peuvent simplement cliquer sur le bouton pour faire apparaître l'interface du chatbot.

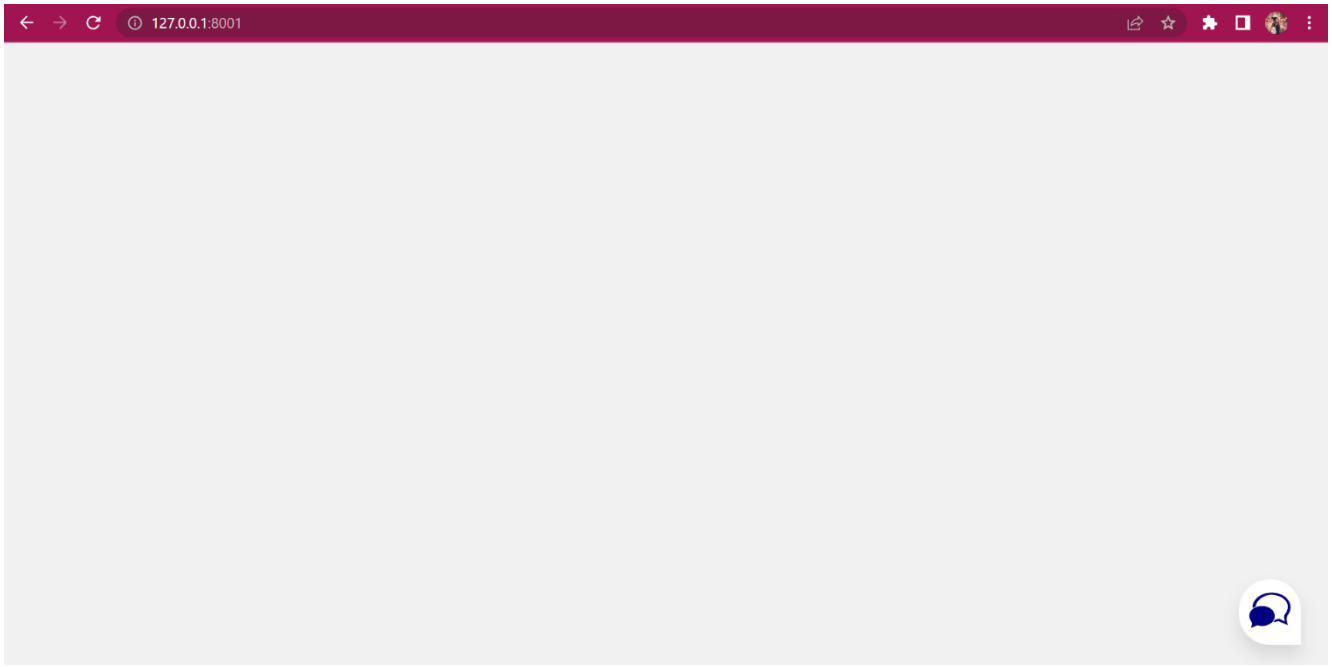

Figure 28 L'interface principale

Lorsque le bouton est activé, l'interface du chatbot s'affiche sous la forme d'une fenêtre de discussion. Les utilisateurs peuvent alors saisir leurs messages, poser des questions, et recevoir des réponses du chatbot.

