MEMOIRE DE PROJET

DE FIN D'ETUDES

PRESENTE POUR OBTENIR LE TITRE **:**

DIPLÔME NATIONAL DE LICENCE APPLIQUEE EN TECHNOLOGIES DE L'INFORMATIQUE (TI)
Parcours : Développement des Systèmes d'Information (DSI)

Chatbot de soutien psychologique

Réalisé par : Mhamdi Ahlem
&
Zairi Rihem

SOUTENU LE **29/09/2021**DEVANT LE JURY D'EXAMEN :

Mlle/Mme/Mr NOM Prénom	**Président**
Mlle/Mme/Mr NOM Prénom	**Rapporteur**
Mlle/Mme/Mr NOM Prénom	**Encadreur-ISET**
Mlle/Mme/Mr NOM Prénom	**Encadreur-Entreprise**

A.U. : 2020-2021

Dédicaces

Nous dédions ce modeste travail à nous même,

Et à Monsieur Ferid Helali qui nous beaucoup encouragé et soutenue.

A tous ceux qui nous connaissent et contribuent du près ou de loin à entamer ce travail et auxquels nous devons être reconnaissant.

Ahlem Mhamdi & Rihem Zairi

Remerciements

D'emblée nous tenons à remercier DIEU le tout puissant de nous avoir aidé à entamer et à terminer ce travail.

Nous devons être reconnaissante en remerciant infiniment nos encadrant universitaire ***Mr Mohamed Bjaoui*** *ainsi que nos encadrant de stage* ***Mr Ferid Helali*** *qui nous ont beaucoup aidé et soutenu pour que ce travail soit bien organisé en nous mettant sur la bonne voie sur tous les plans et que sans eux ce travail ne pourrait voir le jour.*

Tous nous remerciements les plus sincères à toute personne qui nous aidé de près ou de loin pour réaliser et présenter ce projet dans de bonnes conditions.

Sommaire

Liste des Tableaux

Liste des Figures

INTRODUCTION GENERALE

Les voitures autonomes, ou machines parlant et agissant comme des humains, ont toujours occupé une place privilégiée dans la science-fiction. Aujourd'hui, la réalité rattrape l'imaginaire.

Plusieurs applications que nous pouvons commander par la voix nous aident à trouver le chemin optimal pour éviter embouteillage, ou nous informer des meilleures offres d'un même produit dans toute une région. Ces applications sont le résultat des avancées réalisées dans le domaine de l'intelligence artificielle. Ces avancées ne se sont pas produites du jour au lendemain. Il y a d'abord eu la naissance du Big Data.

En fait, l'évolution d'Internet et de l'Internet des objets (IOT) a ouvert la voie à nombreuses activités commerciales et sociales qui ont ouvert la voie à l'ère du Big Data, à chaque moment, dans le monde, plus d'une centaine de millions d'emails sont envoyés. Chaque minute, les moteurs de recherche comme Google enregistrent des millions de requêtes différentes sur leurs moteurs de recherche, et sur les réseaux sociaux comme Facebook et Twitter autant de post et de réactions. Commerce électronique des transactions d'une valeur de 85000$ sont effectuées, cette tendance ne cesse d'augmenter. Chaque étape génère des données, utilisables par les entreprises, mais impossibles à réaliser manuellement.

La nécessité d'automatiser l'extraction de valeur à partir de l'énorme quantité de données stimulées recherche dans les domaines de l'apprentissage automatique et de l'apprentissage profond. Avec l'accomplissement de la loi de Moore, qui a déclaré que les progrès informatiques augmenteraient approximativement du double tous les deux années, il est devenu possible d'exploiter le flux de données du Big Data.

L 'IA est devenue le sur-ensemble pour plusieurs domaines, tels que la vision par ordinateur et Traitement du langage naturel. Le premier est préoccupé par la façon dont les machines peuvent

voir le monde comme le font les humains, ce dernier par la façon dont les machines comprennent la parole humaine par écrit ou en format parlé.

Dans ce projet, nous visons à développer une solution intelligente pour prendre les séances de thérapies qui s'appuient sur la puissance du Deep Learning afin d'offrir la meilleure recommandation dans une mode plus humaine. . Au final, la solution sera vue comme un agent conversationnel qui est capable de parler à une personne curieux et qui permet de connais son état psychologique et de prend les recommandations nécessaire pour être en bonne santé.

Feuille de route

Dans ce document, nous couvrirons l'ensemble du processus du projet.il est organisé comme suit:

Le chapitre 1 Contexte du projet donne un aperçu du contexte du projet. On commence par présentant l'entreprise d'accueil du stage, nous fournissons ensuite l'énoncé du problème, brièvement décrire la solution et discuter de la méthodologie de gestion de projet de choix ainsi comme technologies de choix. A la fin, nous donnons le calendrier du projet.

Le chapitre 2 Contexte couvre les concepts de base et les principes fondamentaux des sujets d'intérêt et les concepts et technologies de pointe. Nous commençons par explorer le concept de systèmes de recommandation, puis nous passons en revue les technologies d'apprentissage en profondeur et sémantiques, et terminons le chapitre par une revue du traitement du langage naturel. Au Enfin, nous présentons nos choix pour le modèle qui convient à nos exigences.

Le chapitre 3 Solution proposée décrit la solution proposée du chat bot et comment il travaux.

Le chapitre 4 Implémentation selon Scrum Framework décrit les étapes de tous les étapes pour créer la solution

Dans la conclusion, nous évaluons nos réalisations, discutons de la marge d'amélioration et travail futur.

Chapitre 1 : Cadre Général du Projet

INTRODUCTION

L'objectif de ce chapitre est de définir le contexte global du projet. Premièrement, nous présenter la société de stage Alfa Computer & Consulting, nous donnons l'énoncé du problème et comment le résoudre, y compris la méthodologie de gestion de projet que nous adoptons pour gérer le processus et présenter le calendrier du projet.

I. PRESENTATION DE LA SOCIETE

Dans le cadre de notre stage de perfectionnement, nous avons réalisé notre stage au sein de la société Alfa Computer & Consulting dans le but d'enrichir nos connaissances théoriques par l'aspect pratique.

Alfa Computers est une agence des services numériques spécialisé en solutions interactives, en création des sites internet, e-commerce, multimédia, design, développement et hébergement situé à l'avenue Bechir Sfar 5100 Mahdia, Tunisie. Alfa computers a débuté son activité l'année 2009.

II. CADRE DE PROJET

II.1 Prevue de concept

Dans cette section, nous décrivons le projet à long terme ainsi que le module que nous allons construire. Ensuite, nous formulons l'énoncé du problème et fixons les objectifs.

II.1.1 Emily : Présentation

Le projet à long terme est de construire un psychologue qui peut agir en tant que mentor ou conseiller d'ami virtuel dans de nombreuses situations, allant de l'assistance pour apporter des solutions aux problèmes émotionnels.

Le premier pas vers la construction d'Emily, est de construire un système de conversation qui sémantiquement comprend la signification de l'entrée de la personne curieux.

II.1.2 Énoncé du problème

Le succès des logiciels de conversation, également appelés systèmes de chat bot, dépend fortement de les règles de saisie et la qualité des réponses de la machine. Le but est d'avoir un système qui permet à l'utilisateur de formuler ses pensées, ses souhaits ou simplement de parler sans règles de formatage, tout en étant capable de détecter ce qu'il veut et de donner une réponse précise. En ordre pour capturer le sens caché d'une entrée utilisateur, les techniques de traitement du langage naturel être utilisé.

