

Mémoire de projet de Fin d'étude

Developpement d'un chat boot SFR basé sur le Traitement du langage Naturel (NLP)

Réalisé par :Akarkab maryam
Année Universitaire: 2024-2025

Etudiante en année terminale du cycle d'ingénieur



Dédicace

Je dédiée ce travail :

À ma chère maman, source de vie, d'amour, de joie et de soutien inconditionnel.

À mon très cher père, qui a toujours soutenu mes décisions et a fait des sacrifices pour m'encourager à atteindre des sommets.

À mes chères sœurs, Nohaila et Imane, qui m'ont comblé d'amour et de bonheur , Je leur souhaite une vie remplie de réussite et de

bonheur

Je remercie tout particulièrement ma sœur Imane pour son soutien et ses idées qui m'ont aidée à réaliser ce projet

À toute ma famille, pour leur soutien constant et leurs encouragements tout au long de ce parcours

À mes précieuses amies, avec qui j'ai partagé de merveilleux moments.

À mes professeurs, pour leur dévouement et leurs efforts qui ont contribué à me fournir une solide formation

À toutes les personnes qui m'ont encouragé et soutenu tout au long de cette expérience

Merci du fond du cœur.

Maryam Akarkab

Remerciements

Je souhaite également remercier particulièrement M Mohamed snineh mon encadrant pédagogique pour son soutien et ses précieux conseils tout au long de ce projet. Son expertise, sa disponibilité ont été des atouts

Mes remerciements vont à l'ensemble du personnel que j'ai eu le plaisir de travailler avec eux, J'ai été chaleureusement accueillie et j'ai bénéficié de l'aide et de l'assistance nécessaires

Enfin, je tiens à exprimer ma profonde gratitude envers tout le corps enseignant de l'Institut supérieure de génie et des affaires , en particulier les professeurs du département informatiques , pour les compétences qu'ils ont su nous transmettre

Abstract:

With the memory of the final memory in the Big Data & Intelligence Artificial Intelligence, I have the opportunity to develop a superior agent, so that I can communicate with the owners of the MANEO RESEAUX device, prepare the SFR device. This chatbot helps assist the equipment members in the immediate response to questions that arise on the technical requirements, the online treatment process or bring the errors to the table, also in the training rooms, as well as through the training of new collaborators in the past. Provide necessary information to be more active.

For function, the chatbot is alimenté with donné files: an Excel file with projectors and JSON files, as well as a dataset.

This chatbot uses pre-printed models (NLP) to answer users' questions

Résumé

Dans le cadre de ce mémoire de fin d'études en Ingénierie Big Data & Intelligence Artificielle, j'ai eu l'occasion de développer un agent virtuel, ce dernier afin d'améliorer la communication des membres de l'équipe MANEO RESEAUX , précisément l'équipe SFR.

Ce chatbot doit assister les membres de l'équipe en leur répondant instantanément aux questions qu'ils se posent sur les sujets techniques, les process internes de traitement ou encore les erreurs à éviter , les règles d'ingénierie aussi, mais également comme outil de formation des nouveaux collaborateurs en leur fournissant les informations nécessaires pour bien engager leur activité

Pour fonctionner, le chatbot est alimenté par des fichiers de données : un fichier Excel de suivi des projets et deux fichiers JSON, ainsi qu'un dataset

Ce chatbot utilise des modèles préentraînés (NLP) pour bien répondre aux questions des utilisateurs.

Table des matières

Chapitre 1 :Etat de l'art /cadre théorique	9
A-Etat de l'art	9
1-1 définition de chatboot NLP intention entité	9
1-2 Evolution des agents conversationnels (chatbots)	10
1-3 Cas d'usage des chatbots dans la Formation ou l'onboarding	12
1-4 Comparaison avec d'autres outils (slack bot, Microsof Teams bot ,etc)	13
1-5 Cadre théorique :Fonctionnement des chatboot IA /NLP /agents autonomes	13
1-6 Technologie existantes sur les boots	15
1-7 Référence Scietifque technique	16
1-8 Extrait des articles Scientifique	17
Chapitre 2 :Présentation de l'entreprise/Contexte professionnel.....	
1-1 présentations de l'entreprise	20
1-2 Organisation interne	20
1-3 Mission confiées.....	21
1-4 Rôle personnel	21
1-5 Présentation de l'équipe SFR et ses missions	21
1-6 Besoin identifié pour les nouveaux arrivés	21
1-7 Objectif de chatboot dans le contexte	22
1-8 Role personnel dans le projet	22
Chapitre 3 :Analyse du besoin /Cahier des charges	
A-Problématique et enjeux :	23
1-1 Difficultés rencontrés lors de l'intégration des nouveaux membres	23
1-2 Manque de la disponibilités des managers	23
1-3 Problème de centralisation de l'information.....	23
1-4 Inccohérence dans la transmissions des règles	23
1-5 besoins d'autonomie rapide	23
1-6 Manque de la tracabilité de l'apprentissage	23
B-Règles de gestion :	24
1-1 Présentation générales des règels de gestion	24
1-2 Fiabilités de réponses	26
1-3 Accès au donnés sensibles	27
1-4 Administarion de contenu statique	28
1-5 Capitalisation des question non comprises	28
1-6 Autonomie de chatboot	29
1-7 Synthese de règels de gestion	29
C-Solution proposés :	29
1-1 Présentation de l'architecture générale	29
1-2 Choix de technologies justifiés	31
1-3 Source de donnés	35
1-4 Objectif de boot télégramme	35
1-5 Fonctionnalités attendue	35
1-6 Contrainte technique est organisationnelle	35
D-Diagrammes :	36
1-1 Diagramme de flux de donnés (DFD)	36
1-2 Diagramme de scénarios conversationnel	36
1-3 Arbre décisionnel	36

1-4 Pipeline de traitement IA /NLP	36
1-5 Schéma d'archetecture logique	37
E- Role et importance des diagrammes dans la conception d'un chatbot NLP.....	37
E- Représentation des diagrammes	38
Chapitre 4 :Conception et développement	
4-1 :Annalyse de l'archetecture globale	51
A- Description générale de la solution	57
B- Choix de la plateforme télégramme pour le boot	57
C- Choix de NLP ,pour comprendre les questions.....	57
4-2 :Choix des technologies utuisés	57
A- Python est bibliothéques utilisés	57
B- Modèle NLP	59
C- Environnement d'exécusion : spyder.....	59
4-3:Conception fonctionnel du boot	60
A- Gestion des commandes principales(exemple:/start)	60
B-Implémentation des réponses via similarités sémantiques	67
C-utilisation d'une base FAQ enrichie (question réponse)	67
4-4:Développement du système « système de FAQ »	67
A-Encodage des question FAQ en vecteurs (embeddings)	67
B-Calcule de la similarité cosinus pour la recheche de la meilleur réponse.....	67
C-Seuille de confiance et gestion de réponses non comprise	
4-5 :Interface utilisateur et interaction	67
A-Flux de conversation avec l'utulisateur	67
B-Gestion de salutation et des réponses Automatique	67
C-Conclusion	68

Introduction Générale

Dans le cadre de mon projet de fin d'études d'ingénieur spécialisé en Big Data et Intelligence Artificielle, j'ai participé au développement d'un chatbot intelligent visant à améliorer la communication et l'efficacité au sein du bureau d'études MANEO RESEAUX. Ce travail s'inscrit dans une démarche d'innovation technologique visant à optimiser les processus internes et à maximiser l'utilisation des données de l'entreprise

Context :

Dans les équipes techniques, comme dans le bureau d'études de MANEO RESEAUX, la communication d'information importants, la gestion des instructions techniques et l'intégration de nouveaux employés peuvent être difficiles en raison d'un manque d'uniformité ou de centralisation des connaissances. Dans ce contexte, les technologies d'intelligence artificielle, en particulier les chatbots basés sur le traitement du langage naturel (NLP), offrent des possibilités innovantes pour automatiser l'accès à l'information et faciliter la collaboration.

Problématique

La question centrale de ce projet était la nécessité de développer un outil interactif intelligent capable de fournir des réponses instantanées aux questions techniques, de guider les nouveaux membres de l'équipe et de prévenir les erreurs fréquentes dues à une mauvaise compréhension des règles d'ingénierie ou des processus internes. L'objectif était de dépasser les limites de la communication traditionnelle en proposant un assistant numérique accessible et pertinent.

Objectifs du Mémoire

Les principaux objectifs de cette thèse sont les suivants

- Développer un agent virtuel fonctionnel et adapté aux besoins spécifiques de l'équipe
- Intégrer des données stockées dans les fichiers Excel, json
- Utiliser des techniques de NLP pour comprendre et traiter les questions des utilisateurs
- Aider les nouveaux employés pendant leur processus d'intégration.
- L'objectif générale est d'améliorer la productivité et la qualité du travail collaboratif au sein de l'équipe.

Chapitre 1 : État de l'art / Cadre théorique

A-Etat de l'art

1-1 : définition de chatboot ,NLP ,intention ,entité

a-Définition de chatboot :

Un chatbot (abréviation de « chat » et « robot ») est un programme informatique capable de communiquer automatiquement avec un utilisateur en langage naturel (écrit ou parlé).

- Il peut répondre à des questions, aider à la navigation, automatiser le service à la clientèle, etc.

B- Definition NLP

est une technologie de traitement des machines qui permet aux ordinateurs d'interpréter et de générer du langage humain.

🧠 À quoi sert la NLP ?

Le NLP est utilisée pour rendre les machines capables de comprendre le langage humain de la même manière que nous le comprenons, afin de :

- ➕ lire et comprendre des textes
- ➕ comprendre le sens du texte, même s'il ne s'agit pas de la même phrase
- ➕ répondre à des questions

🎯 Role de NLP :

- ➕ interprétation de texte : retirer le sens d'un texte ou d'une phrase
- ➕ Génération de texte : générer du texte automatiquement
- ➕ Diagnostic Grammatical : Reconnaître les mots leurs natures , leurs fonction
- ➕ Reconnaissance de la parole : Identification de la parole ,
- ➕ Reconnaissance vocale :Convertir ce qu'une personne a dit (discours) en mots écrits (texte).

C-Intention

Qu'est-ce qu'une intention ?

Une intention reflète le besoin que l'utilisateur exprime dans sa phrase.

Exemple 1 :

- ➕ L'utilisateur écrit : « Je veux commander un jus de fruit »

- Intention : commander un jus de fruit

Exemple 2 :

💡 L'utilisateur écrit : « À quelle heure ouvrez-vous demain ? »

➡️ Intention : demander les horaires d'ouverture.

🧠 Quel est l'intérêt ?

Dans un chatbot, identifier l'intention permet de comprendre ce que l'utilisateur veut faire, même si les phrases sont différentes.

➡️ Exemple :

💡 « Je souhaite passer une commande de repas»

💡 « Je voudrais commander de la nourriture. »

💡 « Puis-je commander quelque chose à manger ? »

💡 « Je voudrais commander quelques plats. »

D-Entité

Entité est une est un élément d'information spécifique extrait du message d'un utilisateur qui

permet d'atteindre un objectif (aide à complire l'intention)

1-2 : Evolution des agents conversationnels (chatbots)

D'après plusieurs années , il y a une communication entre l'homme et la machine à cause de l'avancement de la technologie, précisément ce qui est en relation avec l'intelligence artificielle, qui est une partie des agents – conversationnels, plus précisément appelés chatbots ,parmi lesquels le commercial, l'éducation et l'administration Ces solutions logicielles ont été créées pour comprendre la communication humaine (texte ou voix), c'est-à-dire reconnaître ce que l'utilisateur veut, et répondre d'une façon correcte et utile, qui correspond exactement à la question posée.

Le développement des chatbots repose directement sur les progrès effectués dans la pertinence du traitement de langage naturel (NLP). Plus, le traitement du langage naturel (NLP) est efficace, plus le chatbot dispose d'une intelligence ajustée. C'est-à-dire, le traitement de langage naturel (NLP) est le savoir-faire donné à la machine de (comprendre le sens d'une question ou d'un texte écrit/ou énoncé dans l'oral, traiter et interpréter la demande de l'utilisateur de manière appropriée et délivrer une réponse en se servant de langage naturel comme le ferait un être

humain) :

Les simples programmes en règles de correspondances datant des années 1960, les chatbots sont devenus capables de dialogues de la complexité des échanges entre humains s'appuyant sur des modèles d'apprentissage automatique par réseaux neuronaux profonds.

Historique de l'évolution des agents conversationnels (chatbots)

A. Les débuts avec ELIZA (1966)

le premier chatbot historique, ELIZA, a été développé en 1966 par Joseph Weizenbaum au MIT. Ce système utilisait des consignes simples pour examiner les entrées de l'utilisateur et répondre en simulant un psychothérapeute. Même si la contrainte de reconnaissance de mots-clés et à la correspondance de modèles, ELIZA a démarré des interactions homme-machine en langage naturel.

B. Approfondissement dans les années 1970-1980

Durant cette époque , des chatbots comme PARRY ont été développés pour reproduire des réactions plus spécifiques, comme celui d'un patient souffrant de paranoïa. Les systèmes étaient encore basés sur des instructions strictes, ce qui freinait leur flexibilité et leur capacité de compréhension

C. L'avènement d'ALICE (1995)

En 1995, ALICE a introduit l'utilisation de l'AIML (Artificial Intelligence Markup Language), permettant la production de bases de connaissances plus étendues et modulaires. Bien qu'ALICE reste un moteur basé sur des règles, il offre une interaction plus riche que ses prédecesseurs.

D. Chatbots et IA émergente dans les années 2000

Le développement d'internet a permis l'utilisation de chatbots pour aider le service client en ligne sur les sites web. En même temps, l'amélioration du traitement de la parole et le début de vrais algorithmes d'apprentissage automatique ont aidé à créer des chatbots plus logiques capables d'apprendre de nouvelles formulations et à s'optimiser.

E. L'ère des assistants vocaux (2010 et au-delà)

Les années 2010 ont vu la montée en puissance des assistants vocaux intelligents, à l'exemple de Siri, Alexa et Google Assistant. Ces assistants virtuels exploitent des techniques avancées de NLP et de deep learning pour percevoir la situation, interpréter la voix humaine

6. Les modèles de langage à grande échelle (depuis 2020)

Aujourd'hui, les agents conversationnels basés sur des modèles linguistiques pré-entraînés, tels que GPT-3 et ChatGPT, ont transformé profondément le domaine. Ces modèles utilisent de nombreux paramètres et des réseaux neuronaux multicouches pour produire des réponses plus complexes, contextuelles et hautement naturelles

1-3 : Cas d'usage des chatbots dans la formation d'onboarding

les chatbots joueraient un rôle essentiel dans le domaine de la formation et l'onboarding, en particulier pour les entreprises. Ils fourniraient un mentorat interactif et personnalisé aux nouveaux employés et aux stagiaires, adapté à toute heure. Par exemple, lorsqu'un nouvel employé se fait abonner, alors les chatbots seraient capables de lui parler de toutes les procédures internes, les politiques de l'entreprise, les outils utilisés, etc. Ils répondraient également immédiatement aux questions liminaires posées. Cela accélérerait le processus d'intégration et allègerait la tâche du Manager-est. De plus, pour la formation continue, des agents conversationnels décents serviraient d'outils d'apprentissage pratiques pour rappeler des concepts importants, sans se exercer via plusieurs exercices pratiques, et évaluer le niveau de réflexion.

a. Introduction aux chatbots dans le contexte de la formation et de l'onboarding

L'émergence de l'intelligence artificielle fait des chatbots un outil puissant pour améliorer l'expérience utilisateur dans de nombreux domaines. En matière de formation et de recrutement, ces assistants virtuels jouent un rôle clé en automatisant certaines tâches de formation, en facilitant l'onboarding des nouveaux employés et en rendant le processus de formation plus interactif et personnalisé.

B. Chatbots comme outils de formation continue

De plus, les chatbots peuvent dispenser des cours de façon progressive et personnalisée pour chaque individu. Les interactions naturelles des chatbots permettent une autoformation assurée à l'aide des réponses dissertation aux questions des apprenants, quiz, rappel des cours aux apprenants, ce qui motive l'utilisateur tout au long de son apprentissage.

C. Automatisation du processus d'onboarding

Les nouveaux employés peuvent bénéficier d'utilisation de chatbot pour l'intégration. On peut les informer dès le début sur le règlement, les procédures RH, et l'organisation en général. De plus, dans le cadre de leurs missions initiales et des questions fréquemment posées évidemment, le chatbot les guidera, de sorte que les RH soient moins souvent surchargés de demandes. Ce qui affecte évidemment positivement le processus d'intégration, qui devient beaucoup plus fluide, rapide et les conditions sont également standardisées.

D. Suivi et évaluation personnalisée

De plus, les chatbots peuvent être utilisés pour surveiller les progrès des utilisateurs en formation ou en intégration. Avec les informations collectées sur les échanges, il est possible pour eux de produire des évaluations, de recommander des contenus à chaque niveau et de détecter les défaillances à corriger. Il peut ainsi y avoir un meilleur lien entre l'apprentissage et l'ajustement au poste.

E. Accessibilité et disponibilité permanente

L'un des principaux avantages des chatbots est leur disponibilité 24h / 24 et 7j / 7. car un chatbot ne dort jamais. Par conséquent, à la différence d'un formateur ou d'un RH, un chatbot est toujours accessible. De plus, il existe une expérience inégale entre formation à distance et fin en raison des fuseaux horaires et de la question du travail à domicile pour fournir un environnement de travail équitable. Par conséquent, l'accès direct responsabilise les apprenants et réduit les coûts de l'organisation.

1-4 : Comparaison avec d'autre outils (slack bot ,Microsoft teams bot)

Dans le cadre de la numérisation de la formation ,C'est a dire Utiliser le numérique pour former et intégrer efficacement les nouveaux membres. les chatbots deviennent une solution innovante pour automatiser le contact avec les personnelles de l'entreprise Actuellement, il est possible de développer des agents conversationnels et de les exécuter à l'aide de différentes technologies, chacune ayant ses propres avantages et inconvénients. Cette Annalyse propose une comparaison de trois strategies : des bots intégrés à Slack, un agent pour Microsoft Teams et une chatbot personnalisée exécutée à l'aide de l' API Telegram, Python et le langage naturel Toolkit. Les bases de données ont été utilisées pour stocker les fichiers et des informations utiles pour l'utilisateur

a-Présentation des technologies

Les bots Slack sont créés pour s'adapter à l'environnement de collaboration Slack largement utilisé par les équipes techniques. Ils sont qualifiés de répondre aux messages, de proposer des boutons interactifs et de déclencher des automatisations. Cependant, leur déploiement nécessite une configuration technique (API Slack, gestion des permissions, événements), ce qui peut rendre le lancement plus complexe

Les bots Microsoft Teams, en revanche, sont intégrés à l'écosystème Microsoft 365. Ils permettent une interaction transparente avec les utilisateurs d'Outlook, de SharePoint et de Teams, mais nécessitent l'utilisation d'Azure Bot Framework et parfois de Power Virtual Agents. Bien que puissants, ils sont souvent limités par la complexité de l'environnement Azure et la nécessité d'un compte d'entreprise

Enfin, le chatbot développé via l'API de Telegram, combiné à Python et NLTK, est une solution plus flexible et plus accessible. Il permet une interaction directe avec les utilisateurs via une interface largement adoptée et le traitement du langage peut être personnalisé à l'aide de bibliothèques open-source. L'utilisation d'une base de données ou de fichiers Excel pour stocker les réponses et les historiques rend ce type de bot parfaitement adapté aux projets universitaires ou aux systèmes personnalisés

1-5 : Cadre théorique :Fonctionnement des chats boot IA /NLP /agents autonomes

le développement rapide de l'intelligence artificielle, les chatbots sont devenus des outils essentiels dans plusieurs domaines par exemple le service client, l'éducation, la santé, le commerce en ligne, ainsi que dans beaucoup d'autres secteurs ,Leur fonctionnement est basé sur différents domaines de l'IA, comme le traitement automatique du langage naturel (NLP), les modèles d'apprentissage automatique, et, plus dernièrement les agents autonomes capables de prendre des décisions en temps réel.