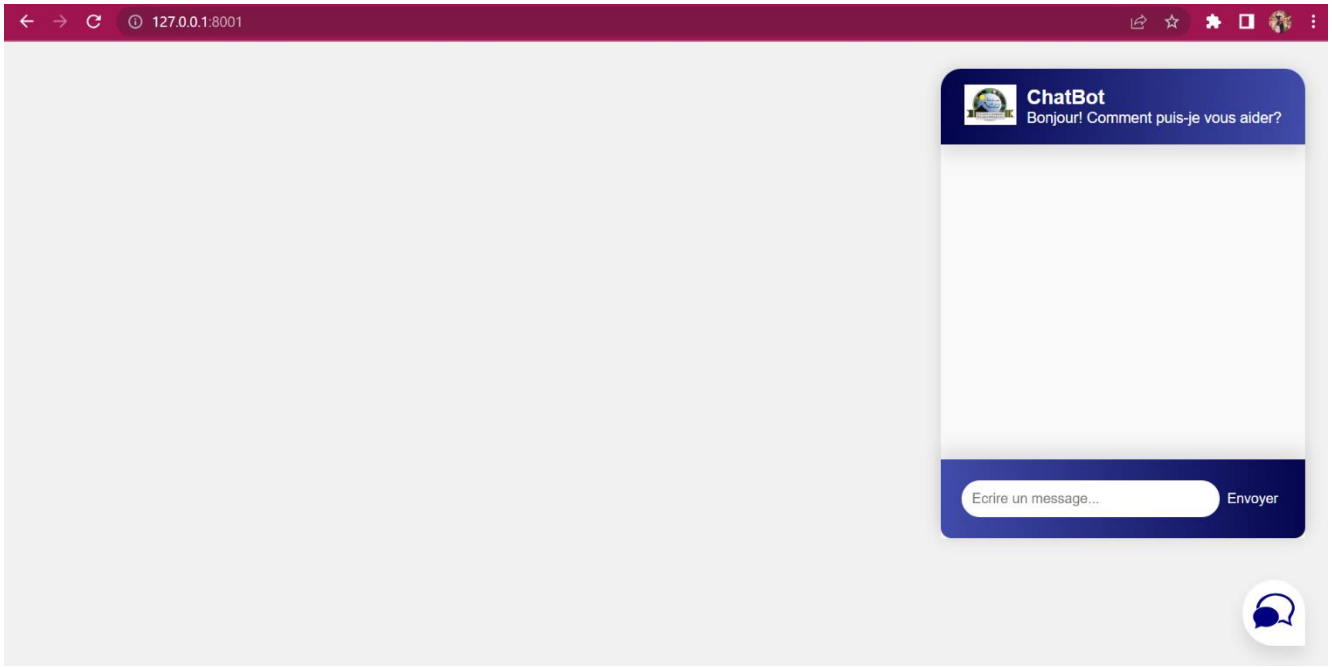

Figure 29 L'interface principale de chatbot

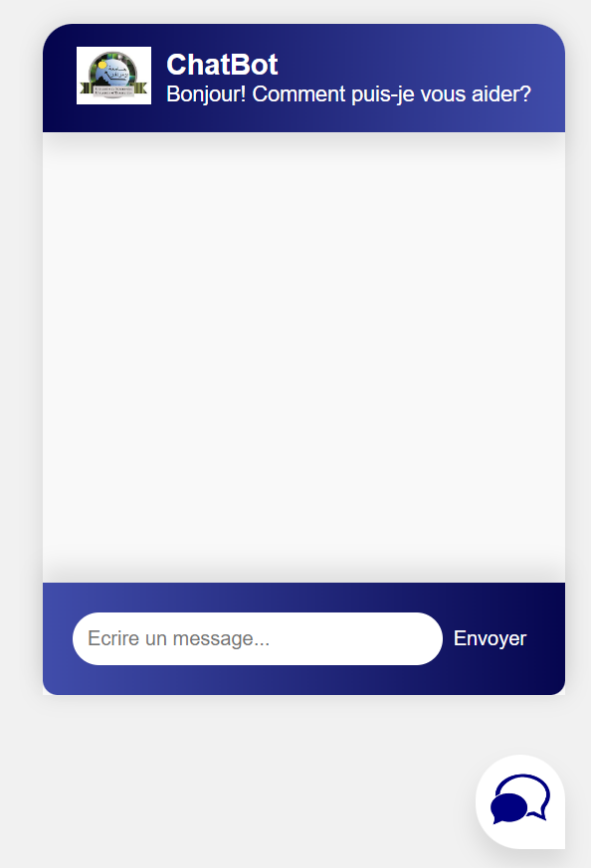

Figure 30 Version zoomée de l'interface principale de chatbot

5.6 Prototype de l'interface administrateur

L'interface administrateur est un élément essentiel pour gérer et configurer la base de connaissance du chatbot. Elle offre aux administrateurs la possibilité d'ajouter, modifier et supprimer les intentions, les patterns et les réponses qui composent la base de connaissances du chatbot.

Cela permet de maintenir et d'améliorer en permanence la base de connaissances du chatbot afin de répondre aux besoins changeants des utilisateurs.

On a créé l'interface administrateur avec le framework Flask en utilisant HTML pour permettre aux administrateurs d'ajouter nouvelles catégories.

- Lorsque vous accédez à l'interface, vous êtes accueilli par un formulaire simple et intuitif.
- Vous pouvez saisir le nom de la catégorie que vous souhaitez ajouter dans le champ prévu à cet effet, les questions et les réponses.
- Une fois que vous avez entré les informations, il vous suffit de cliquer sur le bouton "Ajouter" pour l'ajouter à la liste des catégories existantes.