II.1.3 Objectif du projet

Notre projet consiste à créer un chat bot qui agit comme un psychologue pouvant fournir des conseils à des personnes. La pierre angulaire du chat bot est l'analyse sémantique de la saisie de la personne curieuse. Lorsque la personne curieuse tape un message, le système détecte ce qu'il veut et suggère la meilleure recommandation. Ceci doit être réalisé grâce à un traitement du langage naturel à la pointe de la technologie. La personne curieuse doit être à l'aise avec la communication avec le système, n'a pas besoin d'utiliser de règles pour exprimer ses idées, mais plutôt de parler, et obtient une réponse précise.

II.1.4 Étapes de réalisation du projet

Pour la mise en œuvre de la solution, nous ferons une étude approfondie sur les solutions existantes et sur les techniques les mieux appréciées afin d'atteindre notre objectif. Le choix de la technique se fera en créant un prototype avec plusieurs techniques, et la décision de choix se fera sur la technique qui aura livré les meilleurs résultats.

Les étapes peuvent être résumées sous forme de flux :

- Trouver la meilleure méthodologie pour mener le projet
- Étude comparative des solutions de pointe pour tirer parti des résultats pleinement atteints ;
- Construire des prototypes avec différentes techniques, y compris des approches de pointe pour voir ce qui donne le meilleur résultat dans la portée du projet ;

- Construire une ontologie pour le domaine d'application ;
- Construisez le prototype final avec les outils de votre choix

III. Processus de gestion de projet

Pour mener notre projet, nous comparons dans cette section trois processus de gestion de projet qui sont couramment appliqués aux projets d'analyse de données, et nous en adopterons un en fonction de son adéquation à nos besoins. Étant donné que ces processus sont itératifs, nous adapterons la méthodologie de choix à Scrum agile pour mieux nous adapter aux changements de décision.

III.1 Cadre Scrum

Scrum est une méthodologie agile itérative et incrémentale. Il repose sur un travail d'équipe et un suivi continu à travers des réunions quotidiennes et périodiques. De plus, Scrum permet d'organiser parfaitement le travail, notamment lorsqu'il s'agit de projets de développement. Cette se fait par la distinction entre les membres de l'équipe en attribuant à chacun un rôle et la répartition des tâches à réaliser en sprints.

Dans un contexte SCRUM, je vais devoir utiliser quelques termes propres à cette méthodologie. En voici les plus pertinent:

Table 1 Réunions SCRUM

Réunions	Description
Planification de sprint	Il s'agit d'une réunion tenue le premier jour du sprint et en lequel le Product Owner présente la liste des fonctionnalités à effectué au cours de ce sprint. Puis l'équipe et la mêlée maître choisir parmi ces fonctionnalités celles qu'ils s'engager à effectuer jusqu'à la fin du sprint par fixant la durée et la capacité nécessaire à chaque fonction nationalité.

Mêlé quotidienne	Le mêlé quotidienne est une réunion intermédiaire qui intervient en cours de sprint. Elle réunit les membres de l'équipe de développement. Objectif : faire le point sur les tâches réalisées la veille, et celle prévue pour la journée.
Rétrospective Sprint	La rétrospective de sprint intervient dans la foulée de chaque sprint. Les utilisateurs métier n'y sont généralement pas invités. Elle offre à l'équipe de développement un espace d'échange pour tirer les enseignements du sprint, planché sur des axes d'amélioration des processus et outils. C'est aussi l'occasion de revenir sur les relations entre les membres de l'équipe et les problèmes éventuellement rencontré.

Table 2 L'équipe d'un projet SCRUM

Rôle	Mission
SCRUM Master	Supervision de l'avancement du projet et des activités de l'équipe
Propriétaire du produit (Product Owner)	Présentation des caractéristiques et des fonctionnalités du produit à développer et approbation du produit à livrer
L'équipe de développeurs	Réalisation des user stories et élaboration des sprints

Table 3 Artéfact SCRUM

Artéfact	Description
Backlog du produit	La définition des besoins fonctionnels sous forme de (user story)
Backlog du Sprint	la liste des tâches à implémenter dans un sprint, classées par importances et état
Produit partiel	la liste des tâches à implémenter dans un sprint, classées par importances et état

III.2 KDD

La découverte des connaissances dans les bases de données (KDD) est le processus de recherche de connaissances dans un grand ensemble de données avec des méthodes d'exploration de données. Cette méthodologie est largement utilisée dans les domaines liés à l'IA comme l'apprentissage automatique. Les étapes de la méthodologie KDD sont :

- *Compréhension commerciale :* Couvre la compréhension du domaine d'application, l'exploration des réalisations connexes et l'identification des exigences et des besoins des utilisateurs.
- *Target Dataset :* Dans cette étape, nous effectuons une sélection des données sur lesquelles travailler. Comme un en règle générale, plus il y a de données, meilleurs sont les résultats.
- *Nettoyage des données :* après avoir sélectionné l'ensemble de données, nous devons le prétraiter et traiter anomalies, telles que des valeurs manquantes, du bruit ou de mauvaises valeurs
- *Transformation des données :* l'ensemble de données nettoyé doit être apporté dans le format adéquat avant de l'utiliser pour l'exploration de données. Cette étape implique des techniques telles que l'extraction de caractéristiques, ou réduction de dimension pour les tâches de visualisation.
- *Choix de la tâche :* dans cette phase, nous décidons du type de tâche que nous voulons réaliser, sous prise en compte du cas d'utilisation.

- *Algorithme :* Il existe des tâches qui peuvent être réalisées par plusieurs algorithmes. Un exemple est la tâche de classification. Par conséquent, KDD définit une étape où nous devons adopter le meilleur algorithme pour accomplir notre tâche.
- *Data Mining :* Nous appliquons l'algorithme que nous avons adopté à l'étape précédente et obtenons un modèle pour notre tâche, qu'il s'agisse de classification ou de regroupement.
- *Interprétation :* Nous examinons le résultat de l'application du modèle et vérifions si nous avons extrait la bonne information pour notre tâche.
- *Consolidation des connaissances :* La dernière étape consiste à utiliser les informations extraite Résoudre des problèmes.

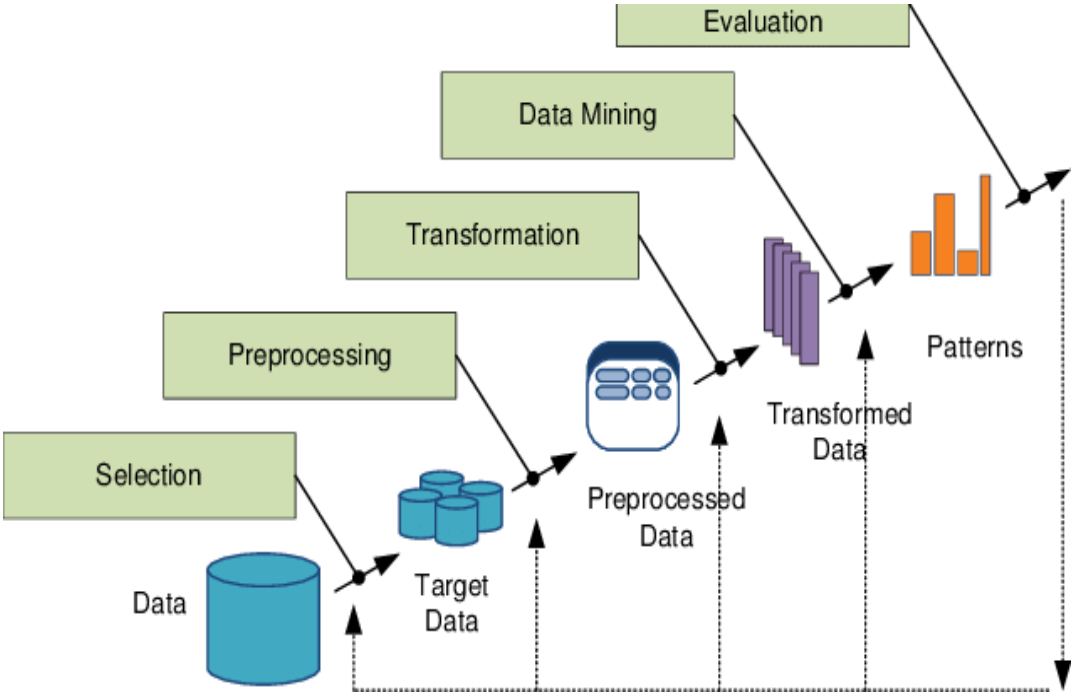

Figure 1 : Les étapes de la méthodologie KDD [1]