Les chatbots qui utilisent l'intelligence artificielle sont basés sur des algorithmes capables de comprendre et d'interpréter le langage humain afin de fournir des réponses adaptées juste est utiles ,ces chatbots apprennent en

analysant beaucoup de textes , Cela leur permet de répondre à beaucoup de type de question différents et d'adapter leur langage selon le contexte et Cela veut dire que certains chatbots peuvent penser un peu comme un humain, c'est-à-dire comprendre et analyser les questions pour y répondre intelligemment, plutôt que de suivre uniquement des instructions fixes.

Des modèles récents, tels que le Generative Pre-trained Transformer (GPT), exploitent des réseaux neuronaux profonds pour générer des réponses cohérentes et contextualisées, par contre les boots classiques, suivent juste les instructions fix .

Le NLP joue un rôle important dans les chatbots. C'est une filière de l'intelligence artificielle ,dont le but est d'enseigner à une machine la compréhension des requêtes d'utilisateurs et la manipulation du langage humain. Le NLP contient plusieurs étapes importantes : l'analyse des mots (tokenisation), syntaxe (parsing), reconnaissance d'entités nommées (NER), et compréhension du sens d'un énoncé dans sa totalité (analyse sémantique). Tout cela donne la possibilité aux chatbots de comprendre l'intention des utilisateurs, ce qui est très important pour fournir la réponse exacte.

a-Les types de chatbots : scriptés vs intelligents

Cette section présente les différences entre les chatbots à règles fixes (basés sur des scripts prédéfinis) et les chatbots intelligents (alimentés par l'apprentissage automatique et le NLP). Leurs avantages, leurs inconvénients et leurs cas d'utilisation sont expliqués ici

b-Architecture générale d'un chatbot intelligent

Voici les composants techniques d'un chatbot à intelligence artificielle :

- Interface utulistaeur (messagerie , web , mobile)
- Moteur NLP (Tokenizer ,parseur ,classificateur d'intentions
- Moteur de dialogue (gestion des etats de conversation)
- Base de connaissances ou intégration a une API externe
- Générateur de réponses

c-Le rôle du NLP dans la compréhension du langage humain

Cette section détaille le fonctionnement de la NLP:

- Prétraitemment de text
- Comprehension d'intention
- Reconnaissances d'entités
- Annalyse de sentimments

d-Apprentissage automatique dans les chatbots avec intelligence artificielle

Cette section décrit comment les chatbots intelligents utilisent l'apprentissage automatique et l'apprentissage profond pour s'améliorer. Inclure ici :

- Entrainement par renforcement
- Entraînement supervisé (ex : classer les intentions)

E-Agents autonomes et prise de décision

un agent autonome, ce n'est pas juste un robot qui balance des réponses toutes faites. Non, c'est un système qui comprends ce qui se passe autour de lui et fait ses trucs tout seul , ça veut dire qu'ils sont capables de capter le contexte , d'enchaîner plusieurs messages sans perdre le fil, et de prendre des décisions sans que l'humain derrière ait à intervenir toutes les deux minutes , Ils ont souvent une sorte de mémoire pour se rappeler des conversations des règles pour éviter de partir en freestyle, et parfois même des plans d'action pour préparer leurs prochains Réactions. Certains sont encore plus futés et s'améliorent au fil de la discussion grâce à des systèmes de récompense.

1-6 : Technologie existantes sur les boots (Microsoft Power Virtual Agents, Dialogflow, Rasa ou encore Talla.

a- Introduction aux technologies de création de chatbots

Le développement des interfaces conversationnelles a conduit à l'apparition de multiples plateformes technologiques destinées à faciliter le développement, le déploiement et la gestion des chatbots. Ces solutions offrent généralement des fonctionnalités de traitement du langage naturel (NLP), des interfaces de conception visuelle et des capacités d'intégration avec des systèmes tiers. Des acteurs tels que Microsoft Power Virtual Agents, Google Dialogflow, Rasa (une plateforme open source) et Talla (spécialisé dans les environnements professionnels) se sont imposés. Chaque technologie présente des caractéristiques distinctes, répondant à des exigences variées en termes de personnalisation, de sécurité et de connectivité

b. Microsoft Power Virtual Agents

Microsoft Power Virtual Agents est une solution SaaS qui offre aux utilisateurs, même qui n'ont pas des compétences en programmation de créer des chatbots conversationnels pour les entreprises. La plateforme collabore de manière transparente aux services Microsoft (Teams, Dynamics 365, Azure) et offre une interface intuitive de type « glisser-déposer ». Elle se base sur le cloud de Microsoft, ce qui assure une sécurité et une capacité d'adaptation robustes.

Grâce à son intégration avec Power Automate

Elle peut être utilisée pour automatiser les processus métier, permettant aux bots d'interagir avec une série d'applications tierces

c . Dialogflow (Google)

Dialogflow est une solution puissante de Google pour créer des chatbots fondée sur le Google Cloud NLP. Il prend en charge plusieurs langues et offre beaucoup des fonctionnalités avancées à titre d'exemple la gestion du contexte, la détection d'intention et la reconnaissance d'entités. Parmi ses intégrations les plus courantes, on peut citer Google Assistant, des applications web, des systèmes vocaux tels que des IVR et n'importe quel logiciel de messagerie tel que Messenger ou Slack. La variante Dialogflow CX est une solution pour les tâches complexes qui impliquent des dialogues étape par étape et des flux structurés

4. Rasa

Rasa est également une plate-forme open source à partir de laquelle les développeurs et les entreprises cherchent obtenir un contrôle total et direct sur l'infrastructure de leur chatbot, et non une tierce personne. Il s'agit d'une machine personnalisée en comparaison avec Dialogflow et d'autres solutions cloud basé, et il convient mieux aux organisations qui prennent au sérieux la confidentialité et la sécurité des données. Rasa comporte plusieurs. Rasa est un moyen d'entreprise avec des besoins de projets très graves

5. Talla

Talla est un chatbot créé pour les environnements professionnels, plus précisément pour le support technique, les RH et la documentation interne. Il possède de capacités d'apprentissage automatique et de gestion des connaissances pour automatiser les réponses aux questions fréquemment posées. Talla se reconnaît par sa capacité à apprendre en continu à partir des documents et interactions passés. Il est conçu pour développer la productivité des équipes en fournissant des réponses rapides, cohérentes et contextualisées

1-7 : Référence scientifique technique

a- Travaux de recherche sur les chatbots et l'IA conversationnelle

De nombreuses publications de recherche ont été consacrées au développement et à l'utilisation d'agents conversationnels. Les travaux de Jurafsky et Martin (2023) et Mauldin (1994) ont posé les bases du traitement du langage naturel dans les systèmes de dialogue. Plus récemment, les modèles de transformation du langage (Vaswani et al., 2017) ont permis un saut significatif dans la qualité des interactions homme-machine. Cette étude montre que la performance des chatbots dépend largement de la qualité des données d'entraînement et de la gestion des intentions et des entités

B . Publications techniques sur les frameworks NLP et agents autonomes

des documents techniques publiés par des experts de l'industrie et du monde universitaire décrivent en détail le fonctionnement interne de cadres tels que Rasa, Dialogflow et OpenAI GPT. Par exemple, la documentation technique

de Rasa explique la division entre NLU (compréhension du langage) et Core (gestion du dialogue). D'autres articles, comme ceux publiés dans arXiv, se concentrent sur des cas d'utilisation spécifiques tels que la gestion du service client ou l'éducation numérique avec des chatbots

C- Normes et standards technologiques

La conception des chatbots repose également sur des normes techniques reconnues. Par exemple, le W3C (World Wide Web Consortium) recommande des normes pour les interfaces vocales et textuelles. La norme ISO/IEC 30170 définit les bonnes pratiques pour la programmation des systèmes intelligents et les normes de cybersécurité (ISO/IEC 27001) sont nécessaires pour protéger les données échangées entre les bots

D- Études de cas et rapports industriels

Plusieurs entreprises et institutions publient des rapports d'analyse sur l'impact des chatbots dans les industries professionnelles. Par exemple, le rapport Gartner (2023) sur les technologies émergentes souligne le rôle stratégique des assistants virtuels dans l'expérience client. D'autres études menées par McKinsey ou IBM Research montrent comment les bots autonomes réduisent les coûts opérationnels tout en améliorant l'expérience utilisateur

E- Synthèse et lien avec notre problématique

Toutes ces références scientifiques et techniques montrent que les chatbots ne sont pas de simples outils logiciels, mais des systèmes intelligents basés sur une tradition de recherche rigoureuse et d'innovation continue. Ce contexte théorique et technique servira de base solide au développement de notre propre chatbot intelligent dans le cadre de cette thèse

1-8 : Extrait des articles scientifique

1. Vaswani et al. (2017) – “Attention Is All You Need”

Extrait d'article n°1 :

We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely... Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task..."

(Vaswani et al., 2017)

Lien d'article :<https://arxiv.org/abs/1706.03762>

➤ Explication :

Cet article fondateur présente le modèle Transformer, une architecture qui a changé le domaine du NLP.

Contrairement aux réseaux traditionnels basés sur la récursivité (RNN), Transformer repose uniquement sur des mécanismes d'attention, ce qui permet un parallélisme énorme et de meilleures performances dans des tâches telles que la traduction automatique. Ce modèle est à la base de GPT, BERT et d'autres architectures modernes

2. Jurafsky & Martin (2023) – *Speech and Language Processing*

Extrait d'article n°2 :

The idea of giving computers the ability to process human language is as old as the idea of computers themselves. This book is about the implementation and implications of that exciting idea. We introduce a vibrant interdisciplinary field with many names... The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech

Lien <https://web.stanford.edu/~jurafsky/slp3/>

➤ Résumé d'article :

Ce manuel de traitement du langage naturel décrit les bases des systèmes de dialogue, en expliquant la structure des chatbots modernes compréhension du langage (NLU), gestion du dialogue (Dialogue Manager) et génération de texte (NLG). Il s'agit d'un manuel de base utilisé dans les universités.

3. Mauldin (1994) – *ChatterBots, TinyMUDs, and the Turing Test*

Extrait d'article n°3:

Early chatterbots, like ELIZA, relied on keyword matching and pattern rules to simulate human dialogue. The conversational abilities were originally implemented as simple IF-THEN-ELSE rules based on pattern matching with variable assignment.”

(Mauldin, 1994)

Lien <https://cdn.aaai.org/AAAI/1994/AAAI94-003.pdf?>

➤ Résumé d'article :

Mauldin (1994) analyse les limites des premiers agents conversationnels tels qu'ELIZA, qui étaient basés sur des règles symboliques rigides. Il souligne la difficulté de modéliser des conversations humaines authentiques sans flexibilité cognitive et suggère d'explorer des modèles plus dynamiques, ouvrant ainsi la voie aux progrès de l'IA de dialogue.

Chapitre 2 : Présentation de l'entreprise / Contexte professionnel

1.1 Présentation de l'entreprise Manéo Réseaux

A-Historique et activités principales

Manéo Réseaux est une entreprise ciblée dans les solutions énergétiques et de télécommunications, plus précisément dans le secteurs des réseaux électriques, des infrastructures de recharge pour véhicules électriques et des projets de télécommunications avec des partenaires tels que SFR

B-Valeurs et vision

L'entreprise place l'excellence et la réactivité et la transformation au cœur de sa stratégie et cherche à automatiser ses mécanismes internes pour optimiser son efficacité

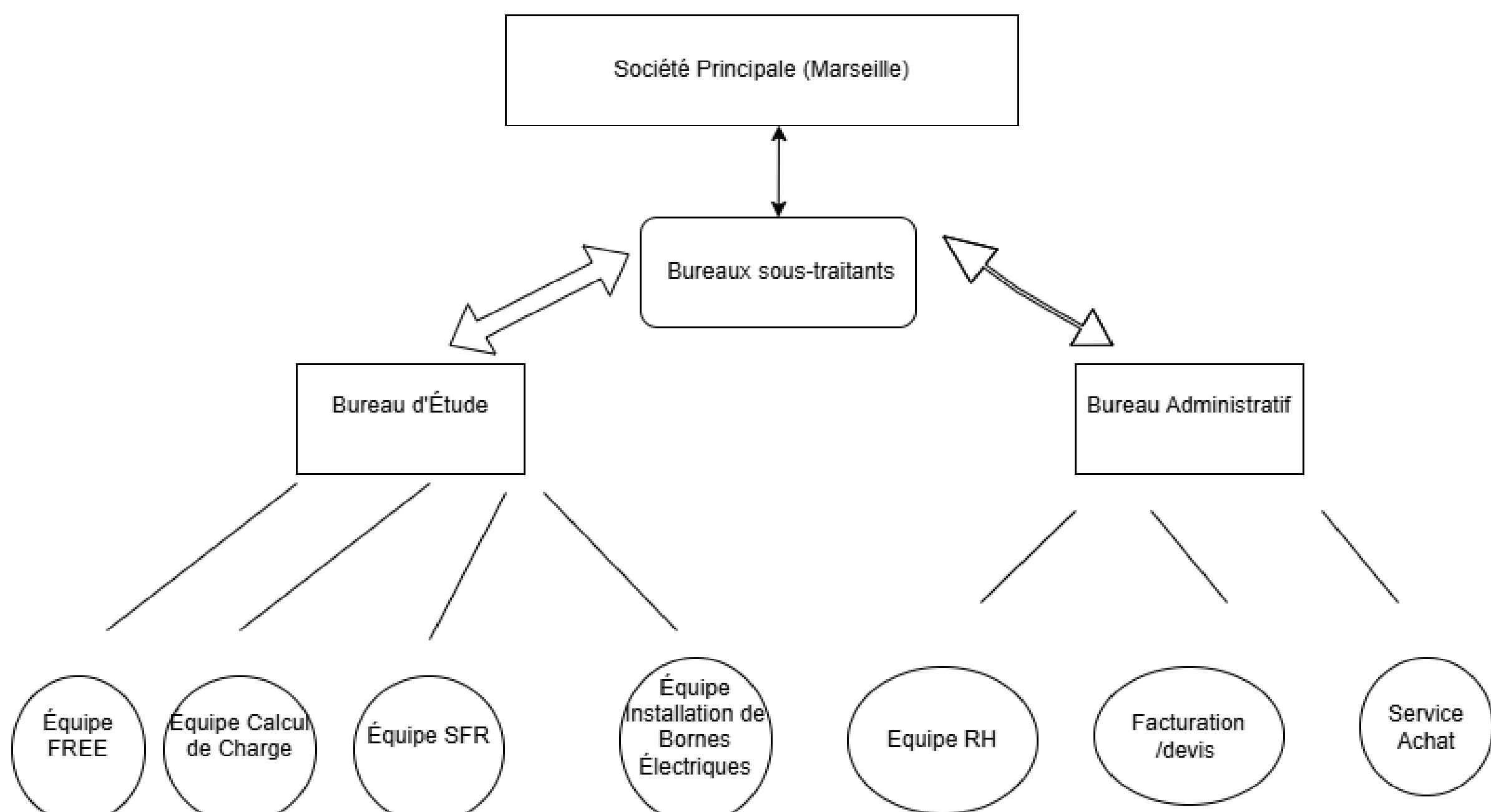
1-2: Organisation interne

A-Les deux pôles principaux

- Bureau administratif : gestion des devis, facturation, contrats, gestion du personnel
- Bureau d'études : études techniques, calculs de charge, installation de terminaux électriques et traitement des projets de télécommunications.

B- Organigramme simplifié

La structure interne de la société se base sur une structure hiérarchique simplifiée qui assure la libre diffusion de l'information et un partage claire des responsabilités. Elle est dirigée par l'équipe du PDG, Les



différents services sont placés sous la direction de responsables destinés , chaque personne est chargé de contrôler le travail de son équipe et de coordonner les actions avec les autres services Cette organisation assure une gestion efficace des projets

1-3: Missions confiées

A-Suivi de projets télécoms (SFR)

Gestion des données du projet, analyse des délais, coordination des techniciens, suivi des problèmes

b-Automatisation et centralisation des informations

Développer et déployer des outils numériques pour réduire la saisie répétitive de données et améliorer la communication a l'intérieur

c-Développement d'un chatbot

Création et mise en place d'un chatbot pour intégrer le nouveau personnel et unifié la documentation du projet

1-4: Rôle personnel

A-Telecom Project Analyst & Automation Specialist:

Mon rôle est d'assurer le suivi analytique des projets SFR, d'automatiser les rapports et outils internes, et de proposer des solutions nouvelles comme le chatbot basé sur le NLP

B-Compétences mobilisées:

Analyse de données, automatisation (scripts,), développement de chatbot, compréhension des flux de télécommunication traitement du langage naturel

1-5: Présentation de l'équipe SFR et ses missions

A- Composition de l'équipe SFR

L'équipe est constituée d'ingénieurs en télécommunications, de chefs de projet, de spécialistes techniques et de coordinateurs

B-Responsabilités principales

■ Réception des demandes SFR

■ Études de faisabilité

■ Coordination des projets

■ Suivi des plannings et comptes rendus de travaux

1-6: Besoin identifié pour les nouveaux arrivés

A-Problèmes rencontrés lors de l'intégration

Les nouvelles personnes ajoutées à l'équipe manquent d'un encadrement structuré , manque de leadership centralisé, etc.

b- Nécessité d'un outil d'assistance intelligent

Il était donc nécessaire de proposer un assistant virtuelle pour aider les nouveaux employés à développer leurs compétences et à devenir plus indépendants

1-7 Objectif du chatbot dans le contexte

A-Objectif fonctionnel

Centraliser les informations sur les projets, répondre aux questions fréquemment posées, guider les nouveaux employés étape par étape sans besoin d'attendre les managers ou les chefs équipes

B-Objectif stratégique

réduire les erreurs humaines, simplifié le travail des chef l'équipe et des managers en leurs réduisant leurs charge, facilité le transfert et le partage de l'information

1-8 : Rôle personnel dans le projet

A-Détection du besoin et proposition de solution

On a basé sur notre observations et les commentaires de l'équipe, pour ça j'ai proposé de développer un chatbot conversationnel qui centralise l'information est disponible tout le temps

B-Conception, développement et déploiement

On a développé un flux conversationnel, formé un modèle NLP, connecté avec des sources de données internes

Chapitre 3 : Analyse du besoin / Cahier des charges

A-Problématique et enjeux:

1-1 – Difficultés rencontrées lors de l'intégration des nouveaux membres

Lorsque une nouvelle personne est ajoutée à l'intérieur de l'équipe SFR est un mécanisme long , Ces employés évoluent dans un environnement professionnel chargé avec des outils, des procédures et une culture d'entreprise qu'ils doivent comprendre rapidement. En l'absence d'un encadrement organisé , cette période d'adaptation est perturbante ce qui peut causer un retard dans le développement des compétences.