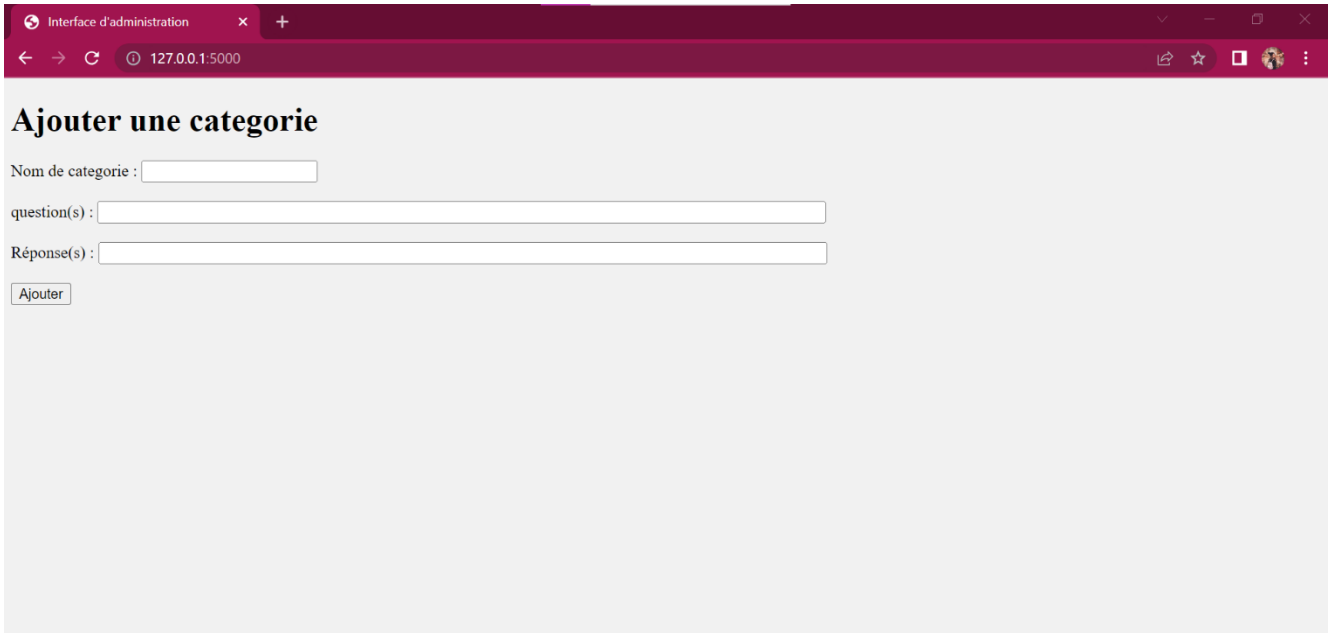

Figure 31 Prototype de l'interface administrateur

6. Tests de performance et d'efficacité de l'agent conversationnel

L'agent conversationnel développé a été testé à travers diverses discussions pour évaluer sa capacité à interagir avec les utilisateurs de manière efficace. Voici quelques exemples de discussions avec l'agent conversationnel :