III.3 SEMMA

L'acronyme SEMMA (sample, explore, modify, model, assess) qui se traduit en français par : échantillonne, explore, modifie, modélise, évalue) se rapporte au noyau du processus de conduite de l'exploitation de données. [2]

Le processus se décompose en son propre ensemble d'étapes. Ceux-ci inclus:

- *Échantillon :* Cette étape consiste à choisir un sous-ensemble du jeu de données de volume approprié à partir d'un vaste jeu de données qui a été fourni pour la construction du modèle. L'objectif de cette étape initiale du processus est d'identifier les variables ou les facteurs (à la fois dépendants et indépendants) influençant le processus. Les informations collectées sont ensuite triées en catégories de préparation et de validation.
- *Explorer :* Au cours de cette étape, une analyse univariée et multivariée est menée afin d'étudier les relations interconnectées entre les éléments de données et d'identifier les lacunes dans les données. Alors que l'analyse multivariée étudie la relation entre les variables, l'analyse univariée examine chaque facteur individuellement pour comprendre sa part dans le schéma global. Tous les facteurs d'influence susceptibles d'influencer les résultats de l'étude sont analysés, en s'appuyant fortement sur la visualisation des données.
- *Modifier :* Dans cette étape, les enseignements tirés de la phase d'exploration à partir des données collectées dans la phase d'échantillonnage sont dérivés avec l'application du logique métier. En d'autres termes, les données sont analysées et nettoyées, puis transmises à l'étape de modélisation et explorées si les données nécessitent un raffinement et une transformation.
- *Modèle :* Une fois les variables affinées et les données nettoyées, l'étape de modélisation applique diverses techniques d'exploration de données afin de produire un modèle projeté de la façon dont ces données atteignent le résultat final souhaité du processus.
- *Accès :* Dans cette dernière étape SEMMA, le modèle est évalué pour son utilité et sa fiabilité pour le sujet étudié. Les données peuvent maintenant être testées et utilisées pour estimer l'efficacité de ses performances. [3]

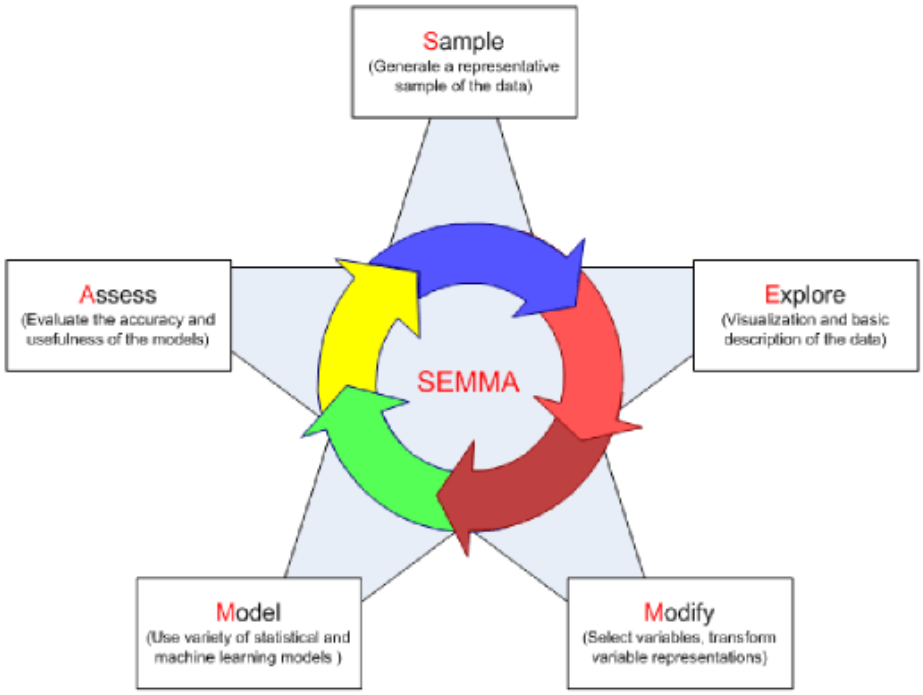

Figure 2 : les étapes de la méthodologie SEMMA [4]

III.4 CRISP-DM

CRISP-DM, qui signifie Cross-Industry Standard Process for Data Mining, est une méthode mise à l'épreuve sur le terrain permettant d'orienter vos travaux d'exploration de données. [5]

Les étapes de CRISP-DM sont :

• *Compréhension commerciale :* la première étape consiste à comprendre ce qu'il faut accomplir à partir d'un point de vue commercial. Cela signifie qu'une liste d'objectifs et de contraintes doit être correctement réglée. À travers cette étape, nous visons à saisir les facteurs qui peuvent avoir un impact sur le résultat du projet. Sans cette étape, cela signifie essayer de répondre aux mauvaises questions.

• *Compréhension des données :* La deuxième étape implique la collecte de données, l'analyse par l'outil de visualisation et d'intégration si les données ne sont pas dans le même format.

• *Préparation des données :* cette étape couvre le nettoyage des données et l'extraction des fonctionnalités pour la prochaine étape. Le nettoyage des données consiste à traiter les valeurs manquantes, etc.

• *Modélisation :* Dans cette phase, nous sélectionnons la technique de modélisation que nous appliquerons. Si plusieurs techniques sont nécessaires, cette étape doit être accomplie pour chacune.

• *Évaluation :* Dans cette étape, nous évaluons le modèle que nous avons créé à l'étape précédente en analyser des métriques telles que la précision de la prédiction, etc.

• *Déploiement :* Dans la dernière étape du processus CRISP-DM, nous prenons le modèle et utilisez-le une application du monde réel.