1-2 Manque de disponibilité des managers

Les managers et les chefs équipes bien qu'ils soient obligatoires à l'intégration des nouveaux employés , possèdent de peu de temps pour commencer à cette mission. Leur masse de travail limite leur capacité à offrir une aide personnalisée et continue aux nouveaux personnes ajoutées à l'équipe . Cette limite de capacité influe directement la qualité , et le temps nécessaires pour que les employés deviennent indépendants est plus long

1-3: Problème de centralisation de l'information

Les données essentielles à l'intégration (règles d'ingénierie, documentation de projet, contacts principaux, processus internes de traitements) sont souvent réparties sur plusieurs supports,

Cette séparation complique l'obtention rapide d'une réponse exacte

L'absence d'une base de données unique peut causer un gaspillage de temps , et toujours les nouveaux personnes ajoutées à l'équipe doivent attendre les managers ou bien les chefs d'équipes pour qu'ils puissent obtenir l'information

1-4– Incohérences dans la transmission des règles

On n'a pas un système central qui pour transfer les informations , les instructions internes sont parfois communiquées verbalement ou via des documents qui ne sont plus à jour. Cela peut entraîner que chaque personne peut comprendre différemment les instructions

. Ces différences affectent la méthode de travail dans le groupe et peuvent causer des fautes lors de la production

1-5 Besoin d'autonomie rapide des nouveaux collaborateurs

Dans un milieu professionnel motivant il est important que les nouveaux employés soient indépendants , on attend de l'équipe SFR qu'elle soit capable prendre les décisions de comprendre les principes de gestion et d'utiliser les outils internes sans avoir besoin des managers , ou bien des chefs d'équipes Il est donc essentiel d'avoir une solution qui donne la possibilité d'apprentissage autonome et d'accès indépendant à l'information.

1-6 Manque de traçabilité de l'apprentissage

On n'a pas un système pour suivre les questions mentionnées par les nouveaux salariés, donc on ne peut pas améliorer le processus d'intégration ni détecter les erreurs qui reviennent. Pour y remédier, il faut mettre en place un système qui collecte les erreurs afin d'améliorer l'apprentissage et les futures intégrations , et ça facilite aussi le processus d'intégration

B-Règles de gestion

1.1 Présentation générale des règles de gestion

Dans la création d'un agent virtuel intelligent, les règles de contrôle sont essentiel pour garantir une conduite structurée et adaptée aux besoins de l'utilisateur. Elles définissent la méthode dont le chatbot est supposé de répondre en fonction des intentions identifiées.

Le but des règles de fonctionnement est d'assurer :

- Prise de décision logique par le chatbot,
- interaction homme-machine fluide,
- respect des contraintes métiers de SFR (authentification, filtrage des requêtes, traitement hiérarchique des requêtes),
- des réponses cohérentes même dans des situations complexes et variées

1- RG01 – Fiabilité des réponses

- Le chatbot est créé pour offrir des réponses exactes et logiques, il se base uniquement sur des informations confirmées présentes dans la base de données
{Fichier excel /les deux fichiers json}

Justification de l'utilisation de données validées:

La condition principale pour garantir la confiance des utilisateurs et éviter la diffusion d'informations incorrectes est la fiabilité des réponses. Grâce à une base de données Excel et deux fichiers json, validée par l'équipe techniques de SFR, le chatbot assure que les informations fournies respectent les procédures internes.

Description du mécanisme d'extraction depuis Excel

Les données sont collectées à partir d'Excel en utilisant des bibliothèques Python comme pandas ou openpyxl. Pour chaque question identifiée, le chatbot analyse les colonnes convenables en utilisant des mots-clés ou des expressions standardisées, puis fournit la réponse adaptée

RG02 – Accès sécurisé aux données sensibles

Seules les personnes authentifiées avec leur nom peuvent accéder à l'information des projets SFR (date de dépôt / date de réception / Etat de projets /Responsable de projets /regions)

Gestion des identifiants Telegram

Le nom de l'utilisateur Telegram est enregistré lors de la première interaction. Il est vérifié pour chaque requête nécessitant un niveau d'accès supérieur, une vérification est effectuée. Si le nom n'est pas valide, une réponse par défaut est donnée

- Historique des tickets associés à un numéro de projet

► Accès accordé :

Quand un membre de l'équipe demande des infos sur un projet sur notre application, le bot vérifie d'abord qu'il a le droit. Ensuite, il va chercher les infos dans le fichier envoie la réponse directement à la personne.

RG03 – Administration du contenu statique

Le contenu statique (procédures de traitement, définitions, règles d'ingénierie) peut être mis à jour sans modifier le code

Définition de contenu statique vs dynamique :

■ **Statique** : contenu rarement modifié (procédures internes, explications du code)

■ **Dynamique** : Une réponse créée juste quand on en a besoin

(a titre d'exemple , réponses NLP provenant d'un modèle d'apprentissage)

L'interface ou la méthode d'administration prévue :

Les données sont récupérées à partir d'Excel , suivie des projets {en utilisant des bibliothèques Python comme pandas ou openpyxl} , et les deux fichiers json.

Pour chaque question identifiée, le chatbot examine les colonnes pertinentes en utilisant des mots-clés ou des expressions standardisées, puis fournit la réponse appropriée.

Exemple de mise à jour des règles

■ Ajouter des nouveaux projets dans le suivie d'avancement sfr

■ Correction de la procédure , ou d'un ensemble des règles d'ingénierie en modifiant les fichiers json

RG04 – Capitalisation des questions non comprises

Le chatbot doit enregistrer toutes les questions auxquelles il n'a pas pu répondre dans un fichier spécial. Cela permet de voir ce qui ne va pas et d'améliorer le système, surtout pour gérer les questions qui ne sont pas dans sa base de données.

Si le message ne correspond à aucun élément enregistré dans la base de données Excel, il est catégorisé comme « hors base de données »

Processus d'amélioration continue

Des analyses hebdomadaires peuvent être programmées par les équipes techniques pour :

■ regrouper les questions similaires,

■ identifier de nouveaux besoins utilisateurs,

■ mettre à jour la base de connaissances

RG05 – Autonomie du chatbot

Le chatbot est conçu pour fonctionner de manière autonome, sans aucune intervention humaine nécessaire.

- Il détecte les intentions via NLP
- Il récupère lui-même les données nécessaires
- Il gère les erreurs et les malentendus (par exemple, les messages alternatifs, les redirections)

Il rédige des carnets de bord pour améliorer ses performances sans supervision constante

L'autonomie est un objectif clé pour :

- réduire la charge de travail humaine, réduire la charge de travail humaine
- répondre 24h/24 aux utilisateurs
- garantissent une réponse rapide, même après les heures de bureau

1-2:Fiabilité de réponse

La **fiabilité des réponses** est un facteur fondamental pour assurer la qualité du travail d'un chatbot. Elle repose sur la pertinence des informations fournies, la vérification des sources et la capacité du chatbot à éviter les erreurs et les malentendus. Cette section détaille les éléments qui garantissent la fiabilité dans le cadre d'un chatbot développé pour SFR.

A-Importance de la fiabilité dans un chatbot métier

Dans un cadre professionnel tel que SFR, un chatbot ne doit pas avoir le droit de fournir des renseignements imprécis. Chaque réponse a un impact direct sur la qualité du service rendu aux techniciens, au personnel ou aux clients. Une réponse inexacte peut conduire à un retard d'exécution,. Il est donc important que les réponses soient fiables.

B-Utilisation d'une base de données validée

Le chatbot se fonde uniquement sur des données ([un fichier Excel suivie de projet + 2 fichiers json](#)) ces derniers sont approuvés par l'équipe techniques SFR . Ces référentiel renferme des procédures de traitements , les règles d'ingénierie , des informations sur les projets

c-Mécanisme d'appariement des questions

Le système de traitement du langage naturel (NLP) intégré au chatbot évalue chaque message entré afin d'identifier l'intention de l'utilisateur. Après avoir identifié l'intention, une liaison est formée avec l'enregistrement approprié dans la base de données Excel. On utilise des méthodes de traitement automatique du langage, comme l'identification de mots-clés ou la similarité sémantique, pour assurer un appariement exact entre la question posée et la réponse fournie

D-Prévention des erreurs de réponse

Pour éviter toute réponse erronée :

➤ Le système de traitement du langage naturel (NLP) du chatbot examine chaque message

entré afin d'identifier l'intention de l'utilisateur.

Après avoir identifié l'intention, une liaison est formée avec l'enregistrement approprié dans la base de données

Excel. On utilise des méthodes de traitement automatique du langage, comme l'identification de mots-clés ou

la similarité sémantique, pour assurer un appariement exacte entre la question posée et la réponse fournie.

➤ Il enregistre les questions non comprises pour analyse ultérieure

E-Mises à jour régulières du contenu

Il est possible de mettre à jour périodiquement la base de données utilisée afin d'incorporer de nouvelles informations ou d'apporter des modifications aux procédures en place.

Cette aptitude à s'adapter confère au chatbot la capacité de fournir des réponses constantes au fil du temps, même en cas de modifications des processus internes.

1-3 Accès au données sensibles

Pour le développement du chatbot SFR, les données sensibles font référence à toutes les informations privées

Ou secrètes que l'utilisateur pourrait partager avec le système.

Étant donné que le chatbot est conçu pour dialoguer avec les clients de SFR sur Telegram

Risques liés à la sécurité des données

L'implémentation d'un chatbot intégré à Telegram pose des enjeux spécifiques en termes de sécurité.

➤ Telegram est une application sécurisée, mais les données passent par ses serveurs, vous devez donc comprendre comment elle fonctionne en termes de stockage et de chiffrement

➤ Sans un filtrage approprié, le traitement NLP peut interpréter ou stocker des données sensibles par inadvertance

➤ En l'absence de contrôles, un pirate peut interagir avec le chatbot pour tenter d'en extraire des informations sensibles

Mécanismes de sécurisation mis en place

Pour garantir la confidentialité, l'intégrité et la sécurité des données, un certain nombre de mesures techniques ont été incluses dans le développement du chatbot :

 Authentification des requêtes Telegram : le bot n'est accessible que via un jeton API sécurisé fourni par [BotFather Telegram](#)

 Échanges cryptés : la communication entre Telegram et Python-bot s'effectue via le protocole HTTPS par l'intermédiaire de bibliothèques sécurisées (par exemple python-telegram-bot).

 Filtrage des entrées NLP : les messages des utilisateurs sont analysés à l'aide de règles prédéfinies afin d'identifier les expressions sensibles et d'empêcher leur traitement automatique.

 Pas de stockage local de données sensibles : le chatbot traite les données dans une mémoire non volatile et ne les stocke pas, sauf en cas de besoin fonctionnel spécifique, en les chiffrant si nécessaire

1-4 Administration de contenu statique

Gestion et intégration de contenus non dynamiques dans le chatbot SFR

Dans le cadre du développement d'un chatbot intelligent pour SFR, il est nécessaire de structurer une base de contenus statiques (informations fixes, questions fréquentes, procédures de traitements , règles d'ingénierie etc.) Ces contenus, stockés dans les {fichiers JSON/fichier exel }, représentent des réponses standardisées que le chatbot peut fournir sans faire appel à des services externes. Cette méthode assure la cohérence, la fiabilité et la mise à jour des réponses, tout en facilitant leur gestion.

L'objectif est que les non-développeurs puissent facilement ajouter, modifier ou supprimer ce contenu en modifiant simplement {les fichier JSON/ suivie exel des projets sfr }Cette méthode apporte une certaine adaptation dans la gestion des documents du chatbot

1-5 Capitalisation des questions non comprises

Un élément essentiel dans l'amélioration continu d'un chatbot NLP est la valorisation des cas de non-compréhension.

Chaque fois qu'un chatbot n'arrive pas à comprendre une question, il enregistre dans une base de données réservée .

Cette base de données offre la possibilité de déterminer les intentions cachées

Cette étude permet à l'équipe de projet d'améliorer le modèle NLP, en intégrant de nouvelles intentions

Utilisateur pose une question

↓

Bot ne trouve pas de réponse/intention

↓

→ Répond : "Je ne comprends pas encore cette question."

→ Enregistre la question dans un fichier JSON

1-6 Autonomie du chatbot

L'indépendance d'un chatbot se mesure à sa capacité à comprendre, gérer et répondre à des demandes. Cette indépendance se base sur l'efficacité du moteur NLP.

Pour le projet SFR, le chatbot doit être en mesure de fournir des réponses automatiques aux questions qui sont beaucoup posées, tout en orientant les utilisateurs vers un intervenant humain lorsqu'il le faut.

1-7 Synthèse des règles de gestion

Les instructions de gestion qui ont été mises en place dans ce projet ont pour but de garantir la performance, la fiabilité et l'évolution du chatbot SFR. Cela englobe la qualité des contenus fixes, la gestion sécurisée des informations, l'apprentissage par l'amélioration de l'indépendance du système. Cette révision vise à formaliser les meilleures méthodes qui seront appliquées afin de garantir une expérience utilisateur optimale et une gestion de projet efficace sur la durée.

C-Solution proposés

1-1 Présentation de l'architecture générale

Le système utilise un chatbot intelligent sur Telegram. L'utilisateur pose ses questions, et un programme en Python analyse ces questions grâce à l'intelligence artificielle pour fournir des réponses adaptées. Tout est connecté à une base de données qui garde les règles et informations importantes. Ce système est flexible, facile à faire évoluer et simple à gérer.

1-2 Choix de technologies justifiés

■ **Telegram** a été sélectionné comme interface en raison de sa simplicité d'intégration grâce aux API et de sa capacité à fonctionner sur plusieurs plateformes.

■ **Le langage Python** : Ce projet utilise le langage Python parce qu'il est facile à comprendre et à utiliser. Il dispose de beaucoup de bibliothèques utiles, notamment pour le traitement du langage, l'intelligence artificielle et l'analyse de données. Python est aussi rapide à mettre en place, ce qui le rend parfait pour tester des idées.

■ **Pour le traitement du langage naturel**, le choix s'est porté sur les bibliothèques spaCy et Transformers en raison de leur complémentarité et de leurs performances reconnues dans le domaine du NLP.

spaCy : On a choisi la bibliothèque spaCy pour sa vitesse, sa fiabilité et son aptitude à réaliser efficacement des tâches fondamentales telles que la tokenisation, l'analyse syntaxique ou l'extraction d'entités nommées. Elle est spécialement conçue pour répondre aux exigences d'un traitement linguistique léger ou en temps réel, tout en étant simple à incorporer dans une structure Python.

Les tableaux ci-après présentent de façon détaillée toutes les technologies utilisées.

Catégorie	Nom	Description / Rôle
Langage	Python 3	Language principale de projet ,pour automatiser le traitement des données , le NLP et l'intégration de modèle
Bibliothèques standard	asyncio	Gestion des tâches asynchrones pour permettre au bot de traiter plusieurs requêtes en parallèle.
Bibliothèques standard	json	Lecture et écriture de fichiers JSON (FAQ, procédures, historique de questions).
Bibliothèques standard	os	Manipulation de fichiers et répertoires (ex. vérifier l'existence d'un fichier).
Bibliothèques standard	datetime	Création de timestamps pour journaliser les interactions.
Bibliothèques standard	re	Expressions régulières pour détecter les codes projets (ex. P24403)
Bibliothèques standard	collections.Counter	Compte la fréquence des questions afin de trouver les plus posées
Bot Telegram	python-telegram-bot	Framework pour créer un bot Telegram, gérer les commandes et les conversations

Bibliothèques Utilisées dans le Projet

Technologie	Description / Rôle
spaCy	Librairie NLP pour charger un modèle de détection de salutations (./mon_modele_salutation).
transformers	Librairie Hugging Face pour utiliser et entraîner des modèles BERT/DistilBERT pour la classification de texte.
datasets	Chargement et gestion de jeux de données pour l'entraînement
torch	Framework PyTorch pour la manipulation de tenseurs et l'entraînement de modèles NLP
sentence-transformers	Création d'embeddings sémantiques pour comparer les similarités entre questions (recherche FAQ)

Technologies et leur rôle dans le projet NLP

Nom	Description / Rôle
mon_modele_salutation (spaCy)	Modèle entraîné localement pour reconnaître les salutations (bonjour, salut...).
DistilBERT (distilbert-base-uncased)	Modèle Transformers pré-entraîné et affiné pour classer les intentions (procédure, FAQ, projet)
all-MiniLM-L6-v2	Modèle Sentence Transformers utilisé pour la recherche sémantique et la comparaison de similarité entre questions FAQ.
intent_classifier_model	Modèle DistilBERT finement entraîné sur la dataset (intent_dataset.csv) pour reconnaître les intentions des questions utilisateurs.

Modèle NLP utilisé

Justification du choix du stockage des données (Excel, JSON sur Google Drive)

Le système se base sur des fichiers organisés en tant que sources d'information, notamment :

- Un document **Excel** dédié au suivi de l'avancement des projets SFR,
- Deux fichiers au format **JSON**, contenant les procédures et la FAQ.

Ces fichiers sont stockés sur Google Drive, facilitant leur centralisation, leur actualisation et leur consultation à distance. L'application intègre automatiquement ces fichiers via la bibliothèque Python gdown, ce qui facilite leur récupération sans besoin d'interaction manuelle.

1-3 Source de données :

Ce projet s'appuie sur des documents internes utilisés pour la gestion des projets SFR. Il utilise notamment un fichier Excel qui sert au suivi des projets, ainsi que deux fichiers JSON contenant les questions fréquemment posées et les modes opératoires.

Ces ressources permettent d'alimenter le chatbot avec des exemples concrets et pertinents, facilitant ainsi la constitution d'une base de données réaliste pour entraîner les modèles de traitement du langage naturel. L'objectif est que le chatbot fournisse des réponses plus précises et adaptées aux besoins des utilisateurs

Ce passage est extrait du fichier de suivi des projets, l'une des principales sources utilisées par le chatbot. Les informations y sont structurées afin que le système de traitement du langage naturel (NLP) puisse accéder efficacement aux éléments essentiels tels que le secteur, le numéro UO, le type de dossier, le statut, ainsi que les dates clés.