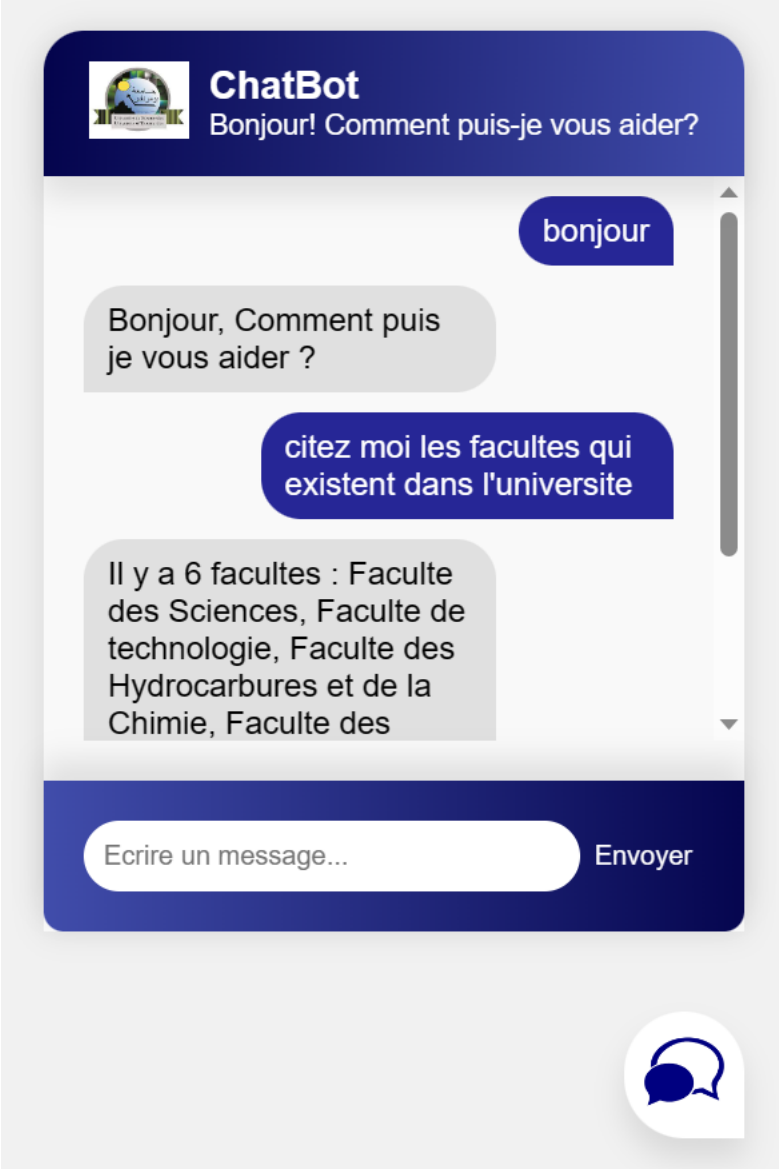

Figure 32 Exemple 1 de discussion avec l'agent conversationnel

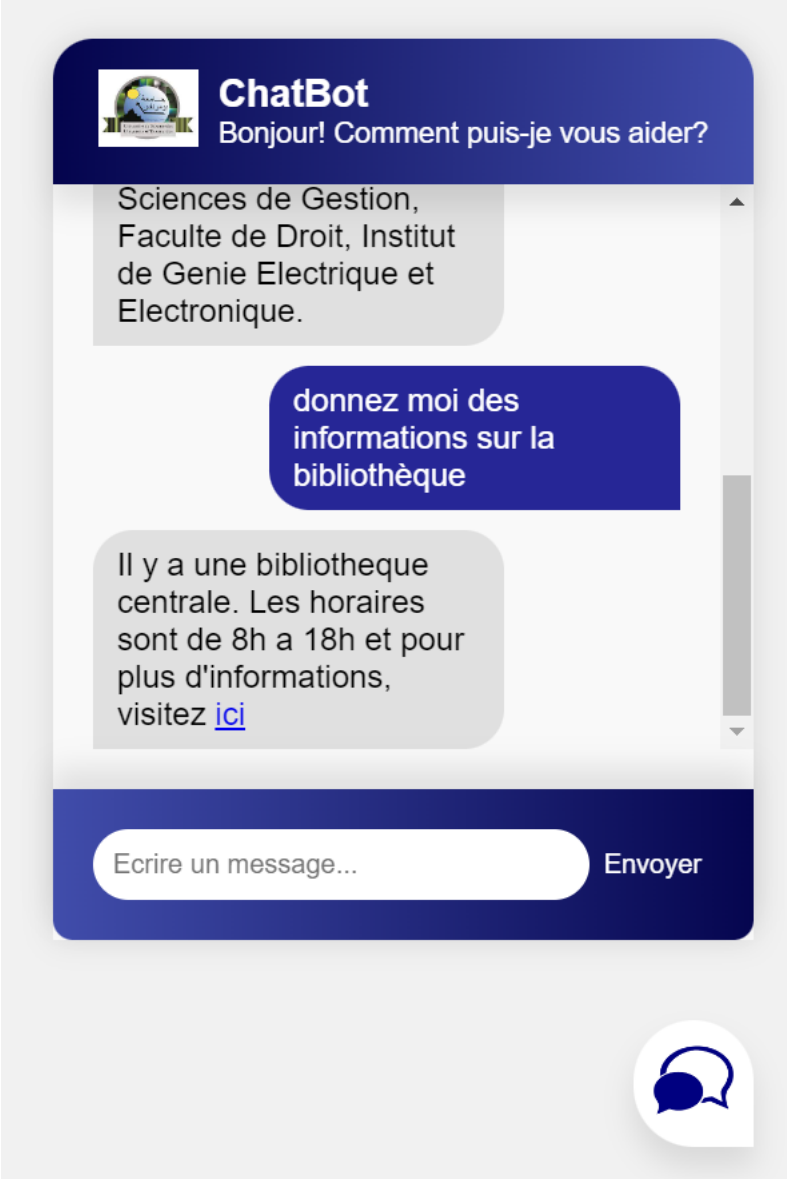

Figure 33 Exemple 2 de discussion avec l'agent conversationnel

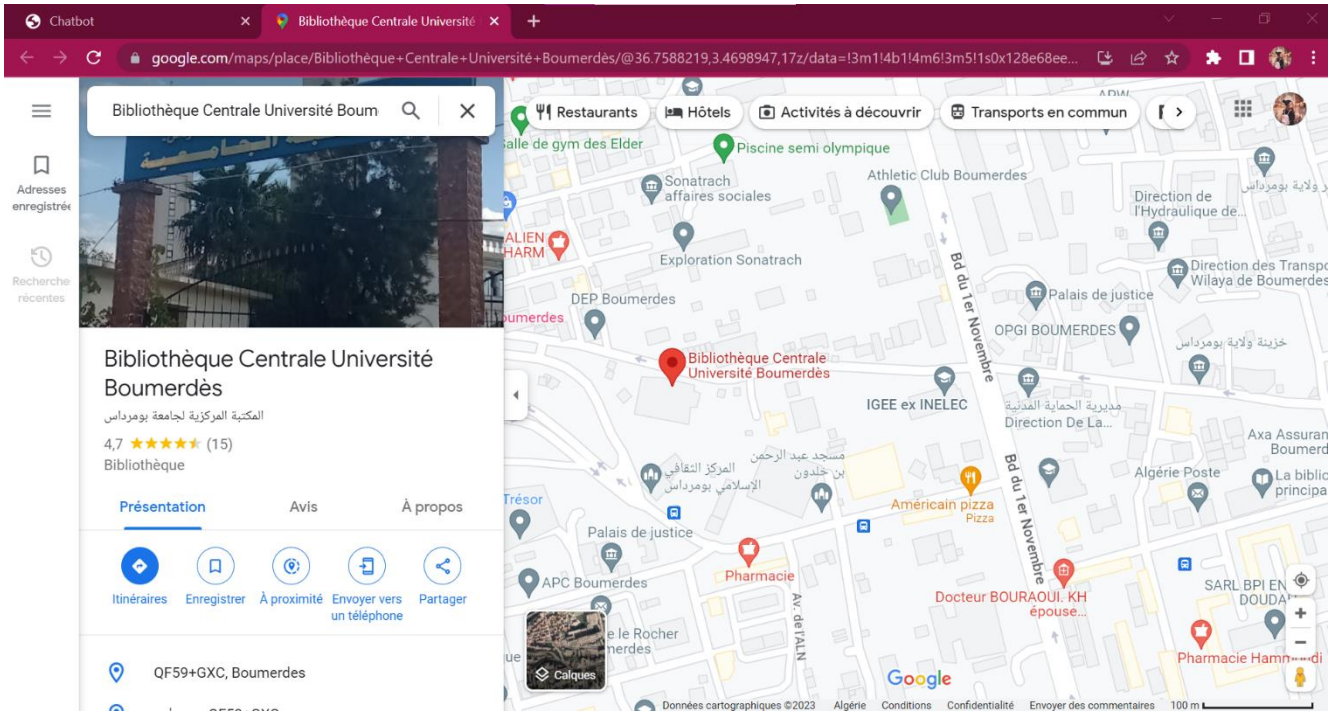

Figure 34 Exemple 3 de discussion avec l'agent conversationnel

Conclusion

En conclusion ce chapitre a exploré en détail les étapes de conception, de réalisation et de création d'un agent conversationnel performant, en couvrant différentes étapes essentielles du processus.

Conclusion Générale

Notre projet consiste à concevoir et mettre en œuvre un agent conversationnel pour les étudiants de l'UMBB.

Ce mémoire a abordé ces différentes phases à travers trois chapitres clés : l'étude bibliographique, l'étude de l'existant, et la conception et réalisation.

Dans le premier chapitre, L'étude bibliographique a permis d'explorer en profondeur les différentes approches, techniques et technologies utilisées dans le domaine des agents conversationnels. Et leur utilisation dans le domaine de l'éducation. Cette analyse a fourni une base solide de connaissances et de meilleures pratiques pour guider la conception et le développement de l'agent conversationnel.