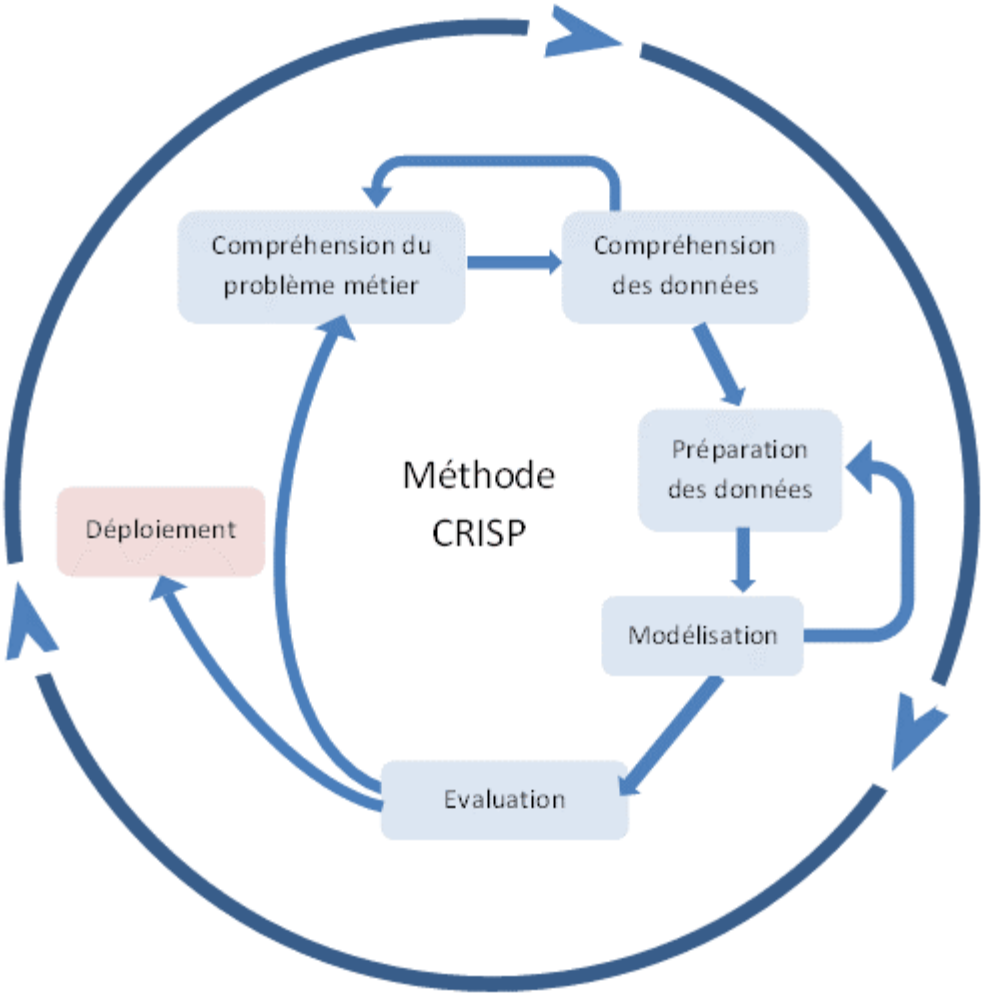

Figure 3 : les étapes de la méthodologie CRISP-DM

III.5 Comparaison et méthodologie de choix

On note une nette similitude entre les méthodologies KDD et CRISP-DM en termes de description des étapes. CRISP-DM a moins d'étapes que KDD, en fait KDD a trois étapes distinctes et bien étapes définies de la phase de modélisation, tandis que CRISP-DM définit abstraitement une étape. Dans ce sens, l'abstraction ne nous lie pas à un ordre ou à un processus particulier de construction de modèles. Dans ce point, CRISP-DM exprime simplement le processus souhaité mais avec moins de détails. SEMMA est défini par une société de logiciels pour guider les projets réalisés avec leur outil, n'est donc pas un standard, plutôt une méthodologie spécifique à un outil. Les normes offrent aux différents acteurs un moyen commun de gérer leurs projets. Les principaux avantages des normes sont l'interopérabilité qui conduit à efforts conjugués et innovation. Comme son nom l'indique, CRISP-DM est un standard ouvert norme [7]. Par conséquent, notre méthodologie de choix est la méthodologie CRISP-DM.

Table 4: Aperçu des méthodologies

KDD	SEMMA	CRISP-DM
Steps : 8	Steps : 5	Steps : 6
Business Domain Understanding		Business Understanding
Data Selection	Sample	Data Understanding
Data Pre-processing	Explore	
Data Transformation	Modify	Data Preparation
Data Mining Task	Model	Modeling
Choose Algorithm		
Apply Algorithm		
Interpretation / Evaluation	Assess	Evaluation
Pattern/Information Usage		Deployment

III.6 CRISP-DM Adapté en Scrum : Présentation

Pour mieux conduire l'exécution des six phases, nous appliquons CRISP-DM dans le Scrum agile, car nous sommes confrontés aux défis suivants : L'un des avantages de Scrum est qu'il s'adapte aux changements fréquents d'exigences des parties prenantes. Puisque nous sommes construire une preuve de concept, nous serons confrontés à des situations où nous devons progresser, tout en ayant à changer constamment de décisions si nous sommes confrontés à des processus de goulot d'étranglement. Afin d'assurer une progression constante du travail nous nous appuierons sur le time-boxing de Scrum, afin que nous puissions diviser les tâches en tâches encore plus petites et nous assurer d'avoir des artefacts à la fin de chaque sprint.

Étant donné que nous allons parcourir les six phases de CRISP-DM, pour effectuer des tâches de réglage fin et ajustements, il est nécessaire de suivre les progrès en adoptant l'aspect incrémental de Scrum. Nous montrons comment planifier les étapes CRISP-DM en tant que sprints dans le tableau suivant :

Table 5: Mappage des étapes CRISP-DM en tant que sprint Scrum

CRISP-DM Phases	Sprint
Business understanding	Sprint 0
Data understanding	Sprint 1
Data preparation	Sprint 2
Modeling and Evaluation	Sprint 3
Deployment	Sprint 4

CONCLUSION

Dans ce chapitre, nous avons donné un aperçu du projet. Nous avons présenté l'hôte de stage entreprise, puis nous avons donné une brève description du projet énoncé le problème et proposé la solution que nous adopterons pour le résoudre.

Chapitre 2 : Planification Et Architecture

INTRODUCTION

Dans ce chapitre, nous allons fournir les concepts de base des sujets d'intérêt. nous allons d'abord explorer les principes fondamentaux de l'apprentissage en profondeur et comment ils sont inspirés par le cerveau humain. Nous couvrirons brièvement les architectures courantes et nous concentrerons sur les réseaux de neurones récurrents. Enfin, nous discutons d'un des domaines d'application de l'IA qui est pertinent pour notre projet, à savoir le traitement du langage naturel.

I. Traitement du language naturel

Après avoir donné un aperçu de l'apprentissage machine et en profondeur, nous discutons d'un domaine de l'IA où l'apprentissage machine et l'apprentissage en profondeur sont appliqués. Dans cette section, nous présentons les concepts clés et les algorithmes de la PNL.

I.1 Définition

Le traitement du langage naturel (NLP) est la branche de l'IA pour le développement de systèmes informatiques qui comprennent et interprètent le langage humain, parlé ou écrit, à l'aide d'algorithmes d'apprentissage automatique. La PNL est appliquée dans une variété de tâches telles que la reconnaissance vocale, les traductions linguistiques, la synthèse de texte, les systèmes de questions-réponses, la génération de texte ou de parole et les moteurs de recherche.

I.2 Algorithmes et techniques de la PNL

Tokenisation Un jeton est un élément d'une donnée séquentielle. la tokenisation consiste à diviser une longue séquence, comme du texte ou de la parole, en unités plus petites basées sur un séparateur. Dans le cas du langage naturel, les mots tels que les noms et les verbes sont des jetons séparés par un espace.

La lemmatisation et le radicalisme sont deux techniques permettant de trouver la racine des mots, car le langage naturel est construit avec des mots dérivés les uns des autres. La différence réside dans la manière dont chaque technique opère sur les mots. La radicalisation supprime les

préfixes et les suffixes qui peuvent produire des mots inexistants, tandis que la lemmatisation trouve vraiment le mot de base.

Entité-Nom-Reconnaissance identifie des noms ou des entités (personnes ou organisations) et les associe dans leurs classes ou catégories, comme les noms de célébrités, de grandes entreprises, de lieux célèbres, de devises, etc. Cette technique est utilisée dans les moteurs de recherche.

Le terme Fréquence-Inverse Document Fréquence (tf-idf) est appliqué lorsque nous essayons de trouver un document correspondant à un mot-clé de recherche. La correspondance est basée sur la fréquence à laquelle le terme dans le texte de recherche apparaît dans un document.

Allocation Dirichlet Latente fait référence à la tâche d'affecter un sujet à un document. Étant donné un document comportant plusieurs mots, nous essayons de saisir le sujet (caché) qu'il décrit, d'où les mots latent (caché) et allocation.