SECTEUR	UO	NB.Lien	Date de réception	Odeon	type de dossier	Date de depot	ETAT
NY SOCIETY	P29694	1	6/1/2025	15878761 001 392	Depot	6/1/2025	Déposé Axis
NY SOCIETY	P29695	1	6/1/2025	15878682 001 962	JR	6/1/2025	Déposé Axis
NY SOCIETY	P29696	2	6/1/2025	15878814 001 316	JR	6/1/2025	Déposé Axis
NY SOCIETY	P29697	1	6/1/2025	15878794 001 596	Depot	6/1/2025	Déposé Axis
NY SOCIETY	P29698	1	7/1/2025	15880799 001 904	JR	7/1/2025	Déposé Axis
EST	P29699	1	7/1/2025	15882467 001 741	Depot	7/1/2025	Déposé Axis
EST	P29700	1	7/1/2025	15882459 001 924	Depot	7/1/2025	Déposé Axis
NY SOCIETY	P29701	1	7/1/2025	15882978 001 556	Depot	7/1/2025	Déposé Axis
NY SOCIETY	P29702	1	7/1/2025	15880755 001 535	JR	7/1/2025	Déposé Axis
NY SOCIETY	P29703	1	7/1/2025	15880778 001 938	JR	7/1/2025	Déposé Axis
EST	P29704	1	7/1/2025	15883631 001 897	JR	7/1/2025	Déposé Axis
CENTRE EST	P29705	1	7/1/2025			création ultaedge	7/1/2025
CENTRE EST	P29706	1	7/1/2025			création ultaedge	7/1/2025
CENTRE EST	P29707	1	7/1/2025			création ultaedge	7/1/2025
CENTRE EST	P29708	1	7/1/2025			création ultaedge	7/1/2025
CENTRE EST	P29709	1	7/1/2025			création ultaedge	7/1/2025
CENTRE EST	P29710	1	7/1/2025			DEPOSE ULTRA EDGE	7/1/2025
CENTRE EST	P29711	1	7/1/2025			création ultaedge	7/1/2025
CENTRE EST	P29712	2	7/1/2025			JR	7/1/2025
CENTRE EST	P29713	2	7/1/2025			JR	7/1/2025
CENTRE EST	P29714	2	7/1/2025			JR	7/1/2025
IDF	P29715	1	8/1/2025	15887445 001 108	Depot	8/1/2025	Déposé Axis
IDF	P29716	1	8/1/2025	15879884 001 836	JR	9/1/2025	Déposé Axis
EST	P29717	1	9/1/2025	15884272 001 428	JR	9/1/2025	Déposé Axis

Extrait de la base de données « suivie des projets SFR »

```
{
  "procedure_SFR": {
    "description": "Processus pour réaliser un projet SFR étape par étape.",
    "etapes": [
      {
        "numero": 1,
        "titre": "Vérification du SVI",
        "description": "Vérifier le SVI de proche et s'assurer qu'il est clôturé ou non. Demander à Delphe qui s'occupe de la partie Odeon / SVI."
      },
      {
        "numero": 2,
        "titre": "Téléchargement de l'attribution Axiss",
        "description": "Télécharger l'attribution Axiss de SFR. Demander les identifiants Axiss SFR soit à Maxim Trazzera soit à Maryam Akarkab."
      },
      {
        "numero": 3,
        "titre": "Téléchargement du retour terrain",
        "description": "Télécharger le retour terrain envoyé par le technicien dans l'adresse mail CA. Demander les identifiants de cette adresse mail soit à Maxim soit à Maryam."
      },
      {
        "numero": 4,
        "titre": "Copie du retour terrain",
        "description": "Copier le retour terrain dans la feuille d'attribution Axiss."
      },
      {
        "numero": 5,
        "titre": "Recherche du numéro de projet",
        "description": "Dans la feuille téléchargée (retour terrain et attribution Axiss), chercher le numéro de "
      }
    ]
  }
}
```

Extrait du premier fichier JSON {procedure -SFR.json}

Cette partie décrit comment est organisé le fichier JSON qui contient toutes les informations utiles pour le chatbot. Dans ce fichier, chaque question est associée à sa réponse. Cette structure permet au système de comprendre le langage naturel et de trouver rapidement la bonne réponse quand un utilisateur pose une question.

Le format JSON est simple à utiliser : il permet de modifier ou d'ajouter facilement de nouvelles questions et réponses, ce qui garantit que le chatbot reste fluide et peut s'adapter facilement.

Le fichier est accessible ici : https://drive.google.com/file/d/1tQHxKRmejdD7SciBTrA5V_jBA4n_2Sdl/view.

```
{
  "faq": [
    {
      "question": "Quel est le nom de l'équipe ?",
      "réponse": "Équipe SFR"
    },
    {
      "question": "Quelle est l'activité de l'équipe SFR ?",
      "réponse": "La réalisation des livrables PV et REC"
    },
    {
      "question": "Quels sont les projets sur lesquels travaille l'équipe ?",
      "réponse": "Projets SFR et projets UE"
    },
    {
      "question": "Qui sont les membres de l'équipe SFR ?",
      "réponse": "Maryam Akarkab, Abderhmane Darouiche, Fatimazahra Bouzakraoui, Adelphe"
    },
    {
      "question": "Quelle est la description générale des tâches ?",
      "réponse": "La première étape pour traiter un projet est de créer un ODEON. Ensuite, le projet peut être traité."
    },
    {
      "question": "Qu'est-ce qu'un ODEON ?",
      "réponse": "C'est un ordre de travail ou une autorisation d'accès aux datacenters. Il doit être créé en premier avant de pouvoir traiter un projet."
    },
    {
      "question": "Quels sont les objectifs du bot ?",
      "réponse": "Objectif du bot :\n- Comprendre le rôle de chaque membre\n- Connaître les règles d'ingénierie\n- Être autonome"
    }
  ]
}
```

Extrait du deuxième fichier JSON {FAQ-sfr complet}

Ce fichier JSON rassemble une liste de questions et de réponses (FAQ) destinée à l'équipe SFR. Il contient :

- Les questions les plus fréquentes concernant l'équipe, ses activités et ses projets
- Les réponses précises et organisées qui y sont associées
- Des informations complémentaires comme (Le nom des membres de l'équipe /Les coordonnées des techniciens/La description des missions et travaux réalisés par l'équipe)

Le format JSON a été choisi car il est léger, bien structuré et facile à utiliser par une application (comme un chatbot). Cela permet de répondre automatiquement aux questions courantes sans intervention manuelle

Le fichier est accessible ici : <https://drive.google.com/file/d/1XtZesLx35tN-YOBiOyKfYgmlAlbNkQz5/view?usp=sharing>

Ce tableau présente en détail les sources de données de ce projet

Catégorie	Nom	Description / Rôle
Module interne	faq_sfr_complet	Contient un classificateur NLP local et des fonctions pour chercher les réponses dans la FAQ
Fichiers de données	faq_sfr_complet.json, procedure_sfr.json, faq_augmented.json	Bases de connaissances pour répondre aux questions fréquentes et procédurales
Fichier Excel	avancement_memoir_modif.xlsx	Données sur l'état d'avancement des projets (UO, dates, état, etc.)

données du projet

1- 4 Objectif du bot Telegram

L'objectif principal du chatbot sur Telegram est d'offrir une assistance rapide, simple et disponible 24 heures sur 24 aux membres de l'équipe SFR. Il permet de faciliter l'accès aux informations essentielles sur les projets, de répondre aux questions courantes et de guider les utilisateurs dans l'application

Le chatbot vise à réduire le besoin d'intervention humaine, à améliorer le suivi des demandes et à aider les nouveaux membres de l'équipe grâce à un outil interactif toujours accessible

1-5 Fonctionnalités attendues

Parmi les fonctionnalités attendues souhaitées :

- la capacité à comprendre les questions en langage courant,
- capacité détecter l'intention, (c'est à dire comprendre l'intention de quoi il s'agit) et les éléments clés dans les demandes,
- l'accès à des réponses contextuelles

1-6 Contraintes techniques et organisationnelles

A- Contraintes techniques

Le développement de ce chatbot pour SFR, qui utilise l'intelligence artificielle pour comprendre le langage, pose plusieurs défis techniques importants. D'abord, la compréhension du langage naturel, surtout en français, est difficile à cause des nuances de la langue, des fautes de frappe ou des différentes façons de poser une question. Cela peut rendre complexe la détection précise de l'intention de l'utilisateur. Il est donc important d'utiliser des modèles de traitement du langage performants et de les entraîner avec des données adaptées au contexte métier de SFR.

De plus, l'intégration avec Telegram apporte aussi ses limites, notamment à cause des restrictions de son interface de programmation (API), comme la gestion des conversations, la sécurité des échanges ou encore la taille maximale des messages. Pour faire fonctionner le chatbot correctement, il faut aussi qu'il soit hébergé sur un serveur dans le cloud.

B- Contraintes organisationnelles

Pour établir ce chatbot , il y'a une nécessité que tous l'équipe SFR travaillant ensemble , pour se mettre d'accord sur le plan organisationnelles , l'équipe technique (composée de développeurs) , l'équipe métier (incluant les gestionnaires de règles)

Un nombre aussi élevé d'intervenants nécessite une coordination efficace et un processus de validation des scénarios de conversation bien défini.

Finalement , les dates limites de développement doivent être placées sur la même direction avec les phases du projet , pour respecter le calendrier

D-Diagramme

1-1 : Diagramme de flux de données (DFD)

Un diagramme de flux de données donne une représentation visuelle d'un flux d'information entre les divers éléments d'un système

Il montre comment les données se déplacent à travers les procédures internes ,

1-2 : Diagramme de scénarios conversationnel

Ce diagramme montre les différents chemins de conversation possibles entre un utilisateur et un chatbot. Il modélise l'interaction sous la forme de scénarios (par exemple, salutation, demande d'informations, malentendu, etc.) afin d'anticiper le comportement et la réaction du robot dans différents cas d'utilisation

1-3: Arbre de décisionnel

Un arbre de décision est un modèle logique utilisé pour représenter les choix possibles lors d'une interaction avec l'IA. Chaque nœud correspond à une question ou à une condition, et chaque branche conduit à une action ou à une nouvelle décision. Il s'agit d'un outil important pour structurer la logique conversationnelle ou le traitement automatique des requêtes

1-4: Pipeline de traitement IA / NLP

Ce pipeline explique les étapes enchaînées du traitement du message d'un utilisateur par un système

d'intelligence artificielle. Il comprend typiquement :

la tokenisation (analyse syntaxique des phrases),

l'analyse syntaxique

la compréhension du langage naturel (NLU)

la génération de réponses (NLG),

et l'exécution d'actions

1-5 : Schéma d'architecture logique

Un diagramme d'architecture logique est un diagramme qui représente les différents composants d'un système logiciel et les relations entre eux de manière claire et organisée. Il montre comment les modules (tels que l'interface utilisateur, le moteur NLP, la base de données, etc.) fonctionnent ensemble sans entrer dans les détails techniques (adresses IP, serveurs, etc.).

Dans un projet de chatbot, ce diagramme montre comment l'ensemble du système fonctionne : du message de l'utilisateur à la réponse générée par le chatbot.

E-Rôle et Importance des Diagrammes dans la Conception d'un Chatbot NLP

A- L'importance du diagramme de flux de données (DFD)

Dans la situation de notre projet, un diagramme de flux de données (DFD) est nécessaire pour voir la manière dont l'information se déplace à travers les différents éléments d'un chatbot. Il définit les points d'accès (messages des utilisateurs), les processus internes (analyse NLP, gestion des intentions, génération de réponses) et les sorties (réponses textuelles). Ce diagramme explique le mouvements entre les entités internes et externes telles que les bases de données. Il permet ainsi de organiser la structure fonctionnelle du chatbot, d'améliorer l'organisation des processus et de garantir un traitement fluide et cohérent des données utilisateurs.

B-L'importance d'un diagramme de scénario conversationnel

Ce diagramme est au cœur de la conception de l'interaction entre le chatbot et l'utilisateur. Il est utilisé pour représenter les descriptions typiques qu'un utilisateur peut rencontrer, à titre d'exemple demande d'information, ou bien une question qui se pose beaucoup ou un message d'erreur. Dans notre projet, il participe à apporter les parcours de conversation afin que le chatbot réponde de manière naturelle et opérationnelles dans différents cas d'utilisation. Ce diagramme est également important pour l'entraînement du moteur NLP, car il permet d'identifier les intentions sous-jacentes et de construire le dialogue de manière réaliste et pertinente, en fonction des besoins de l'équipe SFR.

C- Utilisation d'un arbre de décision

Dans ce projet, l'arbre de décision a été employé pour modéliser la logique de décision de notre chatbot en fonction des intentions détectées dans les messages des utilisateurs. Cet moyen nous a permis d'organiser exactement et hiérarchiquement les distinctes réponses possibles à une intention donnée, en considérant que des cas particuliers et des conditions spécifiques.

D-Utilisation du pipeline de traitement IA / NLP

Le pipeline de traitement AI / NLP était un composante clé de l'organisation technique du traitement des messages des utilisateurs. Dans ce projet, ce pipeline nous a permis de préciser chaque phase par laquelle passe un message, depuis sa réception jusqu'à la génération de la réponse convenable. Cela nous a permis d'organiser les modules de tokenisation, de classification d'intention, d'extraction d'entité et de génération de texte. Cette structure modulaire simplifie l'intégration d'outils NLP convenable et l'amélioration des performances du chatbot. Le pipeline nous a également permis de garantir la cohérence entre les étapes de traitement linguistique, assurant une réponse rapide, précise et contextualisée.

E- L'importance d'un diagramme d'architecture logique dans un projet

Dans le contexte de ce projet, le diagramme d'architecture logique a affecté un rôle essentiel dans la conception et l'organisation de l'ensemble du système de chatbot SFR. Ce diagramme donne une orientation claire des différents composants logiciels du projet (interface utilisateur, moteur NLP, base de données, module de réponse, etc.). Il donne une vue générale de l'architecture du chatbot, ce qui donne aux développeurs et aux chefs de projet de comprendre plus facilement le mécanisme du système.

En raison à ce diagramme, il a été possible d'assurer la modularité du développement, une meilleure organisation des flux de données et une intégration cohérente entre les différentes couches de l'application. Il permet également d'anticiper les changements futurs du système, tels que l'ajout de nouveaux services ou de modules NLP. Etant ce diagramme est donc essentiel pour assurer la flexibilité, la maintenabilité et la confiance du chatbot dans un environnement réel tel que SFR.

F-Représentation des Diagrammes

A-. Diagramme de Flux de Données (DFD) — niveau 0

Objet :

Décrire la circulation d'informations entre les entités (utilisateur, fichiers, chatbot) et les processus internes

Exemple DFD Niveau 0 :

Entités externes :

- Utilisateur du télégramme
- Fichier Excel (suivi du projet)
- Fichier JSON (questions et réponses de la FAQ)

Processus :

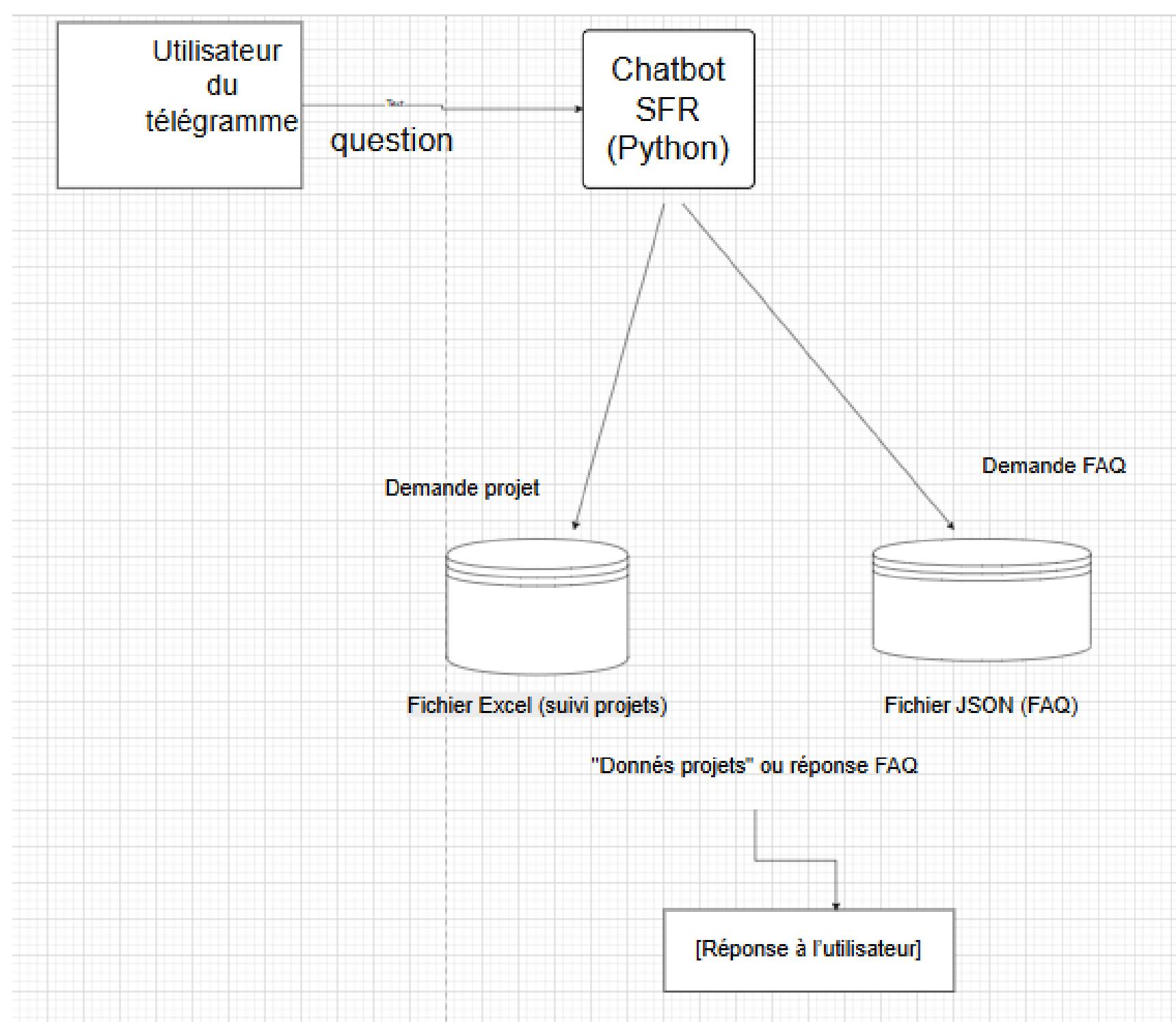
- Traitement d'une demande d'utilisateur (NLP)
- Recherche d'une réponse (via JSON ou Excel)
- Envoi de la réponse via Telegram

Flux :

- Un utilisateur envoie une question via Telegram.
- Le chatbot transmet la question au module NLP.
- Le module NLP sélectionne la source : JSON ou Excel.
- Le système renvoie la réponse à l'utilisateur

DFD Niveau 1 :

- Analyse syntaxique de la requête
- Classification des intentions
- Recherche JSON
- Recherche dans un fichier Excel par ID de projet
- Formatage de la réponse
- Envoi via Telegram



B-. Diagramme de Flux de Données (DFD) — niveau 1

Le DFD de niveau 1 précise les sous-processus internes du système « SFR Chatbot (Python) , en divisant les échanges générales présentées au niveau 0. Il clarifie plus précisément la circulation d'information entre l'utilisateur, les composants internes du chatbot et les fichiers de données utilisés

Éléments de base du DFD niveau 1

1. Objet externe : utilisateur

L'utilisateur communique avec le système via l'application Telegram. Il envoie une demande textuelle au chatbot, la question étant une question sur le processus de traitement ou pour un projet spécifique

2. Processus : analyse de la demande

Cet étape permet à l'utilisateur de recevoir un message. Il s'agit de nettoyer, de tokeniser et de préparer le texte afin qu'il puisse être analysé par les composants NLP du système.

➤ Par exemple : suppression des caractères inutiles, mise en minuscules, segmentation des phrases, etc

3. Processus : Détermination des intentions

Après l'analyse de la requête de l'utilisateur, ce élément applique des techniques de classification NLP (modèle ou règles pré-entraînées) pour préciser les intentions de l'utilisateur.

On identifie deux grands types d'intentions :

➤ une requête liée à un projet (par exemple, "Faites-moi savoir où en est le projet U123456")

➤ une question/FAQ générale (par exemple, "Comment puis-je suivre le projet ?")