Dans le deuxième chapitre, à travers une analyse de la situation actuelle, nous avons identifié les lacunes et les défis auxquels étaient confrontés les étudiants dans l'accès aux informations et aux ressources de l'université. En réponse à ces problèmes, nous avons proposé la mise en place d'un chatbot qui offrirait des informations correctes et des réponses immédiates, tout en réduisant les coûts de main-d'œuvre et en simplifiant le processus d'inscription.

Finalement, dans le troisième chapitre, nous sommes parvenus à réaliser notre agent conversationnel simple. Ce chapitre a été consacré à la conception et au développement de l'agent conversationnel, en mettant l'accent sur sa simplicité et son fonctionnement de base.

Bien que notre agent conversationnel soit actuellement simple dans sa portée et ses fonctionnalités, il est principalement axé sur un système questions-réponses. Cependant, il représente une première étape prometteuse dans l'amélioration de la communication avec les étudiants de l'UMBB. Il offre un moyen pratique et efficace d'obtenir des informations de base et de répondre aux questions courantes.

Sur le plan technologique, le thème qui nous a été attribué est très intéressant. Nous en tant qu'étudiants en fin d'études Il nous a permis de :

- Accroitre nos connaissances.
- Initier aux différentes technologies de l'intelligence artificielle (ML, ANN . . .)
- Améliorer nos compétences dans le langage de programmation python.

Enfin, Ce chatbot est encore à son stade primitif. Voici des améliorations qui peuvent être apportées à l'avenir.

- Élargir la base de connaissance : Pour améliorer les performances du chatbot, il est important de compléter et d'enrichir régulièrement sa base de connaissances. Cela implique d'ajouter de nouvelles questions et réponses, de mettre à jour les informations existantes et d'intégrer des ressources supplémentaires pertinentes. Une base de connaissances solide permettra au chatbot de répondre à un plus large éventail de questions posées par les étudiants.
- Améliorer l'interface administrateur : des améliorations peuvent être déployés pour rendre l'interface encore plus intuitive, conviviale et attrayante visuellement, Ajouter des fonctionnalités de modification et de suppression : En plus de l'ajout de catégories, On peut ajouter des fonctionnalités pour modifier ou supprimer les catégories existantes.
  Ajouter une fonctionnalité d'authentification. Cela permettra de contrôler l'accès à la page et de garantir que seuls les administrateurs autorisés peuvent y accéder.
- Intégration de l'agent conversationnel : L'intégration du chatbot au site Web existant de l'université permettra aux étudiants de l'utiliser de manière transparente lors de la recherche d'informations en ligne.
- Gestion de la base de connaissance : Stocker et organiser les questions posées par les étudiants auxquelles le chabot n'a pas trouvé une réponse dans sa base de connaissance, cela va permettre à l'administrateur de les visualiser afin d'ajouter les réponses correspondantes et les stocker.

Pour conclure, ce mémoire a permis de mettre en évidence l'importance des agents conversationnels dans le contexte éducatif, en particulier pour les étudiants de l'UMBB.

Références

[1] Khouzaimi, H. *Les agents conversationnels avec python (chatbot).* https://www.stat4decision.com/fr/les-agents-conversationnels-avec-python-chatbot/ (Dernière consultation : 15 mars 2023)

[2] Kharbouch, A. *Chatbots : définition, enjeux et bonnes pratiques 2023.* https://blog.smart-tribune.com/fr/chatbots (Dernière consultation : 15 mars 2023)

[3] Alburger, J. *Rule-Based Chatbots vs. AI Chatbots : Key Differences. hubtype.* https://www.hubtype.com/blog/rule-based-chatbots-vs-ai-chatbots (Dernière consultation : 17 mars 2023)

[4] Schmitt, S. (2021). *Chatbot & IA : prêt pour une expérience conversationnelle ?* https://one-inside.com/fr/chatbot-ia-experience-conversationnelle (Dernière consultation : 15 mars 2023)

[5] Patel, S., & Snigdha. (2022). *5 Different Types of Chatbots for Business Growth. REVE Chat.* https://www.revechat.com/blog/types-of-chatbot/ (Dernière consultation : 22 mars 2023)

[6] BRAGG, G. *How Do AI Chatbots Work : Exploring the Basics.* https://www.webio.com/blog/how-ai-chatbots-work (Dernière consultation : 22 mars 2023)

[7] Mainstay. (2022, March 4). *Georgia State University Chatbot Supports Every Student.* https://mainstay.com/case-study/how-georgia-state-university-supports-every- student-with-personalized-text-messaging/ (Dernière consultation : 27 mars 2023)