Incorporations de mots est une technique qui transforme les mots en caractères alphabétiques en vecteurs de nombres réels d'une manière qui préserve les relations sémantiques entre les mots. Il s'agit d'un aspect important de la préparation des données, car les algorithmes d'apprentissage en profondeur ne peuvent traiter que des valeurs numériques. Un autre avantage est l'aspect relation sémantique qui peut être préservé, afin de relever les défis de la PNL, dont nous discutons dans la section suivante.

I.3 Les défis de la PNL

Dans ce qui suit, nous énumérons les raisons de la complexité de la PNL. Ces propriétés reflètent la nature du langage naturel.

• Acronymes : une manière d'exprimer des noms d'entités composées de plusieurs mots, en prenant le caractère initial de chaque mot, et en les combinant en un seul mot.

• Ambiguïté lexicale : c'est quand plusieurs mots avec la même orthographe mais un sens différent sont mis dans une longue phrase.

• Ambiguïté syntaxique : c'est lorsqu'un texte grammaticalement écrit n'est pas conforme au bon sens comme je parle à mon poisson mort.

• Mots-Composés : sont une combinaison de plusieurs mots pour décrire une seule chose.

II. Incorporations de mots : modèles sensibles au contexte et modèles non sensibles

Dans cette section, nous explorerons les principaux modèles de représentation des mots sous forme de vecteurs. Les modèles discutés ici appartiennent à deux catégories : les modèles non sensibles au contexte et les modèles sensibles au contexte.

• Les modèles non sensibles au contexte représentent des mots avec des vecteurs basés sur un contexte unique ;

• Les modèles contextuels examinent plusieurs contextes pour coder un mot dans un vecteur.

II.1 Word2Vec et Doc2Vec

est un algorithme qui apprend à représenter les mots sous forme de vecteurs en prenant chaque mot du corpus d'entrée, et essaie de prédire un mot à partir du contexte. Le contexte peut être représenté par des phrases avec des mots manquants, et le modèle non supervisé est entraîné pour prédire les blancs. Word2Vec est très bon pour la recherche d'analogies. Doc2Vec est une extension de Word2Vec , qui représente un document entier (phrase ou paragraphe) sous forme de vecteur.

II.2 Gant

Signifie Global Vectors est un modèle non supervisé pour représenter les mots sous forme de vecteurs qui permet de calculer les distances entre les mots. Avec GloVe est utilisé pour la correspondance de synonymes. La différence avec le modèle Word2Vec, c'est que GloVe utilise le comptage de mots et la cooccurrence dans le corpus entier.

II.3 Le modèle ELMo

ELMo signifie Embeddings from Language Models a été conçu pour représenter chaque mot tout en considérant les mots précédents et suivants. Le modèle fonctionne au niveau des caractères, ce qui signifie qu'il peut casser un seul mot pour détecter sa signification. De cette façon, le modèle peut comprendre que les mots Grand et Grandeur sont liés d'une manière ou d'une autre.

II.4 Le modèle ULM-FiT

(ULM-FiT) ULM-FiT signifie Universal Language Model Finetuning for Text Classification est une approche qui consiste à diviser la tâche en deux étapes distinctes : la première étape appelée pré-formation consiste à créer un modèle générique et la seconde étape est appelée étape de réglage fin.

II.5 Modèles de transformateurs

Sont des modèles qui peuvent prendre une entrée non pas de manière séquentielle uniquement des LSTM, mais simultanément, ce qui en fait une amélioration par rapport à celle-ci, spécialement lors de son entraînement. Les mots circulent indépendamment à travers les couches de transformateurs, ce qui a permis que le traitement se déroule de manière parallèle. Avec cette amélioration, la quantité de données utilisée pour construire une représentation forte peut être construite.

II.6 Encastrements à la pointe de la technologie : le modèle BERT

Signifie Représentations d'encodeur bidirectionnel à partir de transformateurs Représentations d'encodeur bidirectionnel à partir de transformateurs est une approche, qui prend le meilleur des modèles décrits précédemment, qui a été appliquée sur de grands corpus pour produire un modèle d'incorporation de mots fortement contextualisé, à savoir le Modèle de représentation du langage BERT. Il existe également plusieurs variantes de modèles qui sont basées sur la même approche

mais diffèrent par taille, ou ont été modifiés afin d'obtenir un comportement différent tel que la vitesse et la précision sur des tâches spécifiques. DisstillBert est un modèle de variante BERT plus petit, d'où la désignation « BERT distillé », qui est un modèle de pré-formation à utiliser dans le cadre de restrictions de performances de calcul. Il a été démontré qu'il fonctionnait plus rapidement tout en affichant de bonnes performances. RoBert signifie Approche BERT optimisée de manière robuste, est un recyclage de BERT avec plus de données et de puissance de calcul.

III. Apprentissage automatique

III.1 Définition de l'apprentissage automatique

L'apprentissage automatique (ML) est un sous-domaine de l'intelligence artificielle qui concerne les algorithmes qui permettent aux systèmes informatiques d'apprendre automatiquement des règles et des modèles à partir de données [11]. L'application d'algorithmes d'apprentissage aboutit à des modèles d'apprentissage.

III.2 Modèles d'apprentissage

Il existe quatre grandes classes de modèles d'apprentissage :

L'apprentissage supervisé consiste à appliquer une technique d'apprentissage automatique sur des données étiquetées et à s'attendre à ce que l'ordinateur apprenne le mappage entre les caractéristiques des données et l'étiquette, c'est-à-dire la réponse finale. Ce principe est inspiré du comportement d'apprentissage par l'exemple de l'être humain.

L'apprentissage non supervisé, c'est lorsqu'un programme est alimenté avec des données non étiquetées. Au lieu d'apprendre ce que sont les échantillons, il peut apprendre les modèles et les tendances qui apparaissent dans les données [11].

L'apprentissage semi-supervisé se situe entre l'apprentissage supervisé et non supervisé. L'ensemble de données contient des échantillons étiquetés et non étiquetés.

L'apprentissage auto-supervisé (SSL) est un terme plus général pour décrire le même concept mais dans des cas d'utilisation différents tels que la robotique ou l'apprentissage par renforcement. L'idée de base est que les étiquettes ne sont pas fournies à la main, mais plutôt extraites.

L'apprentissage par transfert est l'approche consistant à utiliser une solution qui a été appliquée à un problème différent pour résoudre un problème actuel.

III.3 Learning Process

Après avoir discuté des modèles d'apprentissage, nous passons en revue les étapes de la façon dont supervisé et semi-supervisé

Processus supervisé L'algorithme commence par faire une estimation aléatoire des paramètres pouvant être appris et les utilise dans la fonction de mappage. Une sortie est ensuite calculée. Et utilisé comme paramètre dans une fonction de perte. La fonction de perte qui est paramétrée avec le sortie prévue, indique à quel point la prévision était. Puisque nous essayons de nous rapprocher de l'observation réelle des mots, ce qui signifie réduire la différence entre prédiction et observation, l'algorithme tente de résoudre un problème d'optimisation où l'objectif est de minimiser la fonction de perte. Le passage à la solution optimale (minimum) de la fonction de perte est réalisé par des optimiseurs, l'ajustement du poids est effectué par l'algorithme de rétro-propagation.

Processus semi-supervisé Lorsque les étiquettes sont rares, le semi-supervisé est utilisé. La méthode courante pour résoudre ce problème est l'augmentation des données. Pour la PNL, la quantité de texte non étiqueté dépasse les données étiquetées. Dans ce cas, l'augmentation des données consiste à prélever un échantillon étiqueté et à essayer d'en générer des versions reformulées.

Processus non supervisé Les modèles non supervisés consistent à identifier des tendances ou des tendances répétées dans un grand ensemble de données non étiquetées. Un modèle peut être compris comme un sous-ensemble d'éléments similaires, où la similitude est calculée en appliquant une mesure de similitude.