4. Processus : Interrogation d'Excel

Si l'intention trouvée est liée à un projet, le chatbot recherche le fichier Excel comportant les données du projet. Il select les lignes par ID de projet et récupère les données pertinentes

5. Processus : Requête JSON

Si l'intention correspond à une question concernant le processus de traitement ou bien les informations des techniciens (leurs numéro de téléphone , leurs adresses mail) , le système recherche dans les fichiers JSON qui contient une base de données de questions et réponses prédéfinies (FAQ). Il recherche la réponse la plus adaptée

6. Processus : Formatage de la réponse

Peux importe le type de réponse reçue ([Excel ou JSON](#)), cette procédure prépare le contenu final :

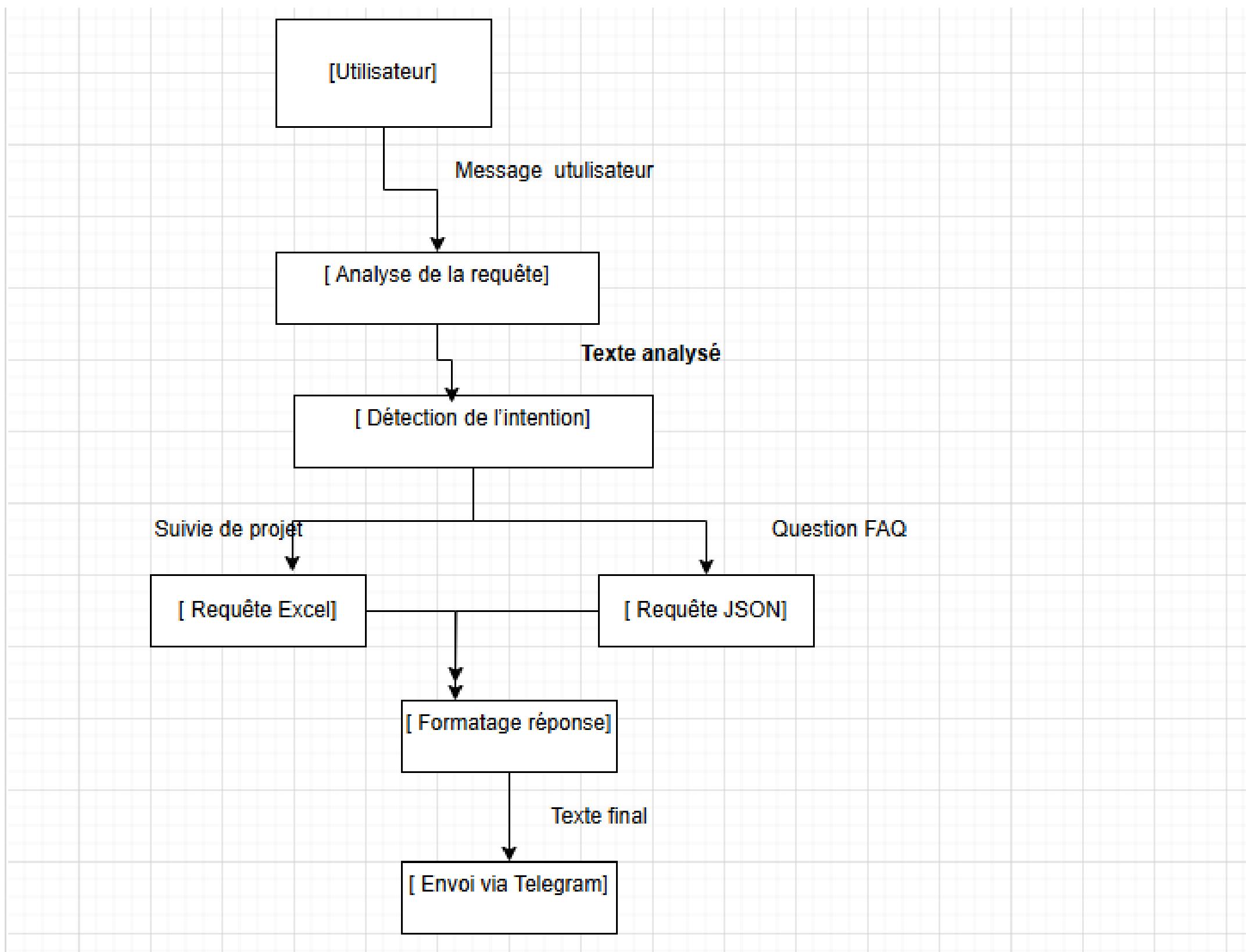
- Ajout d'emoji, de sauts de ligne, de la syntaxe Telegram, etc.
- Structure cohérente et lisible pour l'utilisateur

7. Processus Envoi via Telegram

Le message préparé est ensuite envoyé à l'utilisateur dans Telegram via l'API Telegram Bot. cette opération permet de confirmer que la réponse est partagé sans erreur

Conclusion :

Le DFD de [niveau 1](#) permet de modéliser en détaille la structure interne d'un chatbot. Il montre comment les modules NLP dirigent le flux d'informations entre [les sources de données \(Excel, JSON\)](#) et les réponses adaptées . Ce diagramme simplifie la compréhension des fonctions du système par tous les participants au méthode (chef de projet, développeur, superviseur, etc.).



C-Diagramme de scénarios conversationnel

1- Source de donnée : (suivi de projet SFR)

Au moment où un utilisateur transmet une question sur la situation d'un projet, le chatbot SFR développé en Python entre en action en tant que processus central. Il examine le contenu du message et précise les besoins en information. Si la demande nécessite l'accès à des données dynamiques telles que l'avancement du projet, la date de livraison ou l'identité du responsable, le chatbot accède automatiquement au fichier Excel utilisé pour le suivi des projets SFR. Ce fichier est une source de données structurées contenant toutes les dernières informations sur les projets. Le chatbot effectue une recherche dirigée à partir de mots-clés extraits du message de l'utilisateur, récupère les données exactes, puis formule une réponse adaptée qu'il envoie via l'interface Telegram. Ce dialogue forme une circulation de données bidirectionnel entre la démarche (chatbot) et la base de données Excel, assurant une réponse exacte en temps réel.

Le diagramme ci-dessous montre la circulation de données entre l'utilisateur et un chatbot lors d'une échange typique. Il explique comment un chatbot SFR développé en Python traite une requête envoyée via l'application Telegram.

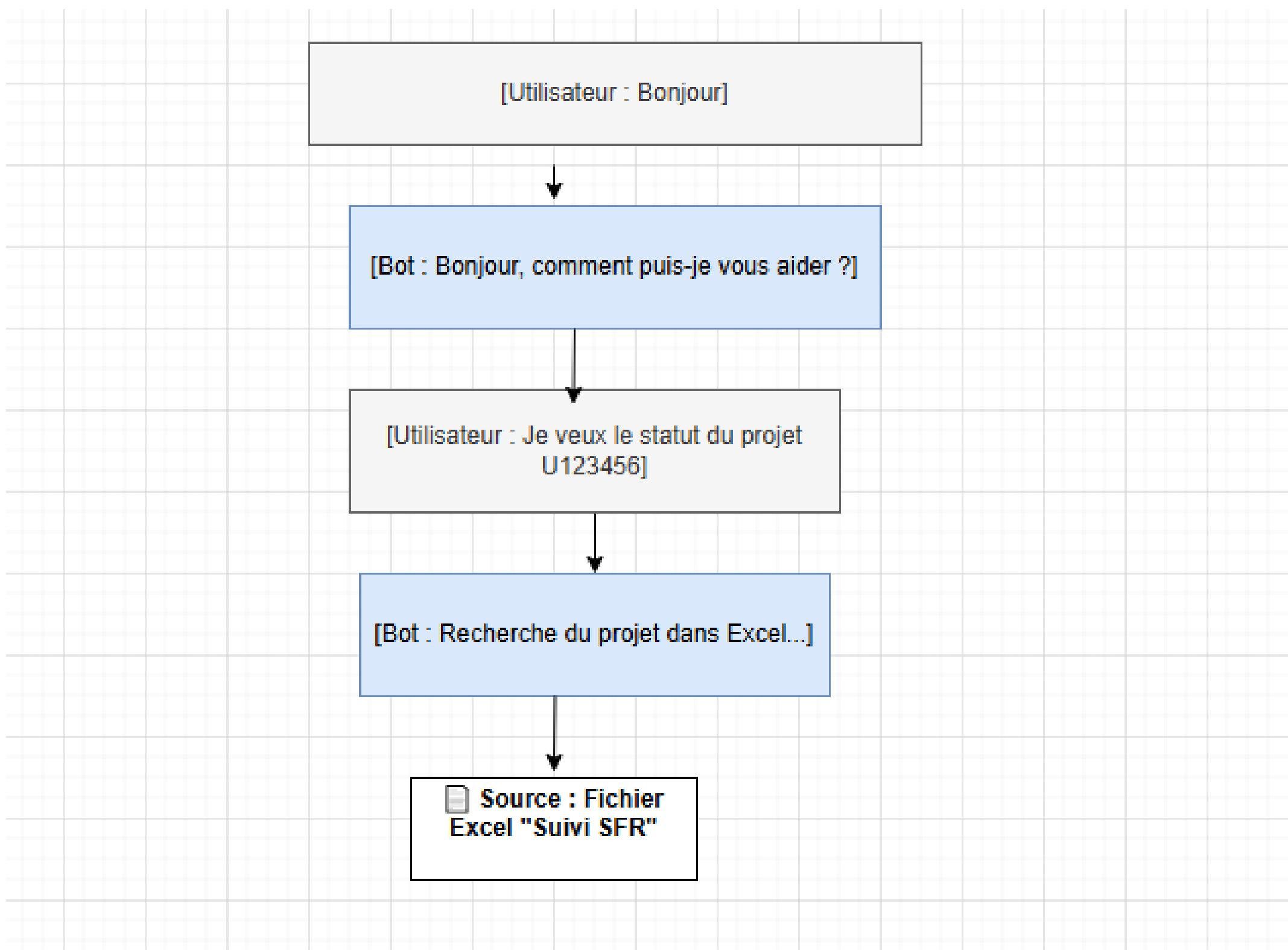
Cet organigramme souligne le rôle principale l du chatbot, qui fonctionne comme un intermédiaire intelligent entre l'utilisateur et les sources d'information.

Plus précisément, cet organigramme décrit un scénario où la question d'un utilisateur nécessite de référencer des données, qu'il s'agisse d'un :

Un fichier Excel de suivi des projets SFR, dans le cas de questions techniques ou opérationnelles ;

ou d'un fichier JSON contenant une FAQ, dans le cas de questions fréquemment posées ou d'informations générales.

Ce schéma montre les différents composants (utilisateur, chatbot, bases de données) et les flux d'information entre eux, depuis la réception d'une question jusqu'à l'envoi d'une réponse personnalisée. C'est un élément visuel important pour comprendre l'architecture logique de la solution mise en œuvre dans ce projet de chatbot intelligent.



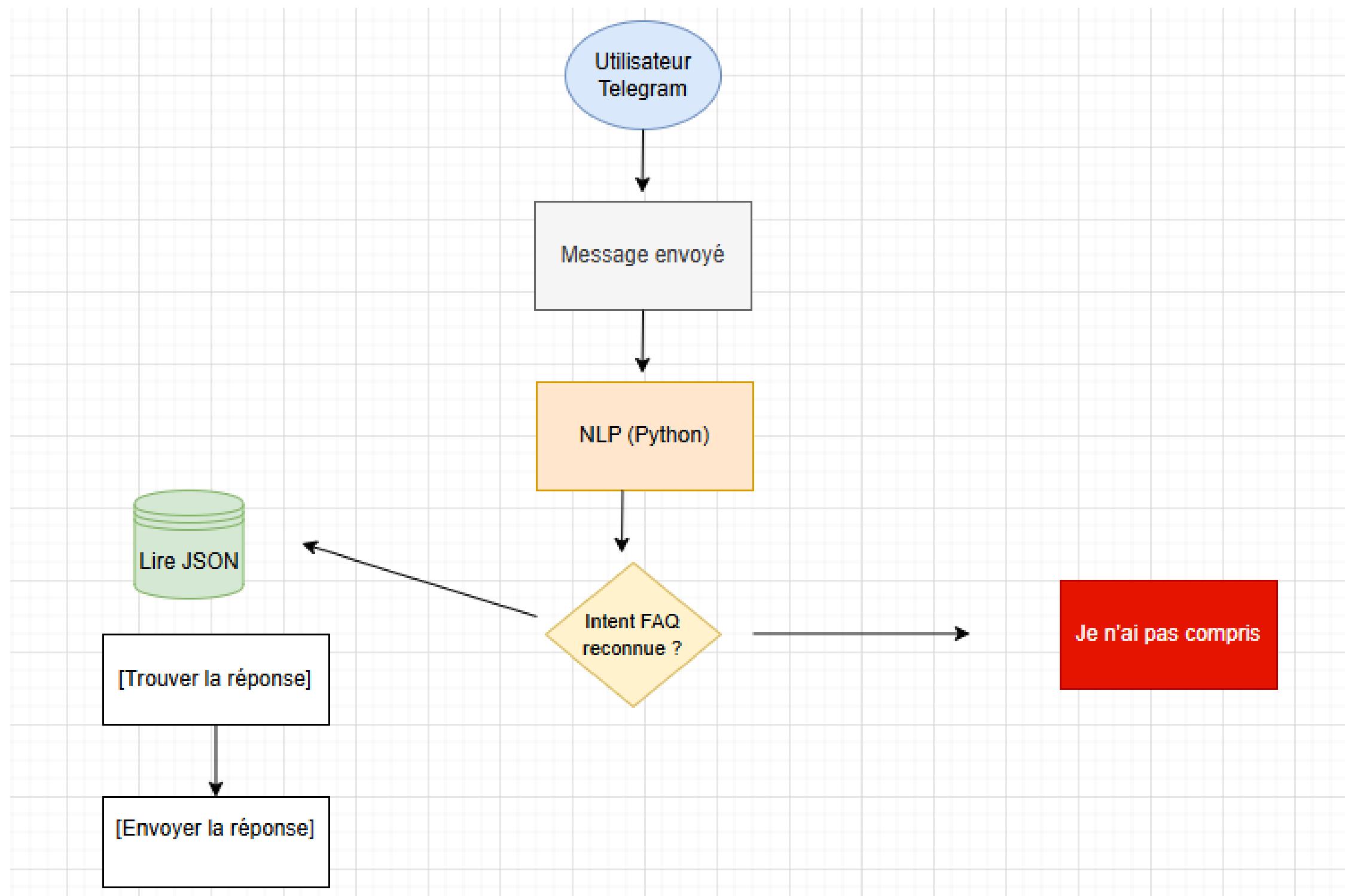
2-Source de données Interaction avec un fichier JSON (FAQ)

Lorsqu'un utilisateur envoie une question générale ou répétitive, telle que « Comment puis-je signaler un problème ? », « Quelles sont les heures d'ouverture du service d'assistance ? » ou "Quelle est la procédure de réclamation ?

Au lieu de consulter une base de données dynamique, il cherche les données dans un fichier JSON organisé qui fait office de base de connaissances statique.

Ce fichier JSON inclut une liste de paires question-réponse qui sont regroupées les FAQ (questions fréquemment posées) c'est à dire les plus courantes. Grâce au module de traitement du langage naturel (NLP), le chatbot peut comparer la question posée avec celles qui se trouvent dans le fichier, même si la formulation est différente. Lors que la correspondance est trouvée, il extrait automatiquement la réponse correspondante.

Cette réponse que le chatbot a trouvé, est ensuite mise en forme et envoyée à l'utilisateur directement via l'application Telegram. Cette opération est une circulation de données simplifiée et rapide qui ne demande pas de calculs complexes ou de communication avec des fichiers de suivi dynamiques. Cela assure une réponse idéale aux questions simples tout en diminuant la charge du chatbot.



D- Arbre de décisionnel

Dans le contexte du développement du chatbot SFR qui se base sur la technologie du traitement du langage naturel (NLP), une arbre décisionnel a été développé pour schématiser la logique de prise de décision du bot selon les messages qu'il reçoit. Cet arbre de décision illustre le chemin logique que prend le message d'un utilisateur depuis sa réception dans l'interface Telegram jusqu'à la génération d'une réponse appropriée.

❖ Étapes du traitement :

Réception du message de l'utilisateur via l'interface Telegram.

L'utilisateur initie un échange avec le chatbot via l'application , l'utilisateur bien sur formule une question

Analyse de l'intention à l'aide du NLP.

Le message est analysé par le module NLP développé en Python pour identifié l'intention. Cette identification est basée sur des techniques comme la classification de texte, la recherche de mots-clés et la similarité sémantique.

Prise de décision basée sur l'intention détectée :

Exemple 1 : Suivi des projets SFR

Si l'intention identifié correspond à **une requête liée au suivi des projets SFR**, le chatbot consulte le fichier Excel qui contient les informations sur les projets . Il collect les données liées à l'identifiant de projet

Cas 2 : question générale (FAQ)

Si la **demande est de nature globale** comme une question souvent posée, le chatbot appelle les fichier JSON qui contient une base de données de FAQ (questions/réponses). À l'aide d'un algorithme de correspondance de texte (ou de vectorisation), la correspondance est trouvée, la réponse est captée et envoyée à l'utilisateur.

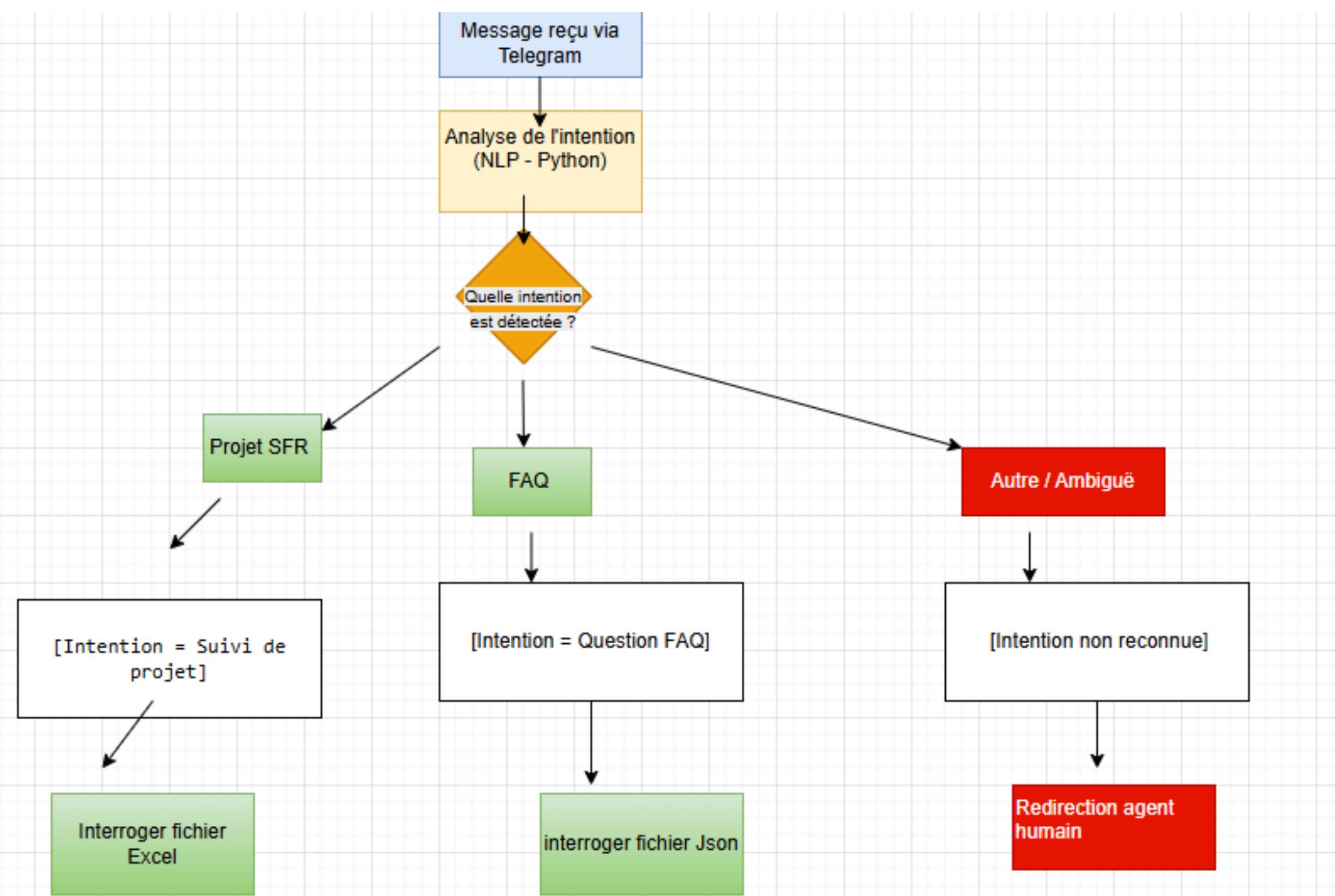
Cas 3 : intention non reconnue ou ambiguë

Si l'analyse NLP ne permet pas de préciser l'intention de l'utilisateur, le chatbot envoie un message d'incertitude , et il demande à l'utilisateur de réécrire la requête ou, si il ya une obligation le chatbot proposant une redirection vers un agent humain

⌚ Objectif de l'arbre :

Ce Système **assure une réponse organisée et rapide aux Exigences** de l'utilisateur, tout en offrant une séparation claire des différentes sources de données (Excel pour le suivi des projets, les fichiers JSON pour les FAQ). Il offre la possibilité de Segmenter la logique métier du chatbot et de garantir son évolutivité.

vous trouverez un diagramme d'arbre de décision qui explique le fonctionnement logique de chatbot qui se base sur la détection d'intention à l'aide du NLP. Ce diagramme **qui montre les différentes pahases du traitement d'un message reçu sur Telegram**, de l'analyse de l'intention à la réponse à l'utilisateur. Trois hypothèses sont prévue : la détection d'une intention liée au suivi d'un projet, ou bien d'une question fréquemment posée (FAQ) ou d'une intention non reconnue nécessitant une Réorientation vers un agent humain. Ce diagramme illustre la logique de décision intégrée dans le système.