[8] ASU Enterprise Technology. *ASU's Experience Center chatbot supports the Sun Devil community.* https://tech.asu.edu/features/experience-center-chatbot (Dernière consultation : 27 mars 2023)

[9] Stanford University. (2019, May 16). *For learning, a chatbot that teaches beats flashcards.* https://news.stanford.edu/2019/05/08/learning-chatbot-teaches-beats-flashcards/ (Dernière consultation : 27 mars 2023)

[10] UC IT Blogger. (2021, June 23). *UC Davis Launches Service Portal Chat Bot. UC IT Blog.* https://cio.ucop.edu/uc-davis-launches-service-portal-chat-bot/ (Dernière consultation : 27 mars 2023)

[11] microsoft. (2023, March 2). *Intelligence artificielle : tout ce qu'il faut savoir.* https://experiences.microsoft.fr/articles/intelligence-artificielle/comprendre-utiliser-intelligence-artificielle/ (Dernière consultation : 20 mai 2023)

[12] SAP. Qu'est-ce que le Machine Learning ? https://www.sap.com/suisse/products/artificial-intelligence/what-is-machine-learning.html (Dernière consultation : 5 avril 2023)

[13] Nathalie. (2022, June 10). *Les principaux algorithmes de machine learning et processus de développement d'un modèle.* https://entreprises-particuliers.fr/les-principaux-algorithmes-de-machine-learning-et-processus-de-developpement-dun-modele/ (Dernière consultation : 5 avril 2023)

[14] Jvc, J. (2022). *Introduction au Machine Learning avec Python.* https://www.data-transitionnumerique.com/machine-learning-python/ (Dernière consultation : 21 mai 2023)

[15] Martin Heller. (2019, Septembre 28). *IA : Tout savoir sur l'apprentissage semi-supervisé.* https://www.lemondeinformatique.fr/actualites/lire-datastax-integre-l-ia-generative-de-thirdai-a-ses-bases-de-donnees-90516.html (Dernière consultation : 21 mai 2023)

[16] DataScientest (2022, December 23). *Machine Learning : Définition, fonctionnement, utilisations.* https://datascientest.com/machine-learning-tout-savoir (Dernière consultation : 21 mai 2023)

[17] Lutkevich, B., & Burns, E. (2023). *natural language processing (NLP).* https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP (Dernière consultation : 6 avril 2023)

[18] Belaidi, N. (2022, Janvier 31). *Word Embedding & NLP : définition, exemples.* https://blent.ai/blog/a/word-embedding-nlp-definition (Dernière consultation : 6 avril 2023)

[19] Crochet-Damais, A. (2022). *Réseau de neurones artificiels : réseaux neuronaux pour l'IA.* https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501851-reseau-de-neurones-artificiels/ (Dernière consultation : 10 avril 2023)

[20] Mestan, A. (2008). Introduction aux Réseaux de Neurones Artificiels Feed Forward. https://alp.developpez.com/tutoriels/intelligence-artificielle/reseaux-de-neurones/ (Dernière consultation : 10 avril 2023)

[21] Rod. (2022, August 30). *Comprendre les réseaux de neurones.* https://moncoachdata.com/blog/comprendre-les-reseaux-de-neurones/ (Dernière consultation : 15 mai 2023)

[22] Antoine Krajnc. *Qu'est-ce qu'un réseau de neurones en Deep Learning.* https://www.jedha.co/formation-ia/reseau-neurones-deep-learning (Dernière consultation : 15 mai 2023)

[23] Keldenich, T. (2022). *Fonction d'activation, comment ça marche ?* https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/#ReLU (Dernière consultation : 15 mai 2023)

[24] statsoft. *Réseaux de Neurones.* https://www.statsoft.fr/concepts-statistiques/reseaux-de-neurones-automatises/reseaux-de-neurones-automatises.php (Dernière consultation : 16 mai 2023)

[25] Bhardwaj, R. (2022, April 19). *What is AIML (Artificial Intelligence Markup Language) »* https://networkinterview.com/aiml/ (Dernière consultation : 9 mai 2023)

[26] Bot Libre. *AIML.* https://fr.botlibre.com/manual-aiml.jsp (Dernière consultation : 10 mai 2023)

[27] RiveScript. *Tutorial.* https://www.rivescript.com/docs/tutorial (Dernière consultation : 23 mai 2023)

[28] Maignan, I., & Séguéla, P. (2018). *La base de connaissances d'un chatbot : la clef de son intelligence ?* https://www.maddyness.com/2018/11/21/base-connaissance-chatbot-clef-intelligence/ (Dernière consultation : 8 mai 2023)