III.4 Tâches d'apprentissage automatique

Dans le tableau 6, nous décrivons brièvement les tâches courantes d'apprentissage automatique et fournissons des algorithmes pour eux.

Table 6: Présentation des tâches d'apprentissage automatique

Tâche	Description
Classification	La classification est une technique d'apprentissage supervisé dans laquelle la machine peut apprendre comment affecter des observations à une classe définie ou une catégorie définie.
Régression	La régression est une famille de techniques statistiques permettant de prédire une relation entre un ensemble de variables dépendantes et une ou plusieurs variables indépendantes (également appelées caractéristiques).
Regroupement	Le cluster est la tâche de construire des groupes d'objets de telle sorte que des objets identiques relèvent du même groupe sur la base d'une mesure de similarité.
Augmentation des données	L'augmentation des données est une approche qui crée de nouveaux échantillons de données à partir d'un ensemble de données existant pour enrichir et économiser l'effort de collecter plus de données.

IV. L'apprentissage en profondeur

IV.1 Définition

L'apprentissage profond (DL) est une approche d'apprentissage automatique où les algorithmes d'apprentissage sont basés sur des réseaux de neurones artificiels (ANN), qui sont inspirés du cerveau humain. La différence entre ML et DL réside dans l'extraction de caractéristiques. En ML, nous définissons explicitement les caractéristiques et laissons le modèle apprendre une

correspondance avec les étiquettes, tandis qu'en apprentissage en profondeur, la seule interaction humaine consiste à fournir des données, laisser l'algorithme extraire la caractéristique et apprendre la correspondance avec la sortie. Le besoin de DL est né à l'ère du Big Data où les données ne peuvent plus être traitées de manière conventionnelle.

IV.2 Les réseaux de neurones

RN est appliquée avec des modèles dont l'architecture s'inspire du cerveau humain. En figure 4, figure 5et figure 6, nous voyons respectivement un neurone biologique, un seul neurone artificiel, que nous appelons unité, et un réseau d'unités artificielles. Une unité se comporte comme un neurone biologique .L'appareil reçoit des données xi (dendrites), effectue ensuite une somme pondérée dessus (corps cellulaire), transmet le résultat par une fonction d'activation. La valeur calculée est l'activation de l'unité est donnée par la formule

$$z = \sum_{i=0}^{n} x_i * w_i + b_i$$

où xi sont l'entrée, wi la valeur de poids respective du lien de connexion, et bi est le terme de biais. Figure 4 montre un neurone biologique, figure 5 montre la représentation d'un neurone avec les xi wi et bi, la fonction d'activation prend la somme pondérée et calcule la valeur de sortie unitaire comme

$$a = \sigma(z) = \sigma(xi * wi + bi)$$

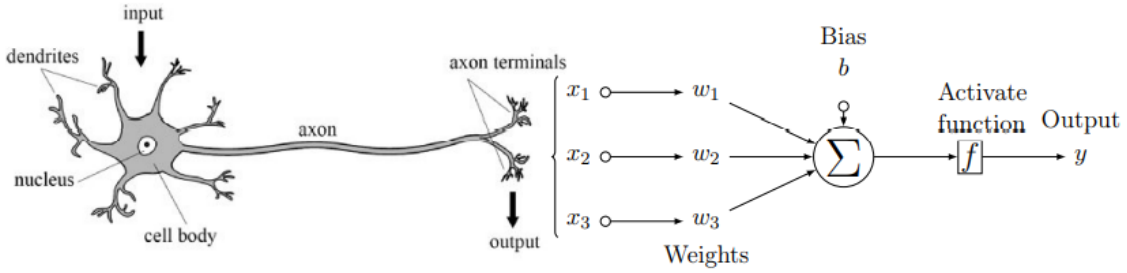

Figure 4: Single Biological Neuron [176] **Figure 5: Graphical Representation of an Artificial Neuron**

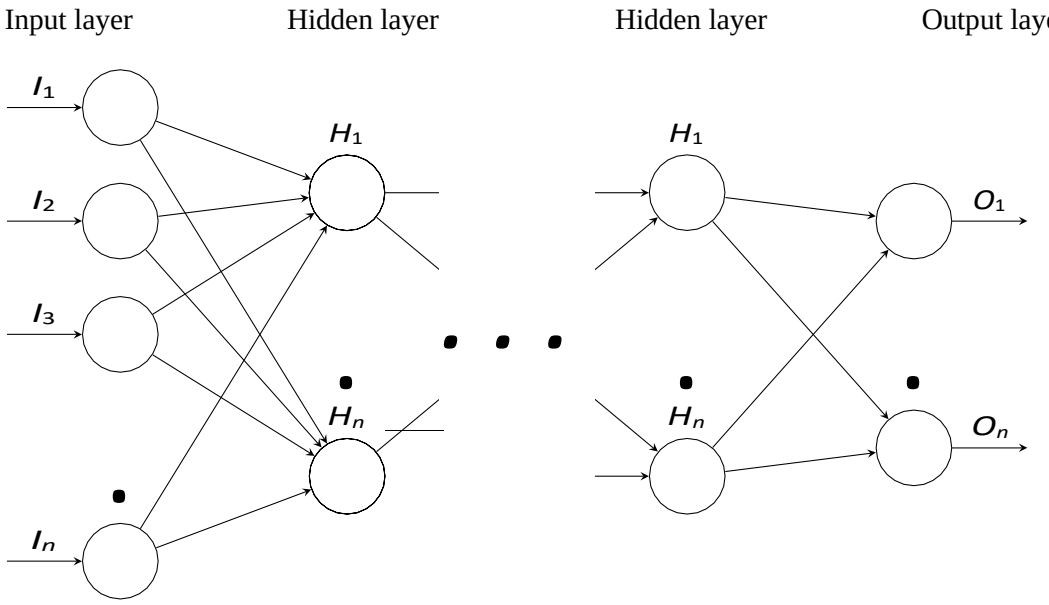

Figure 6 : réseaux des neurones artificielles

IV.3 Types de réseaux de neurone

Table 7: Les Types de reseaux de neurones

Types de réseau	Description et Architecture associée
Réseau de neurones d'anticipation(FNN)	L'information circule dans un seul sens. Les exemples sont les réseaux de neurones profonds traditionnels, le Perceptron multicouche, les réseaux de neurones convolutifs, les réseaux résiduels, les auto-encodeurs, Réseaux accusatoires[12] [13]
Réseauxde neurones récurrents (RNN)	En raison de la récurrence, utilisé pour le traitement de données séquentielles telles que les vidéos, le traitement du langage naturel (NLP) et la parole [14] [12]
Réseau de fonctions à base radiale	Utilisée pour la classification et la prédiction de séries chronologiques, sa couche cachée est construite en utilisant le clustering, avec le centroïde et sa zone environnante, d'où la fonction radiale. [12]
Réseau de neurones modulaire	Décompose les grands réseaux, chaque petit module, effectue indépendamment, les sorties respectives sont fusionnées [12]

IV.4 Paramètres

Poids et biais Les poids et les biais sont les paramètres pouvant être appris. En commençant par la première couche cachée, chaque unité est chargée de détecter une caractéristique particulière. Dans cette étape, le poids est la réponse à la question : Avec quelle valeur l'entrée doit-elle être multipliée par, pour que l'unité capture la caractéristique particulière dont elle est responsable ? Après avoir multiplié l'entrée avec le poids, la question suivante vient, qui est l'imitation du concept de neurones actifs dans le cerveau humain : quelle valeur de l'unité actuelle peut être transmise à la couche suivante, c'est-à-dire active l'unité actuelle ?. La réponse à cette question est la valeur de biais. Un biais de k signifie que l'unité courante ne peut être active que si sa valeur est supérieure à k. Après avoir décalé la valeur par le biais, la valeur du neurone est transmise à une fonction d'activation qui normalise la valeur. Les fonctions d'activation communes sont données dans ??.