E-. Utilisation du pipeline de traitement IA / NLP

Le pipeline de traitement AI / NLP convertit un message librement rédigé par un utilisateur (c'est à dire l'utilisateur peut écrire ce qu'il veut) en une réponse intelligente et adaptée au contexte. Cette procédure passe par une chaîne d'étapes automatisées, de la réception du message à la génération de la réponse. Voici une explication ci-dessous détaillée sur les principales étapes de ce pipeline :

Etapes du pipeline NLP dans le projet

➤ Réception d'un message via Telegram

Le chatbot récupère le message de l'utilisateur en temps réel via l'API Telegram (à titre d'exemple : "donne moi les informations pour P123456.").

➤ Nettoyage et prétraitement du texte

Le message est transformé en minuscules, les symboles de ponctuation et les caractères spéciaux sont supprimés. Le texte est uniformisé pour une analyse optimisée

➤ Tokenisation

Le message est découpé en parties appelées « tokens », généralement des mots.

Il s'agit généralement de mots : "je", "veux", "suivre", "projet", "SFR".

➤ Analyse sémantique et vectorisation

Dans ce stade , le texte est transformé en vecteurs numériques à l'aide d'un modèle NLP (par exemple TF-IDF, Word2Vec ou BERT embedding).

Cela offre la possibilité à l'algorithme de comprendre le sens du message.

➤ Détection de l'intention

Noyau du pipeline : un modèle de classification NLP (entraîné à l'aide de Python, Scikit-learn et un modèle pré-entraîné tel que spaCy ou BERT) précise l'intention de l'utilisateur.

Voici les trois intentions qu'on peut trouver :

- poursuivre le projet SFR
- Question FAQ
- Requête non reconnue

➤ Extraire des entités (optionnel)

des entités comme le nom du responsable du projet , la date, le numéro de projet, etc. peuvent être récupérées pour aider à construire une réponse.

➤ Accès à la base de données de l'entreprise ou aux fichiers

En fonction de l'objectif identifié :

S'il s'agit d'une demande concernant un projet , le bot demande un fichier Excel qui contient les données du projet du client

Exemple d'un scénario spécifique dans notre chatbot :

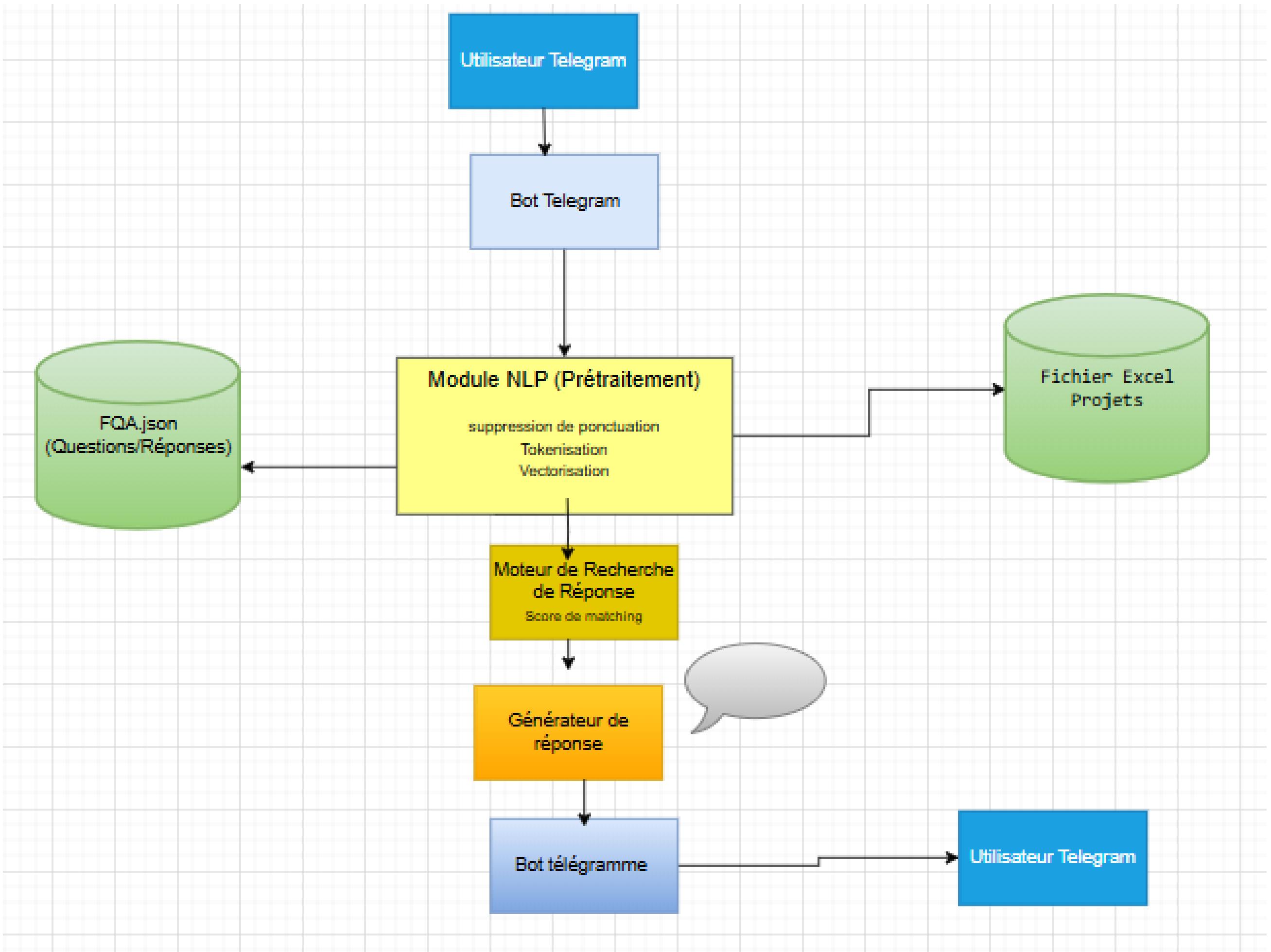
Un utilisateur écrit : "Bonjour, je veux les informations pour le projet PXXXXXX?

Le pipeline efface le message, détecte les mots-clés " PXXXXXX " et " projet ", et le modèle NLP classe l'intention comme une continuation de projet.

Le bot accède ensuite au fichier Excel pour connaître l'état du dossier et répond automatiquement :

« Votre fichier est en cours de traitement par SFR ».

Ce diagramme décrit en détaille le mécanisme de traitement interne



F: Schéma d'architecture logique

Le diagramme ci-dessous explique l'architecture logique du système de chatbot développé dans le contexte de ce projet. Il montre clairement les différents parties fonctionnelles du système et la manière dont ils échangent pour prendre en charge les demandes des utilisateurs.

L'échange se lance lorsque l'utilisateur envoie une question par l'application Telegram. Cette question est reçue par le bot, l'application télégramme est une interface de communication entre l'utilisateur et le système intelligent.

Le message est ensuite transmis au serveur d'application NLP/IA, qui est le noyau du traitement du langage naturel. Ce serveur contient les éléments suivants :

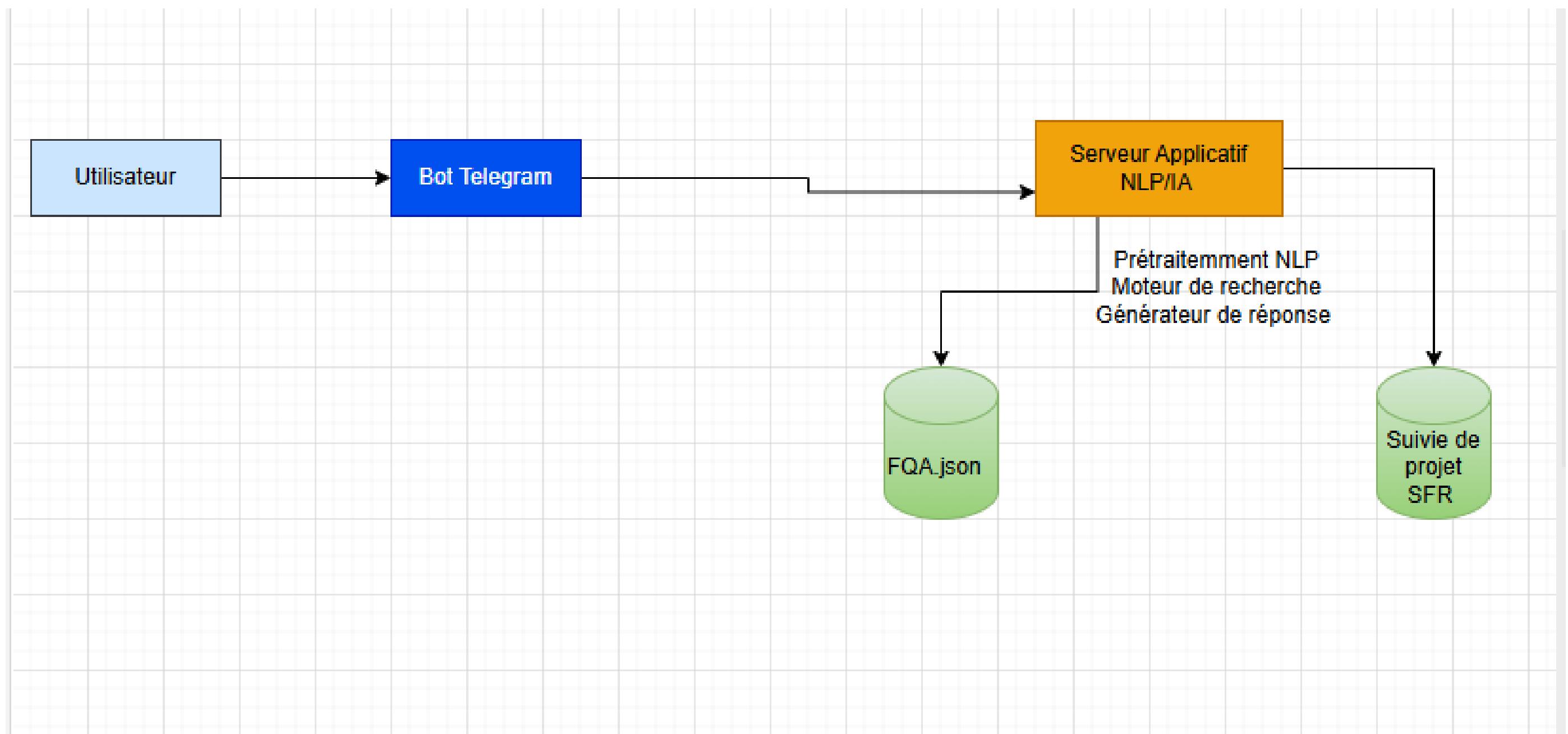
Le module de prétraitement NLP, qui est chargé du filtrage et de la transformation du texte (suppression de la ponctuation, tokenisation, vectorisation, etc.)

C'est Un système qui recherche les réponses , il analyse le message pour obtenir la réponse la plus exact parmi les données disponibles.

- Notre générateur de réponse qui prépare et structure la réponse avant de l'envoyer au bot.
- Le serveur demande deux sources de données :
- Les deux fichiers JSON FQA qui contient beaucoup des informations organisé sous forme d'un couple " question
====➔ réponse "
- Un fichier Excel de suivi de projet SFR qui contient les informations sur les projets

Après la génération de la réponse , elle est renvoyée au bot Telegram , et ce dernier la communique à l'utilisateur final.

Ce schéma montre exactement la logique de connexion entre éléments , qui est la suivante



Chapitre 4 : Conception est développement

4.1 Analyse de l'architecture globale

A. Description générale de la solution

Le système se base sur une conception modulaire construite autour d'un bot Telegram connecté avec un moteur NLP.

L'objectif c'est que les utilisateurs peuvent poser des questions via l'interface Telegram, et recevoir des réponses adaptées, les réponses sont tirées de deux sources de données [suivi du projet SFR](#), ou [des fichiers JSON de type FAQ](#)

Le code source de notre bot contient 5 fichiers :

[Dataset.py](#) : Ce script automatise la création d'un jeu de données affecté à l'entraînement d'un modèle de langage naturel. Il débute par télécharger trois fichiers depuis Google Drive : [deux fichiers JSON](#) contenant des informations sur des procédures et des FAQ, et les informations des techniciens, ainsi qu'un [fichier Excel](#) contenant les données sur des projets. Il lit ensuite ces fichiers, [extrait les informations appropriées](#) et produit automatiquement des phrases sous forme de questions, chacune étant liée à une type d'intention : « [procédure](#) », « [faq](#) » ou « [projet](#) ». Les colonnes obligatoires du fichier Excel sont [vérifiées et uniformisées](#) pour éviter les erreurs de lecture. L'ensemble des questions est combiné afin de supprimer tout ordre successif, puis enregistrer dans un fichier CSV nommé [intent_dataset.csv](#), qui sera utilisé pour l'apprentissage d'un classificateur d'intentions.

text,label
1 J'ai une question : quel est le numéro de téléphone de Maxim ?,faq
2 Quel est le lien d'accès à la plateforme Axis ?,faq
3 Quel est l'état du projet P29766 ?,projet
4 Quelle est la date de réception pour P29755 ?,projet
5 Où en est le projet P29771 ?,projet
6 Quelle est la date de réception pour P29766 ?,projet
7 Donne-moi les détails du projet P29790,projet
8 J'ai une question : qui sont les membres de l'équipe SFR ?,faq
9 Quel est l'état du projet P29722 ?,projet
10 Quel est le numéro de téléphone de Hichem ?,faq
11 Quelle est la date de réception pour P29784 ?,projet
12 Quel est l'état du projet P29699 ?,projet
13 Où en est le projet P29725 ?,projet
14 J'ai une question : quels sont les projets sur lesquels travaille l'équipe ?,faq
15 Où en est le projet P29740 ?,projet
16 Donne-moi les détails du projet P29785,projet
17 Quel est l'état du projet P29740 ?,projet
18 J'ai une question : quel est le mail du chef de projet sud-est ?,faq
19 Donne-moi les détails du projet P29719,projet
20 J'ai une question : quel est le lien d'accès à la plateforme Axis ?,faq
21 Quel est l'état du projet P29700 ?,projet

Extrait du jeu de données

Modele d'entraînement.py : Ce script a pour but de créer et d'entraîner un modèle de classification automatique des intentions. Il utilise le fichier [intent_dataset.csv qu'on a déjà généré](#) qui contient des exemples de questions liés à leurs sections (procédure, FAQ ou projet). Le programme commence par importer ces données, préciser l'ensemble des étiquettes uniques et leur attribuer des identifiants numériques pour simplifier le traitement. Il utilise ensuite le tokenizer pré-entraîné **DistilBERT** pour transformer chaque question en une représentation numérique idéale à l'apprentissage. Les données sont préparées et structurées sous forme d'un jeu exploitable par PyTorch, ensuite un modèle de type *AutoModelForSequenceClassification* est réglé pour gérer la classification multi-catégories. L'entraînement est effectué grâce à la classe **Trainer** de la bibliothèque Transformers, selon des paramètres standardisés (nombre d'époques, taille des lots, stratégie de sauvegarde). Après ce processus, le modèle final ainsi que le tokenizer sont enregistrés localement, prêts à être réutilisés pour prédire automatiquement la catégorie d'une nouvelle question.

Nouveau nom (D:) > Soutenance MOIS 9 > env > intent_classifier_model

Nom	Modifié le	Type	Taille
config	14/07/2025 17:16	Fichier source JSON	1 Ko
model.safetensors	14/07/2025 17:16	Fichier SAFETENS...	261 559 Ko
special_tokens_map	14/07/2025 17:16	Fichier source JSON	1 Ko
tokenizer	14/07/2025 17:16	Fichier source JSON	695 Ko
tokenizer_config	14/07/2025 17:16	Fichier source JSON	2 Ko
training_args.bin	14/07/2025 17:16	Fichier BIN	6 Ko
vocab	14/07/2025 17:16	Document texte	227 Ko

Structure du modèle de classification d'intentions

Le modèle de **classification d'intentions** réalisé est basé sur la bibliothèque **Hugging Face Transformers**, qui donne l'occasion d'utiliser des modèles pré-entraînés et de les personnaliser à des tâches ciblées, à titre d'exemple l'**identification d'intentions dans des messages textuels**. La mise en place et le traitement des données ont été faite par les bibliothèques **datasets**, **pandas** et **numpy**, facilite le chargement, un formatage et un nettoyage efficaces du jeu de données. L'apprentissage est faite avec le framework **PyTorch**, utilisé pour gérer les calculs et améliorer les paramètres du réseau. L'utilisation conjointe de l'API **Trainer** et de **TrainingArguments** offre la possibilité de décrire les hyperparamètres d'entraînement (nombre d'itérations, taille des lots, taux d'apprentissage, etc.) tout en automatisant les étapes clés du processus. Après l'entraînement, le modèle a été enregistré dans un dossier contenant plusieurs éléments nécessaires : le fichier **config** qui présente la conception et les paramètres du modèle, alors que le **model.safetensors** sauvegarde les poids appris. Les fichiers **tokenizer** et **tokenizer_config** précisent la méthode de tokenisation, c'est-à-dire la conversion des phrases en séquences numériques compréhensibles par le modèle.

Le fichier `special_tokens_map` marque les tokens spéciaux utilisés comme [CLS] ou [SEP], indispensable au traitement correct des entrées. Le `vocabulaire complet est sauvegardé dans le fichier vocab`, et les hyperparamètres d'apprentissage sont conservé dans `training_args.bin`, garantissant la reproductibilité. Grâce à cette organisation, le modèle peut être rempli simplement via `AutoTokenizer` et `AutoModelForSequenceClassification` pour de prédire automatiquement l'intention d'un nouveau message.

`faq_sfr_complet.py` : Ce script Python représente un système puissant de questions-réponses basé sur le traitement automatique du langage naturel. Utilise deux modèles essentiels : un classificateur d'intentions personnalisé, est importé à l'aide de la bibliothèque Hugging Face Transformers, et un modèle d'embedding sémantique de Sentence Transformers qui assure de mesurer la proximité de sens entre différentes phrases.