[29] Niharika Srivastava. (August 29, 2022) *Machine Learning Algorithms for teaching AI Chatbots.* https://www.e2enetworks.com/blog/machine-learning-algorithms-for-teaching-ai-chatbots (Dernière consultation : 8 mai 2023)

[30] Leah. (2020, août 24). *5 formidables exemples de chatbots interface utilisateur.* https://www.userlike.com/fr/blog/chatbot-interface-utilisateur (Dernière consultation : 9 mai 2023)

[31] Daci, S. *Présentation de l'université de Boumerdes*. https://www.univboumerdes.dz/universit%C3%A9/presentation.html (Dernière consultation : 15 avril 2023)

[32] Daci, S. *Présentation de la Cellule de Communication - Université de M'hamed Bougara Boumerdes.* https://www.univ-boumerdes.dz/Cellule-de-Communication/Presentation-Cellule.html (Dernière consultation : 15 avril 2023)

[33] Alexandre. (2023, February 20). *PyCharm : tout savoir sur l'IDE Python le plus populaire.* https://datascientest.com/pycharm (Dernière consultation : 20 mai 2023)

[34] IA School. (2023, 24 mars). *Pourquoi utiliser Python pour l'intelligence artificielle ?* https://www.intelligence-artificielle-school.com/ia-python/ (Dernière consultation : 20 mai 2023)

[35] Krimi, R. (2023, March 8). *Comprendre JSON : Syntaxe, Stockage et Exemples.* https://www.hostinger.fr/tutoriels/quest-ce-que-json (Dernière consultation : 20 mai 2023)

[36] Rédac, T. (2023, 29 mars). *NTLK : guide de l'outil de Traitement Naturel du Langage en Python.* https://datascientest.com/nltk (Dernière consultation : 21 mai 2023)

[37] Jvc, J. (2022). *Maîtrisez l'analyse des données avec NumPy Python.* https://www.data-transitionnumerique.com/numpy-python (Dernière consultation : 21 mai 2023)

[38] Thierry Perret. *Bibliothèque TFLearn.* https://morioh.com/p/9c5b568991f0 (Dernière consultation : 21 mai 2023)

[39] L, B. (2018). *TensorFlow : tout savoir sur la bibliothèque Machine Learning open source.* https://www.lebigdata.fr/tensorflow-definition-tout-savoir (Dernière consultation : 21 mai 2023)

[40] Christophe Hirschi. *Bibliothèques python* https://he-arc.github.io/livre-python/random/index.html (Dernière consultation : 21 mai 2023)

[41] Pierre Giraud. (2019, August 29). *L'échange de données en Python avec le module Json* https://www.pierre-giraud.com/python-apprendre-programmer-cours/echange-donnee-module-json/ (Dernière consultation : 21 mai 2023)

[42] Alexandre. (2022, November 22). *Flask – Un des frameworks les plus populaires de Python.* https://datascientest.com/avantages-et-fonctionnement-de-flask (Dernière consultation : 21 mai 2023)

[43] Haller, C. (2020, 7 juillet). *How to train your NLP chatbot.* https://www.christianhaller.me/blog/projectblog/2020-07-07-How-to-train-your-NLP-Chatbot/ (Dernière consultation : 25 mai 2023)

[44] Patel, N. P., Parikh, D. R., Patel, D. A., & Patel, R. R. (2019, June). *AI and web-based human-like interactive university chatbot (UNIBOT).* En 2019 3rd international conference on electronics, communication and aerospace technology (ICECA) (pp. 148-150). IEEE.

[45] Mahesh, B. (2020). *Machine learning algorithms-a review*. International Journal of Science and Research (IJSR). [Internet], 9, 381-386.

[46] Dari, R. S. (2020). *Machine learning chatbot for education search purpose in Dublin* (Doctoral dissertation, Dublin Business School).

[47] Galitsky, B., & Galitsky, B. (2019). *Chatbot components and architectures.* Developing Enterprise Chatbots: Learning Linguistic Structures, 13-51.

[48] Gupta, N. (2013). *Artificial neural network*. Network and Complex Systems, 3(1), 24-28.

[49] Webster, J. J., & Kit, C. (1992). *Tokenization as the initial phase in NLP*. In COLING 1992 volume 4 : The 14th international conference on computational linguistics.

[50] Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). *A survey of the usages of deep learning for natural language processing.* IEEE transactions on neural networks and learning systems, 32(2), 604-624.