IV.5 Problèmes courants

Problèmes d'apprentissage Deux effets secondaires peuvent apparaître pendant la phase d'entraînement, à savoir le sur apprentissage et le sous-apprentissage. Le sur ajustement signifie que le modèle est sur spécialisé dans cet ensemble de données, c'est-à-dire qu'il a appris la tendance ainsi que le bruit, et qu'il est incapable de généraliser, c'est-à-dire de fonctionner sur des données invisibles. Le sous-apprentissage a l'effet inverse sur les données d'entraînement et est incapable d'identifier une fonction de mappage entre les données d'entraînement sous-jacentes. **Complexité de calcul** Dans un réseau de neurones Deep Learning, le nombre d'opérations pour apprendre les paramètres peut être coûteux en fonction du nombre de paramètres. Le nombre de paramètres est le nombre de connexions entre couches ajoutées au nombre de nombres de biais. Avec de nombreuses couches, le nombre de paramètres augmente, de même que la complexité de l'algorithme de rétro propagation

V. Roseau de neurons recurrent

V.1 Représentation des RNN

Le principal avantage des RNN par rapport aux autres types est qu'ils sont adaptés pour traiter des données séquentielles telles que du texte, de la parole ou des vidéos. Les CNN ou les NN standard n'ont besoin d'aucune dépendance temporelle pour s'exécuter, contrairement aux données séquentielles. Idem pour la parole. Pendant que nous parlons, nous comprenons les idées actuelles basées sur ce que nous avons appris ou entendu dans le passé. La représentation graphique est montrée en figure 7. Il semble que plusieurs NN soient enchaînés pour partager des connaissances.

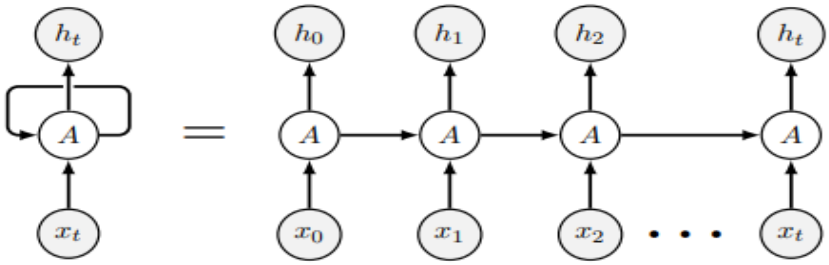

Figure 7: Diagramme de réseaux du neurones récurents

V.2 Réseaux de neurones récurrents : à la pointe de la technologie

L'architecture montrée en figure 7 s'est avérée souffrir d'un problème qui survient lorsqu'on traite de très longues séquences. Lors de l'entraînement sur un texte très long, à un moment donné, l'activation de la nouvelle entrée écrase l'activation de l'activation précédente, ce qui les fait oublier lors de l'application de l'algorithme de rétro-propagation. Le phénomène inverse peut également se produire. Dans les deux cas, lors du calcul du gradient, nous nous retrouvons avec un très petit ou un très grand gradient. Les mémoires à long terme (LSTM) ont été proposées pour surmonter ces problèmes en ajoutant une unité de mémoire pour garder une trace des états qui traversent le réseau. l'état de l'unité mémoire dépend de trois composants, la porte d'oubli, la porte d'entrée et la porte de sortie. La porte d'oubli indique si la mémoire précédente est importante pour le calcul de l'état actuel de la mémoire ou si elle doit être oubliée (supprimée). La porte de sortie calcule ensuite l'état de sortie pour délivrer l'unité suivante, en combinant les

retenues précédentes mémoire et entrée de courant. En figure 7, nous décrivons les utilisations courantes des architectures Seq2Seq

V.3 Architecture séquence à séquence

Seq2Seq est une famille d'architecture qui consiste à relier deux RNN ensemble pour résoudre certaines tâches. Une architecture courante est l'architecture de décodeur d'encodeur.

Table 8: Seq2Seq Architectures

Type	**Description et Application**
Un par un	une entrée (mot), une sortie (mot), traitement d'un seul mot à un instant qui est le comportement des NN standard
Un à plusieurs	L'entrée est un seul mot, la sortie est plus qu'un mot. Utilisé pour générer du texte à partir d'un seul mot-clé. Comme les applications de génération de poésie [15]
Plusieurs à un	L'entrée est une séquence, la sortie un mot. L'analyse des sentiments est couramment utilisée
Plusieurs à plusieurs (de même taille mettre/sortir)	L'entrée et la sortie ont la même taille, utilisées dans la reconnaissance d'entité nommée lorsque la sortie générée est de même longueur mais avec des parties en surbrillance
Plusieurs à plusieurs (entrée/sortie longueur différente)	Entrée et sortie de taille inégale, utilisées dans la traduction linguistique

CONCLUSION

Dans ce chapitre, nous avons exploré les principes fondamentaux de l'apprentissage en profondeur et comment ils sont inspirés par le cerveau humain. Nous avons brièvement présenté les architectures communes et nous nous sommes concentrés sur les réseaux de neurones récurrents. Enfin passé en revue certaines techniques de traitement du langage naturel.

Chapitre 3: Solution Proposée

INTRODUCTION

Dans ce chapitre, nous verrons comment fonctionne le mécanisme de détection d'intention avec le séquentiel model. Tout d'abord, nous présentons deux concepts sur lesquels séquentiel s'appuie, à savoir le concept d'insertion de mots et le modèle de transformateur. Ensuite, nous décrivons comment BERT est construit et comment il peut être affiné pour effectuer différentes tâches.

I. Presentation de l'application

Dans cette section, nous décrivons la grande image de la solution proposée. Nous montrons d'abord les composants du chat bot, puis nous illustrons le flux de travail interne.

I.1 Architecture globale

L'application se compose d'un modèle de traitement du langage naturel qui est exposé comme un service Web qui récupère le texte de l'utilisateur via un client Web et renvoie une réponse au client après avoir effectué sur l'entrée comme le montre la figure suivante.

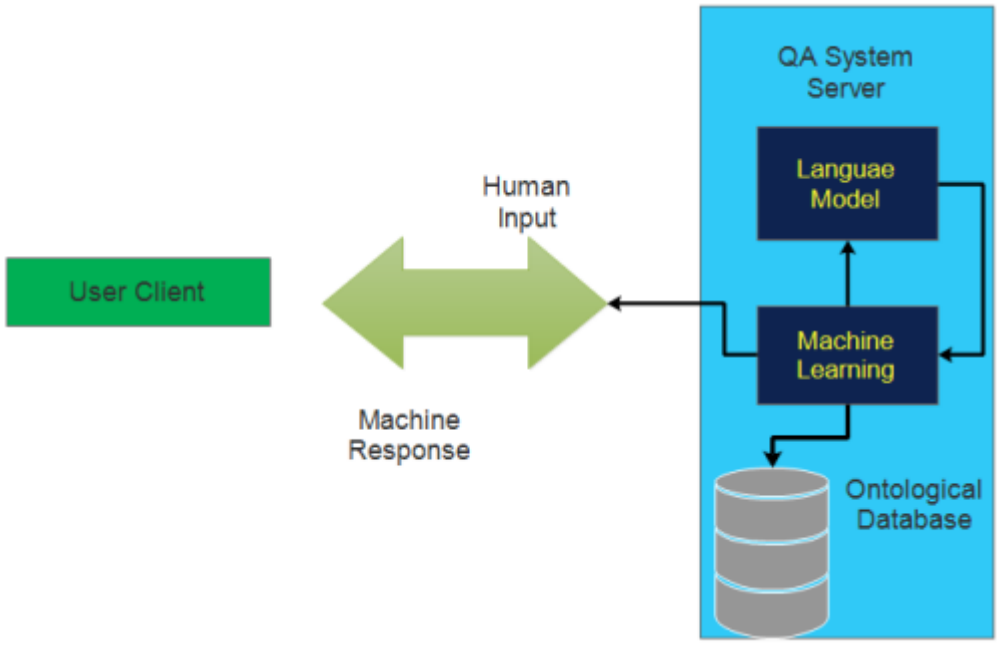

Figure 8: Architecture Globale

II. Représentation de mots dans Chat bot

Dans cette section, nous introduisons le concept de plongement de mots qui est le cœur concept qui apportera de l'intelligence au chat bot.