La première chose , le classificateur est débuté via la fonction pipeline de Hugging Face, il importe un modèle et un tokenizer entraînés. Ce classificateur a un rôle c'est il 'identifier la catégorie ou l'intention principale d'une question posée par un utilisateur, a titre d'exemple une demande d'aide sur le process de traitement ou bien une question sur un projet . En même temps , le modèle SentenceTransformer (ici "all-MiniLM-L6-v2") est importé afin de convertir les phrases en vecteurs numériques (embeddings), ce qui simplifie leur comparaison fine du point de vue sémantique.

Et aussi , le script lit deux fichiers JSON qui contient des ensembles de questions-réponses (FAQ) amélioré , structurés par type d'intentions. Ces fichiers sont importé en mémoire sous forme de dictionnaires Python (`faq_augmented` et `faq_sfr_complet`), où chaque clé illustre une intention et est liée à une liste de questions et leurs réponses.

La fonction `find_best_match_semantic` recoit comme arguments une question de l'utilisateur et une liste de questions provenant des FAQ. Elle encode toutes ces questions en vecteurs en utilisant le modèle d'embedding, puis calcule la similarité cosinus entre la question entrée et chaque question de la FAQ. Elle identifie la question la plus proche selon ce score, et si ce score va au-delà un seuil de confiance paramétrable (ici fixé à 0.6), elle renvoie la réponse équivalente . Sinon, elle renvoie None.

Finalement , la fonction `get_answer` obtient un label (intention) et aussi que la question d'origine. Elle cherche ce label dans les deux bases FAQ chargées. Si elle trouve une liste de questions/réponses liés , elle utilise la fonction de recherche sémantique pour retirer la réponse la plus convenable . Si il n'a pas trouvé la résultat convenable elle renvoie la première réponse disponible pour ce label. Si la donnée liée est une chaîne de caractères, elle la renvoie directement. Si aucun résultat n'est trouvé, la fonction renvoie None.

En résumé , ce programme associe la détection d'intentions et la recherche sémantique dans des FAQ structurés pour proposer des réponses appropriés et convenable à des questions formulées évidemment par des utilisateurs

Ce code présente le module principale du système de réponse automatique. Il se base sur un modèle de classification d'intention qui est déjà entraîné ainsi qu'une structure de recherche sémantique qui utilise des vecteurs d'embedding pour d'identifier et donner la réponse la plus convenable à la question posée par l'utilisateur.

```
import json
from transformers import pipeline
from sentence_transformers import SentenceTransformer, util

# === Chargement du modèle de classification (entraîné par toi) ===
classifier = pipeline(
    "text-classification",
    model="D:/Soutenance MOIS 9/env/intent_classifier_model",
    tokenizer="D:/Soutenance MOIS 9/env/intent_classifier_model"
)

# === Chargement du modèle d'embedding sémantique ===
embedder = SentenceTransformer("all-MiniLM-L6-v2")

# === Chemins des fichiers JSON ===
faq_augmented_path = r"D:\Soutenance MOIS 9\env\faq_augmented.json"
faq_sfr_complet_path = r"D:\Soutenance MOIS 9\env\faq_sfr_complet.json"

with open(faq_augmented_path, 'r', encoding='utf-8') as f:
    faq_augmented = json.load(f)

with open(faq_sfr_complet_path, 'r', encoding='utf-8') as f:
    faq_sfr_complet = json.load(f)

# === Recherche sémantique dans une liste de questions ===
def find_best_match_semantic(question, faqs):
    questions_list = [item['question'] for item in faqs]
    question_embedding = embedder.encode(question, convert_to_tensor=True)
    faq_embeddings = embedder.encode(questions_list, convert_to_tensor=True)
    cosine_scores = util.pytorch_cos_sim(question_embedding, faq_embeddings)[0]

    best_idx = cosine_scores.argmax().item()
    best_score = cosine_scores[best_idx].item()

    if best_score > 0.6: # seuil de confiance ajustable
        return faqs[best_idx]['réponse']
    return None
```

extrait de code source

Boot.py : ce script configure un assistant conversationnel qui fonctionne sur Telegram, il facilite l'échange d'informations entre les utilisateurs et le système qu'on a développé dans le cadre de ce projet SFR. Il inclut un modèle de traitement automatique du langage naturel (basé sur spaCy) pour déterminer les expressions de salutation et analyser le contenu des messages reçus.

Selon le type de la question posée, le bot est compétent soit de proposer une réponse convenable à partir d'une base de connaissances (FAQ), soit d'extraire des données spécifiques liées à un projet en examinant un fichier Excel préchargé

L'accès aux projets est protégé par une vérification du nom de l'utilisateur pour garantir un usage sécurisé. Le fonctionnement du bot se base sur une conception asynchrone, ce qui garantit une disponibilité continue pour répondre en temps réel aux demandes via l'interface Telegram.

Fonctionnalités principales du script:

Voici les Fonctionnalités de ce script :

- Il met en place un bot Telegram Bot SFR, destiné pour échanger efficacement avec les utilisateurs
- Il traite des commandes spécifiques comme /start, /faq, /projet et /contact, facilitant la navigation et l'accès aux informations.
- À cause de ce modèle de traitement du langage naturel entraîné avec spaCy, il est capable de identifier automatiquement les messages de salutation (comme "bonjour" ou "salut")
- Il intègre aussi un classificateur NLP (faq_sfr_complet) qui comprends l'intention des utilisateurs et de leur donner une réponse convenable extraite d'une base de données de questions fréquemment posées
- Quand un utilisateur mentionne un code projet (par exemple : P12345), le bot recherche dans un fichier Excel contenant les informations de ce projet, et renvoie les détails comme les dates, le type de dossier ou l'état d'avancement
- Enfin, le script est conçu pour télécharger automatiquement les fichiers nécessaires (tels que la FAQ, les procédures ou les tableaux Excel) à partir de Google Drive, en utilisant l'outil gdown.

ModeleSalutation.py :

Ce script utilise la bibliothèque spaCy pour entraîner un modèle de classification de texte en français, capable de distinguer si une phrase est une salutation ou non. Il commence par définir un jeu de données composé de phrases annotées avec deux catégories : "SALUTATION" et "NON". Ensuite, un pipeline NLP vide est initialisé et enrichi d'un composant de classification (textcat). Les étiquettes sont ajoutées au modèle, qui est ensuite entraîné sur plusieurs itérations à l'aide de lots de données générés automatiquement. Chaque exemple est converti au format attendu par spaCy, puis utilisé pour ajuster les poids du modèle. Une fois l'entraînement terminé, le modèle est sauvegardé sur disque, prêt à être utilisé dans une application de traitement du langage naturel, comme un assistant virtuel ou un chatbot.

```
45
46 # Créer un modèle vide
47 nlp = spacy.blank("fr")
48
49 # Ajouter le pipe de text categorization
50 textcat = nlp.add_pipe("textcat")
51
52 # Ajouter les labels
53 textcat.add_label("SALUTATION")
54 textcat.add_label("NON")
55
56 # Entraînement
57 n_iter = 10
58 optimizer = nlp.begin_training()
59
60 for i in range(n_iter):
61     random.shuffle(train_data)
62     losses = {}
63     batches = minibatch(train_data, size=compounding(4.0, 32.0, 1.001))
64     for batch in batches:
65         texts, annotations = zip(*batch)
66         examples = []
67         for text, annotation in zip(texts, annotations):
68             doc = nlp.make_doc(text)
69             examples.append(Example.from_dict(doc, annotation))
70         nlp.update(examples, sgd=optimizer, drop=0.2, losses=losses)
71         print(f"Iteration {i+1}, Losses: {losses}")
72
73 # Après la boucle d'entraînement
74 output_dir = "./mon_modele_salutation" # dossier où enregistrer le modèle
75 nlp.to_disk(output_dir)
76 print(f"Modèle sauvegardé dans {output_dir}")
77
78
```

extrait de code

B. Choix de la plateforme Telegram pour le bot

Telegram a été choisi comme interface de communication en raison de sa facilité d'intégration via les bots, de sa riche documentation et de sa facilité de déploiement via l'API Telegram Bot de BotFather. Cela permet un échange transparent et multiplateforme entre les utilisateurs et le système

C. Choix du NLP pour comprendre les questions

Le NLP permet au bot d'interpréter le langage naturel des utilisateurs. Le système devient ainsi plus intelligent, plus souple et plus interactif. Il ne s'agit plus de simples commandes fixes, mais d'une compréhension sémantique des demandes

4.2 Choix des technologies utilisées

A. Python et ses bibliothèques :

Le développement du chatbot repose sur quatre scripts essentiels accompagnés d'un jeu de données spécifique. Ce dernier a été constitué à partir de trois sources distinctes : un fichier Excel assurant le suivi des projets, ainsi que deux fichiers JSON contenant respectivement des données relatives aux procédures et aux questions fréquemment posées (FAQ). Ces fichiers, hébergés sur Google Drive, sont automatiquement récupérés et analysés. Les informations extraites servent à générer des phrases interrogatives, chacune étant associée à une catégorie d'intention : « procédure », « FAQ » ou « projet ».

L'un des scripts a pour rôle d'entraîner un modèle de classification d'intentions à partir du jeu de données construit. Ce modèle est développé à l'aide de la bibliothèque Hugging Face Transformers, spécialisée dans le traitement du langage naturel.

Par ailleurs, le script nommé `faq_sfr_complet.py` joue un rôle central dans le fonctionnement du système de questions-réponses. Il combine deux éléments clés : un classificateur d'intention personnalisé et un modèle de similarité sémantique fourni par Sentence Transformers. Le premier permet d'identifier la nature de la requête utilisateur, tandis que le second calcule la proximité de sens entre la question posée et les réponses disponibles, assurant ainsi une réponse pertinente et contextuelle.

Dans cette partie, nous allons détailler les bibliothèques utilisées dans chaque script afin de mieux comprendre leur rôle dans le fonctionnement global du chatbot

Bibliothèque utilisé pour générée le jeu de donné : (`dataset.py`)

■ `pandas` (import pandas as pd) : Utilisée pour lire, manipuler et filtrer les données du fichier Excel contenant les projets

■ `json` : Sert à charger et manipuler les fichiers JSON, notamment ceux contenant les procédures et la FAQ

■ `gdown` : Permet de télécharger automatiquement des fichiers depuis Google Drive à l'aide de leurs identifiants (IDs)

■ `random` : Utilisée pour mélanger aléatoirement les exemples générés, afin de diversifier l'ordre des entrées dans le fichier final.

→ Dictionnaire stockant les identifiants uniques des fichiers hébergés sur Google Drive (un pour les procédures, un pour la FAQ et un pour les projets)

```
file_ids = {  
    "procedure": "1tQHxKRmejdD7SciBTrA5V_jBA4n_2Sdl",  
    "faq": "1XtZesLx35tN-Y0BiOyKfYgmlAlbNkQz5",  
    "projets": "1-qnd98mvhi3V4v4Jr46leyelaCH45Mmn"  
}
```

Téléchargement automatique de ces trois fichiers : deux en JSON et un fichier Excel.

```
gdown.download(id=file_ids["procedure"], output="procedure_sfr.json", quiet=False)  
gdown.download(id=file_ids["faq"], output="faq_sfr_complet.json", quiet=False)  
gdown.download(f"https://drive.google.com/uc?id={file_ids['projets']}", output="avancement_memoir_modif.xlsx",  
quiet=False)
```

→ Création d'une liste vide destinée à stocker toutes les questions générées et leur catégorie (procédure, FAQ ou projet).

```
examples = []
```

Bibliothèque utilisée dans le scripte (entraînement.py)

✚ **datasets (de Hugging Face)** : Permet de charger et de manipuler facilement des jeux de données, notamment pour le NLP. Utilisé ici pour le typage ClassLabel (même s'il n'est pas exploité plus loin dans le code)

✚ **transformers** : Bibliothèque centrale de Hugging Face utilisée pour charger des modèles pré-entraînés (comme DistilBERT), gérer les tokenizers, les modèles de classification, et l'entraînement via Trainer

✚ **pandas** : Utilisée pour charger et manipuler les données à partir d'un fichier CSV (intent_dataset.csv)

✚ **numpy** : Bibliothèque de calcul scientifique, bien qu'importée ici, elle n'est pas utilisée directement dans ce script

✚ **torch (PyTorch)** : Utilisée pour créer un dataset compatible avec le framework d'entraînement, et pour manipuler les tenseurs nécessaires à l'apprentissage automatique.

Bibliothèque utilisée dans le scripte (Faq_sfr_complet.py) : système performant de questions-réponses

👉 **json** : Il s'agit d'un module standard de Python permettant de lire et d'écrire des fichiers au format JSON. Il a été utilisé ici pour importer des bases de données de questions-réponses stockées localement.

👉 **transformers (Hugging Face)** : Cette bibliothèque facilite l'exploitation de modèles de traitement du langage pré-entraînés. Elle a servi à mettre en place un pipeline de classification d'intentions, basé sur un modèle personnalisé

👉 **sentence-transformers** : Cette bibliothèque est conçue pour générer des représentations vectorielles (embeddings) de phrases. Elle est utilisée ici pour comparer sémantiquement les questions posées par l'utilisateur avec celles présentes dans la base de données, via un calcul de similarité cosinus.

B. Modèle NLP

Nom	Description / Rôle
mon_modele_salutation (spaCy)	Modèle entraîné localement pour reconnaître les salutations (bonjour, salut...).
DistilBERT (distilbert-base-uncased)	Modèle Transformers pré-entraîné et affiné pour classer les intentions (procédure, FAQ, projet)
all-MiniLM-L6-v2	Modèle Sentence Transformers utilisé pour la recherche sémantique et la comparaison de similarité entre questions FAQ.
intent_classifier_model	Modèle DistilBERT finement entraîné sur la dataset (intent_dataset.csv) pour reconnaître les intentions des questions utilisateurs.

Présentation de l'ensemble des modèles NLP intégrés dans les différents scripts.

C. Environnement d'exécution : Spyder

Le développement a été effectué dans l'environnement Spyder, qui est intégré à l'environnement Anaconda. Cela permet de surveiller le code de près, d'exécuter rapidement les blocs et de voir clairement les résultats

4.3 Conception fonctionnelle du bot

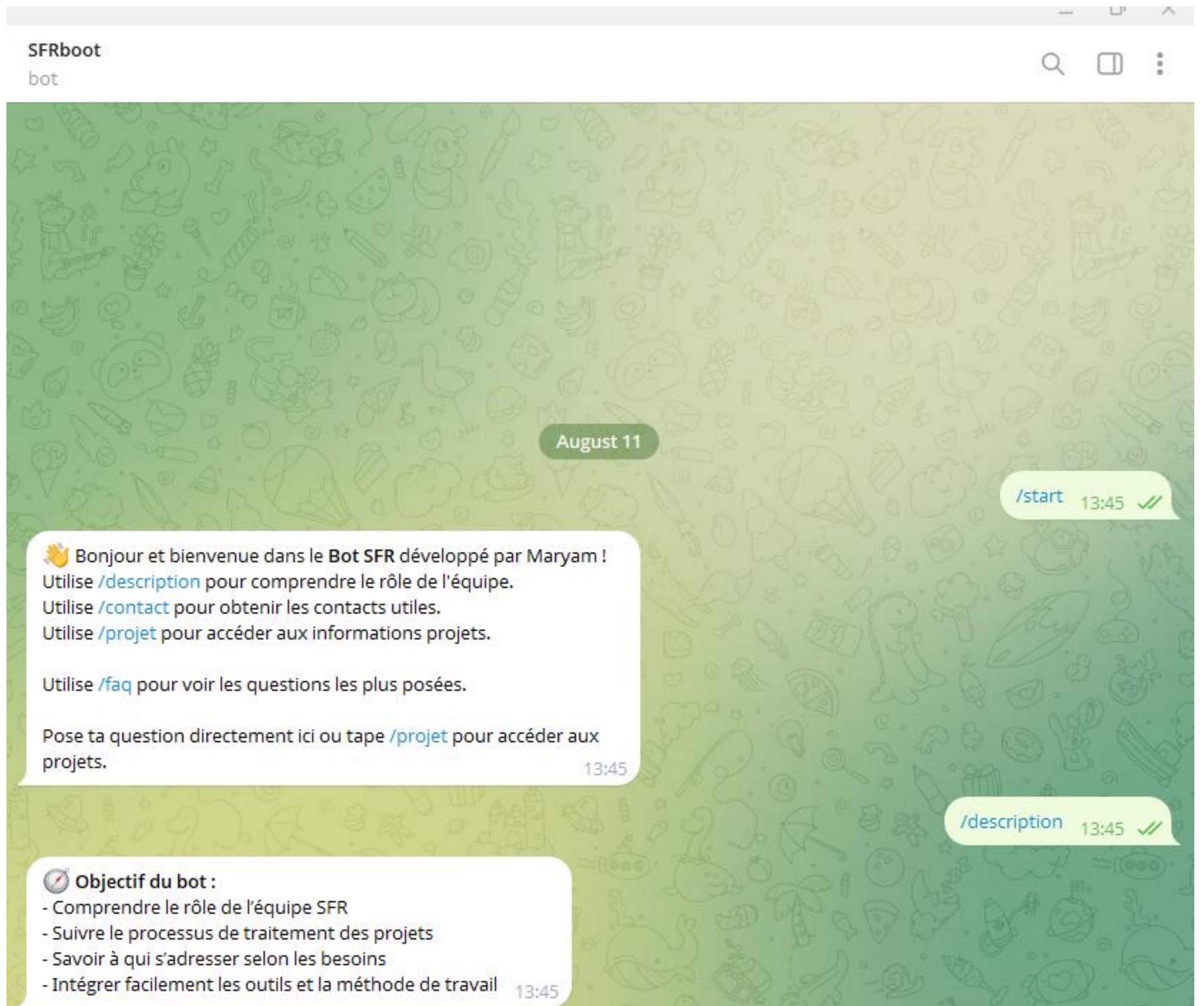
A. Gestion des commandes principales

Les commandes telles que /start , /description ,/Contact , /projet /FAQ



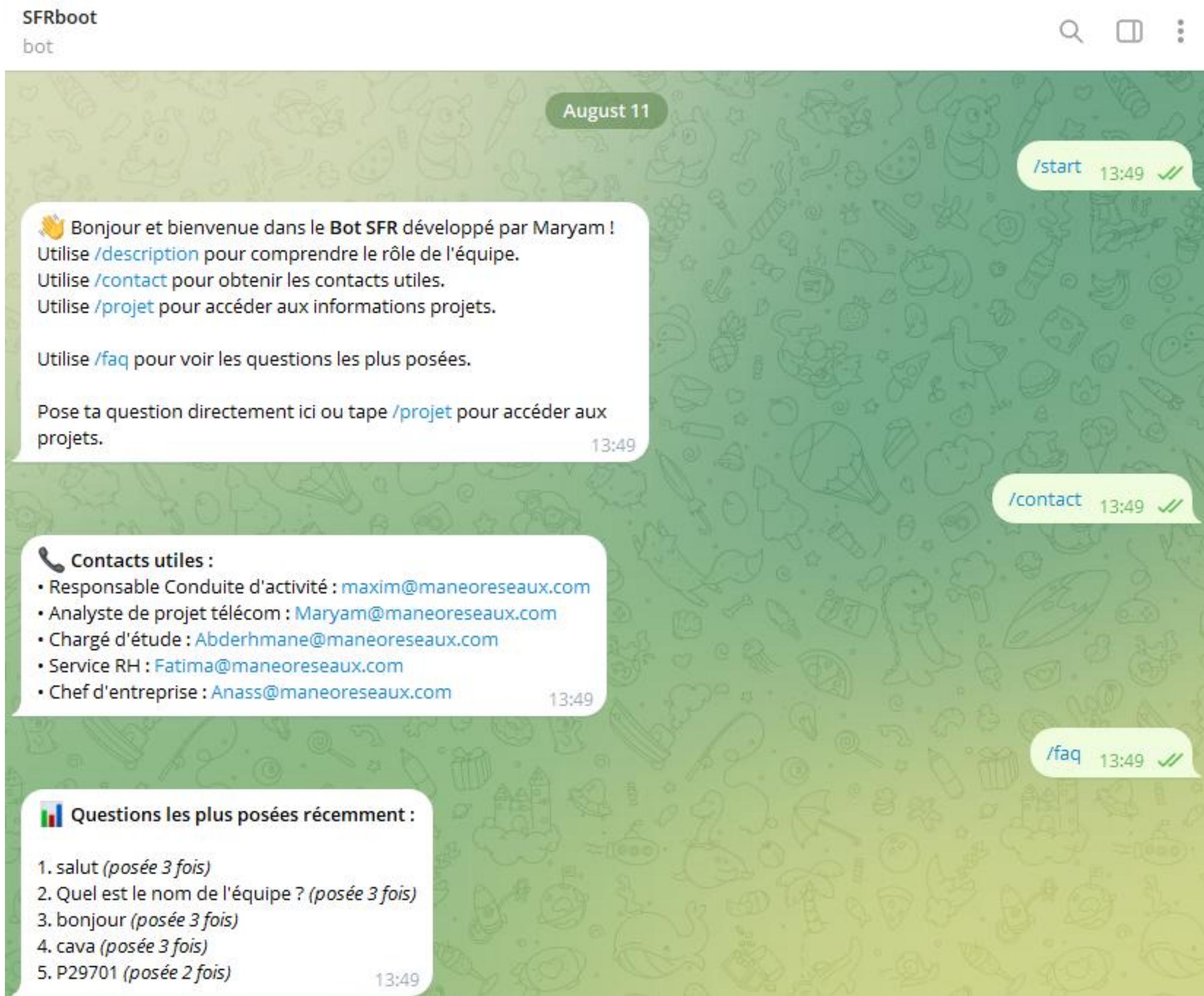
commandes de bases

La figure ci-dessous illustre l'interface d'accueil du bot SFR conçu dans le cadre du projet. Dès l'exécution de la commande /start, l'utilisateur reçoit un message de bienvenue présentant les principales fonctionnalités du bot, telles que la consultation de la description de l'équipe, l'accès aux contacts, la récupération d'informations sur les projets et la consultation d'une FAQ. Cette interface guide également l'utilisateur sur la manière d'interagir avec le système



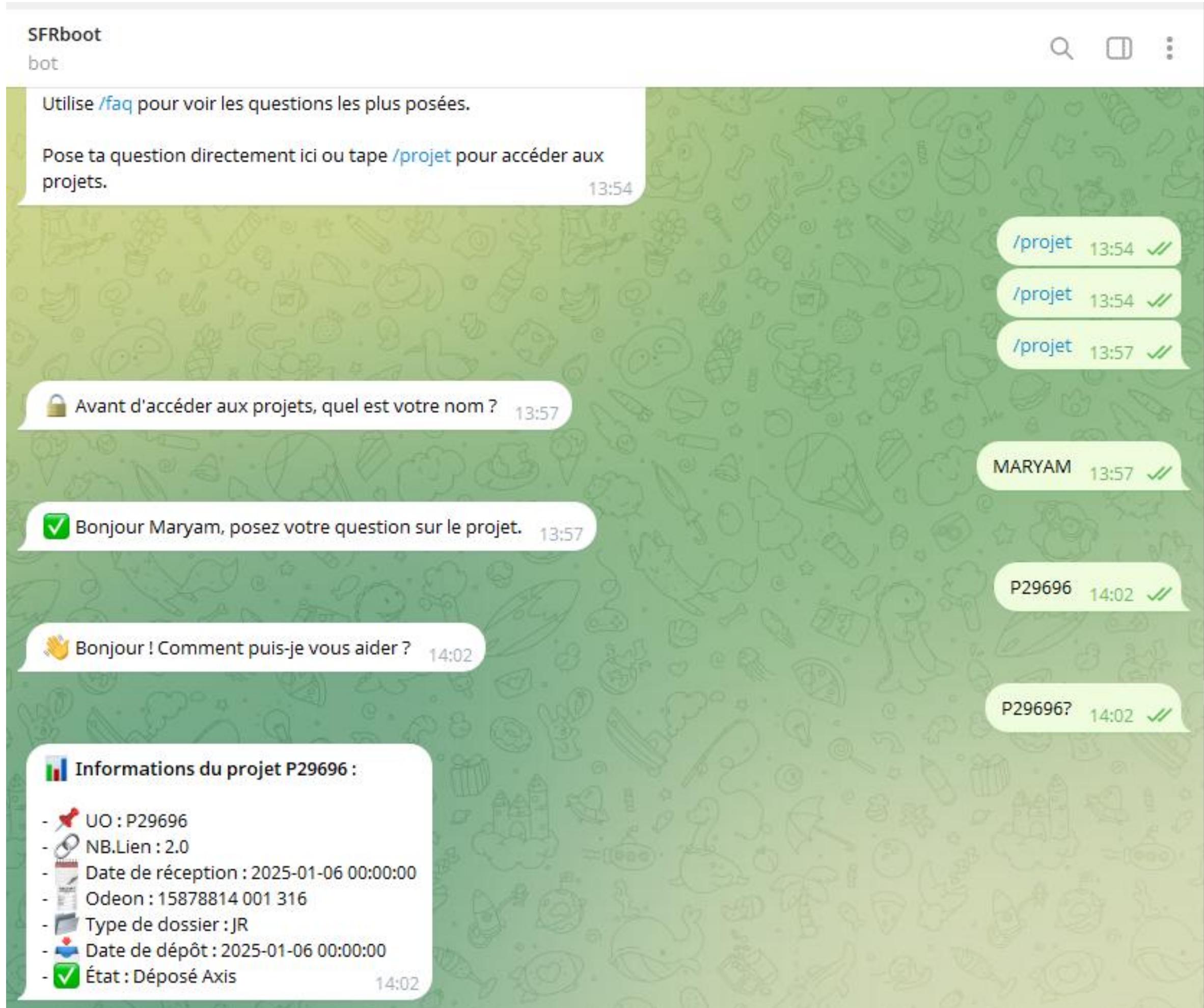
Commande description

L'illustration ci-dessous présente un échange type avec le bot SFR, développé dans le cadre du projet. Après l'exécution de la commande /start, l'agent conversationnel affiche un message de bienvenue détaillant les différentes commandes disponibles pour accéder aux informations sur l'équipe, les contacts, les projets et la FAQ. L'envoi de la commande /description permet de visualiser les objectifs du bot, notamment la compréhension du rôle de l'équipe SFR, la maîtrise du processus de traitement des projets, l'identification des interlocuteurs selon les besoins et la facilitation de l'accès aux outils ainsi qu'à la méthodologie de travail



Commandes contact et FAQ

La figure ci-dessus présente une interaction avec le bot Telegram développé dans le cadre du projet, illustrant son comportement face aux principales commandes. À travers la commande /start, l'utilisateur reçoit un message d'accueil décrivant les fonctionnalités offertes et la manière de les utiliser. La commande /contact renvoie la liste des interlocuteurs clés du projet, accompagnée de leurs adresses professionnelles, ce qui permet un accès rapide aux personnes concernées. Enfin, la commande /faq affiche les questions les plus fréquentes ainsi que leur fréquence d'apparition, facilitant la résolution immédiate des demandes récurrentes. Cette interface met en évidence la clarté de la structure et l'efficacité de l'outil dans la diffusion d'informations essentielle



commande projet

La commande /projet offre à l'utilisateur la possibilité de consulter les détails d'un projet stocké dans le système. Lors de son exécution, le bot invite d'abord l'utilisateur à s'authentifier en indiquant son nom. Après validation, celui-ci peut saisir un numéro de projet précis. Le bot procède alors à une recherche dans la base de données et affiche un résumé clair comprenant les informations essentielles : code du projet, nombre de liens, date de réception, identifiant Odeon, catégorie du dossier, date de dépôt et statut actuel. Cette fonctionnalité permet de regrouper les données importantes en un seul affichage, optimisant ainsi le suivi et l'accès rapide aux informations.



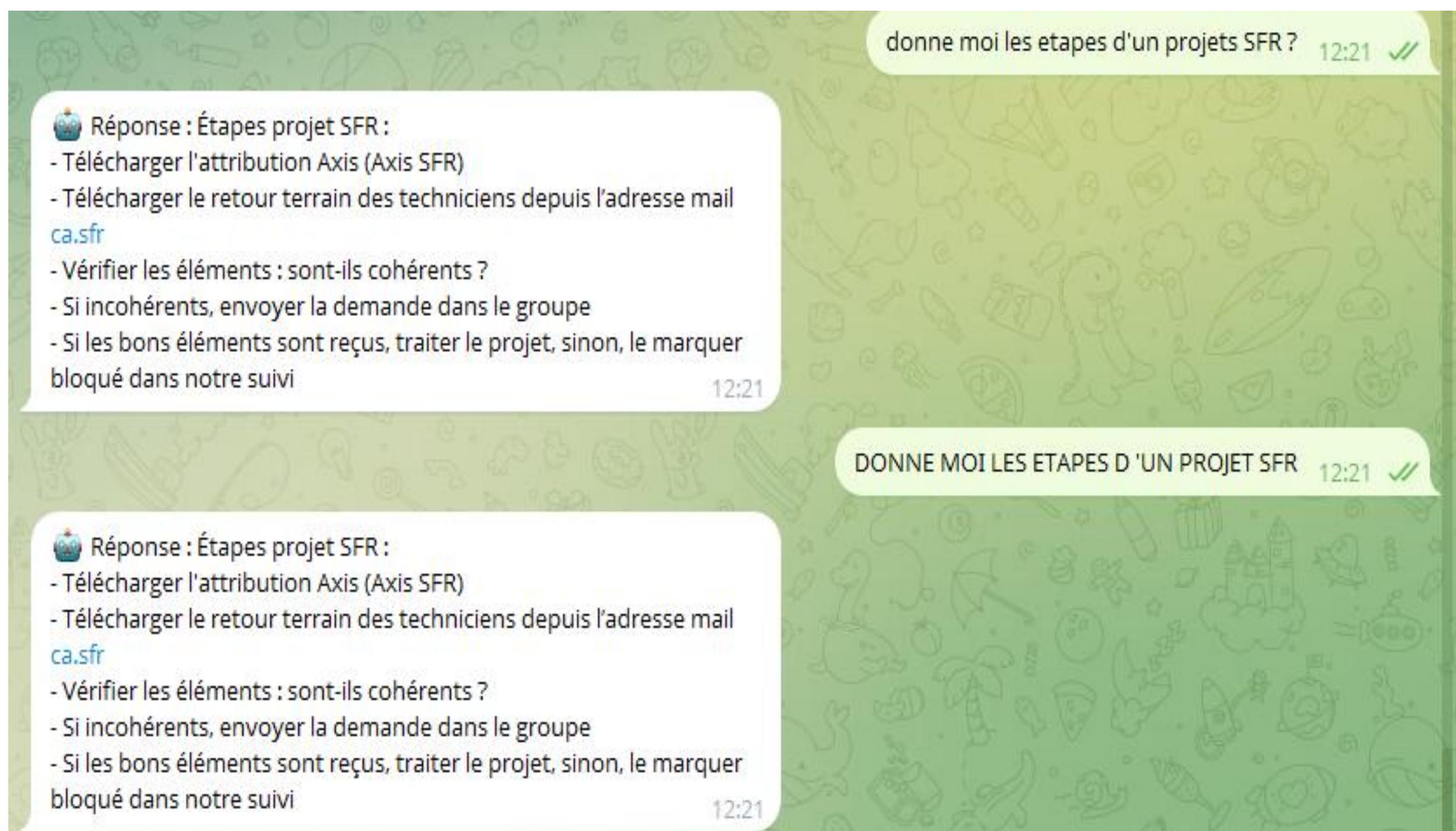
Commade projet -authentification

Cette capture illustre le mécanisme de contrôle d'accès intégré à la commande /projet. Lorsqu'un utilisateur saisit cette commande, le bot lui demande d'abord de fournir son nom afin de vérifier son identité. Dans cet exemple, le nom indiqué ne correspond pas à un utilisateur autorisé. Le système refuse alors l'accès et renvoie un message clair précisant que la consultation des projets est interdite pour cet utilisateur. Cette étape garantit la sécurité des données en limitant la visibilité des informations aux personnes habilitées



Capacité du modèle NLP à interpréter des questions reformulées

Cette capture met en évidence la capacité du modèle NLP à reconnaître et traiter des questions exprimées de manière différente. Bien que les requêtes aient été formulées sous diverses formes (« Quels sont les différents types de demandes ? », « Quels types de demandes existe-t-il ? », « Peux-tu me citer les types de demandes ? »), le bot a correctement identifié qu'elles relevaient de la même intention associée à la FAQ. Il a ainsi consulté les deux sources d'information prévues (procedure_sfr.json et faq_sfr_complet.json) pour fournir une réponse précise et uniforme. Cette performance illustre que le modèle comprend le sens global d'une question, même lorsque sa formulation varie.



Compréhension du texte indépendamment de la casse

Cette illustration démontre que le bot est capable de comprendre une requête indépendamment de l'usage des majuscules ou des minuscules. Dans les deux cas présentés, il a correctement identifié l'intention principale comme étant liée à la FAQ et non à une demande de suivi de projet.

Le modèle a su interpréter le sens de la question, malgré les variations dans la casse du texte, et a fourni la

même réponse pertinente en s'appuyant sur les informations présentes dans la base de connaissances dédiée.

B. Implémentation des réponses via similarité sémantique

Chaque requête est codée en utilisant SBERT comme vecteur sémantique. Une comparaison de similarité cosinus est effectuée par rapport aux vecteurs de la base de données FAQ et des fichiers de projet. La réponse la plus proche est ensuite sélectionnée

C. Utilisation d'une base FAQ enrichie (question-réponse)

Les fichiers JSON contiennent des paires question-réponse enrichies manuellement et servent de base de données commune pour les réponses automatisées aux questions courantes

4.4 Développement du système de FAQ

A. Encodage des questions FAQ en vecteurs (embeddings)

Les questions du fichier JSON sont ensuite encodées en vecteurs numériques à l'aide du modèle SBERT. Ces vecteurs sont stockés pour l'optimisation des performances.

B. Calcul de la similarité cosinus pour la recherche de la meilleure réponse

Lorsque l'utilisateur pose une question, celle-ci est vectorisée. Une comparaison en cosinus est effectuée entre ce vecteur et les vecteurs des questions fréquemment posées. L'algorithme renvoie la réponse dont la question a le score de similarité le plus élevé

C. Seuil de confiance et gestion des réponses non comprises

Un seuil de similarité (par exemple 0,75) est fixé pour une correspondance correcte. Si aucune question n'atteint ce seuil, un message par défaut indiquant une incompréhension (« Désolé, je n'ai pas compris votre question ») est renvoyé.

4.5 Interface utilisateur et interaction

A. Flux de conversation avec l'utilisateur

Le dialogue entre l'utilisateur et le robot suit un schéma simple et fluide. L'utilisateur envoie un message, le robot répond et maintient le contexte avec un identifiant de session (si nécessaire)

B. Gestion des salutations et des réponses automatiques

Des modèles simples de reconnaissance des salutations (bonjour, merci, au revoir) sont intégrés pour améliorer l'interaction naturelle avec l'utilisateur

C-Conclusion :

Le projet de fin d'études intitulé « Développement d'un chatbot SFR basé sur le Traitement du Langage Naturel (NLP) » s'inscrit dans la dynamique actuelle de digitalisation et d'automatisation des services. Dans un contexte où les entreprises cherchent à améliorer la relation client tout en optimisant leurs coûts opérationnels, les chatbots représentent une solution stratégique et innovante. L'objectif poursuivi dans ce travail était de concevoir et de développer un agent conversationnel intelligent, capable de dialoguer avec les utilisateurs en langage naturel, d'interpréter correctement leurs demandes et d'y apporter des réponses précises et adaptées.

La réalisation de ce projet a nécessité l'intégration de plusieurs dimensions techniques et théoriques. D'une part, le Traitement du Langage Naturel (NLP) a joué un rôle central, puisqu'il constitue la pierre angulaire qui permet au chatbot de comprendre et de traiter les messages formulés par les utilisateurs. Grâce à l'utilisation de **Python** et de ses bibliothèques spécialisées (telles que NLTK, spaCy, ou encore TensorFlow pour les aspects d'apprentissage), nous avons pu mettre en place des mécanismes performants d'analyse syntaxique et sémantique, ouvrant la voie à une interaction plus fluide et plus humaine.

D'autre part, l'intégration de ce système à une interface pratique et largement utilisée comme Telegram, **via son API**, a permis de proposer une solution accessible en temps réel, et répondant aux habitudes des utilisateurs modernes.

Les résultats obtenus montrent que le chatbot est capable de fournir une assistance rapide et pertinente, tout en garantissant une disponibilité continue, ce qui constitue un avantage considérable par rapport à un support humain limité par des horaires ou des contraintes de charge. Ainsi, ce projet contribue directement à :

- Améliorer l'expérience client grâce à une interaction simple, rapide et personnalisée
- Renforcer l'image de modernité et d'innovation de l'entreprise en mettant en avant des solutions basées sur l'intelligence artificielle.
- Optimiser les ressources internes en réduisant le temps consacré par les conseillers aux questions répétitives et basiques, leur permettant de se concentrer sur des tâches à plus forte valeur ajoutée.

L'importance de ce projet dépasse donc la simple mise en œuvre technique. Il met en lumière la capacité des nouvelles technologies à transformer les modes de communication entre entreprises et clients, et illustre le rôle central que jouent les systèmes intelligents dans les stratégies de fidélisation et d'efficacité opérationnelle.

Sur le plan académique et personnel, ce travail m'a permis de développer de manière approfondie mes compétences dans plusieurs domaines complémentaires : la programmation en Python, l'application des techniques de NLP, la gestion d'API et l'intégration d'outils dans un environnement réel. Cette expérience constitue une base solide pour de futures réalisations, notamment dans le champ du Big Data, de l'intelligence artificielle et des systèmes intelligents.

Enfin, il est important de souligner que ce projet, bien que fonctionnel et opérationnel, ouvre de nombreuses perspectives d'évolution. Parmi les pistes d'amélioration possibles, on peut citer :

- L'intégration de modèles d'apprentissage profond (Deep Learning) pour améliorer la compréhension du langage et la pertinence des réponses.
- L'ajout d'une reconnaissance vocale afin de permettre une interaction non seulement textuelle, mais aussi orale, ce qui rapprocherait d'avantage le chatbot d'une véritable assistance humaine.
- L'élargissement de la base de connaissances pour couvrir un spectre plus large de besoins et de scénarios d'utilisation.
- L'intégration multicanale, en connectant le chatbot à d'autres plateformes comme WhatsApp, Messenger ou encore un site web, afin d'accroître sa disponibilité.

En conclusion, ce projet a démontré qu'un chatbot intelligent basé sur le NLP et déployé sur une interface grand public comme Telegram constitue une solution efficace et innovante pour répondre aux besoins de communication et d'assistance des entreprises modernes. Il représente non seulement une réussite technique, mais également une contribution significative dans l'amélioration de la relation client et dans l'adoption des technologies d'intelligence artificielle au service des organisations.