II.1 Matrice d'insertion

En PNL, les mots sont représentés comme des vecteurs de caractéristiques. Une caractéristique peut être un sens reliant deux mots. Les deux mots Chats et Chiens peuvent avoir des caractéristiques similaires, si l'on considère des caractéristiques telles que, est animal, âge, couleur, animaux vertébrés. Il y a plusieurs caractéristiques ou points communs qui peuvent apparaître entre les mots. Comme indiqué, les caractéristiques sont des vecteurs de nombres car essayez d'apprendre la distance entre les mots à l'aide de réseaux de neurones d'apprentissage en profondeur. Au final, nous représentons tous les mots d'un vocabulaire donné comme vecteurs, on obtient une matrice 3.1. Une matrice de plongements peut être considérée comme une table de recherche de mots du vocabulaire d'entrée. Pour retourner le vecteur de un mot particulier, on

utilise la colonne id du dictionnaire illustré en table 9, pour trouver la position du vecteur dans la matrice de plongements. Le mot incrustations fait référence au fait que les mots sont intégrés ou enfermés dans un espace avec un nombre fini de caractéristiques. 3.1 montre un enrobage avec dimensions D (nombre de caractéristiques) et mots V (nombre total de mots du vocabulaire).

Table 9 : Représentation d'une matrice d'inclusion

	Feature1	Feature2	···	Feature$_D$
Word1	.235	.00123	.036	.789
Word2	.2687	.26	.365	.6548
···	···	···	···	···
Word$_N$	.3265	.2654	.4123	.574162

Table 10: Dictionnaire

Word	**Indexes**
Word1	id1
Word2	id2
···	id$_k$
Word$_N$	id$_N$

II.2 Apprentissage des fonctionnalités

Pour mieux comprendre comment sont calculées les valeurs des vecteurs de mots, nous prenons un exemple simplifié dans un espace vectoriel à deux dimensions comme indiqué en figure 9. La première étape comme pour toute tâche de PNL, est de fournir un corpus. On prend les deux phrases suivantes : « ce sont des femmes intelligentes» et « c'est une reine intelligente». Lorsque le traitement du corpus atteint les mots femmes et reine, il voit que ces mots sont utilisés dans un contexte très similaire. Le terme contexte désigne les mots entourant le mot actuel que l'algorithme examine.

En langage naturel, un contexte peut être comparé à des phrases ou des paragraphes. Basé sur dans un contexte similaire, les algorithmes apprennent à mapper femmes et reine sur deux vecteurs qui sont géométriquement proches les uns des autres. Des mots qui ne sont pas vus ensemble dans le corps sont séparés par une plus grande distance comme indiqué en figure9. Nous voyons < reine > et <femmes> sont plus proches l'un de l'autre, tandis que <femmes> et <homme> ne le sont pas. On dit alors que la distance reflète la similitude. L'avantage que nous avons lorsque nous transformons des mots en vecteurs est que nous pouvons calculer la distance en utilisant la métrique de distance.

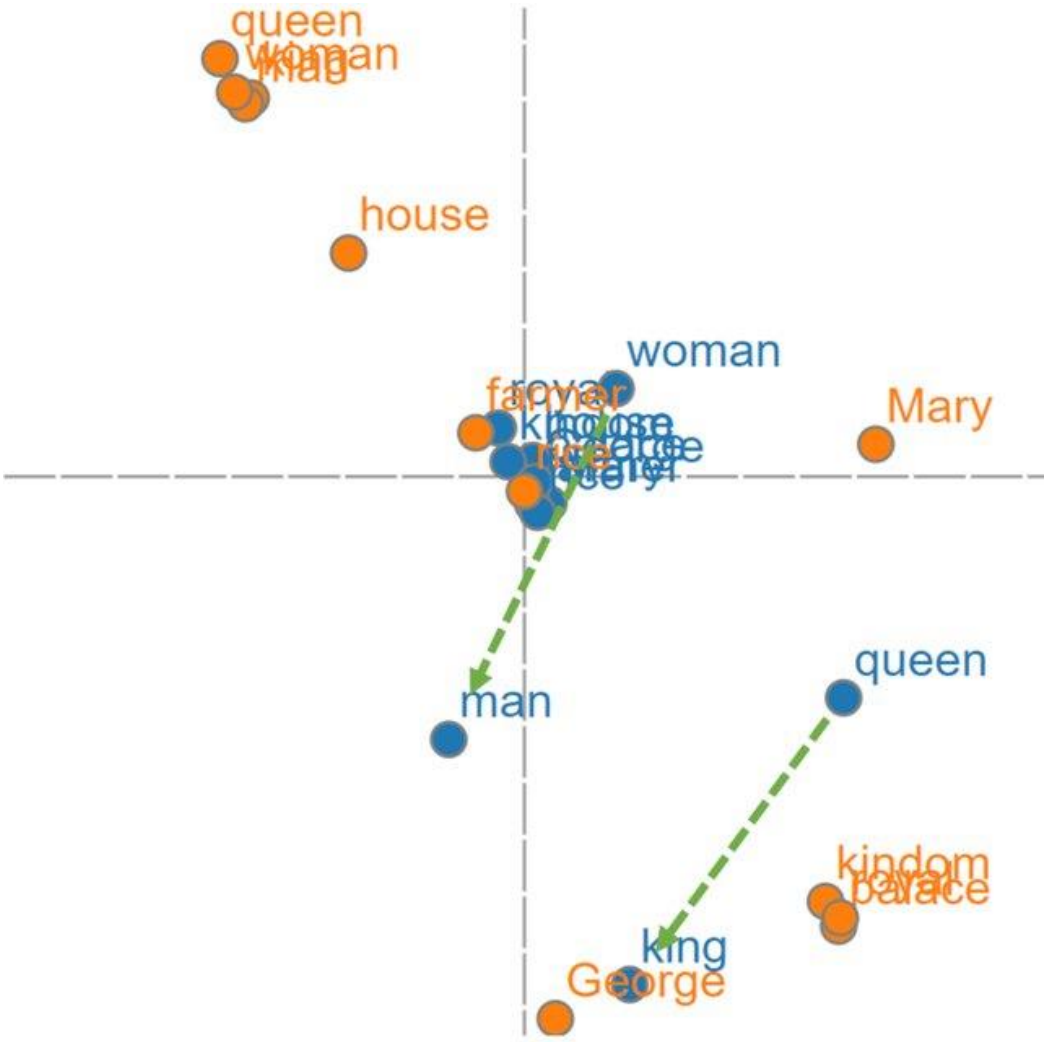

Figure 9:Espace vectoriel dimension pour représenter mots [8]

III. Chatbot avec modele séquentiel

Dans la section précédente, nous avons donné un aperçu de la façon dont les séquences des mots peuvent être représentées comme vecteurs de nombres afin que nous puissions les utiliser dans une tâche d'apprentissage en profondeur. Dans cette rubrique nous décrire le fonctionnement interne du modèle séquentiel.

III.1 Architecture

Un Séquentiel modèle est approprié pour une pile simple de couches (stack of layers) où chaque couche a exactement un tenseur d'entrée (input) et un tenseur de sortie (output) comme on peut le voir en figure 10.

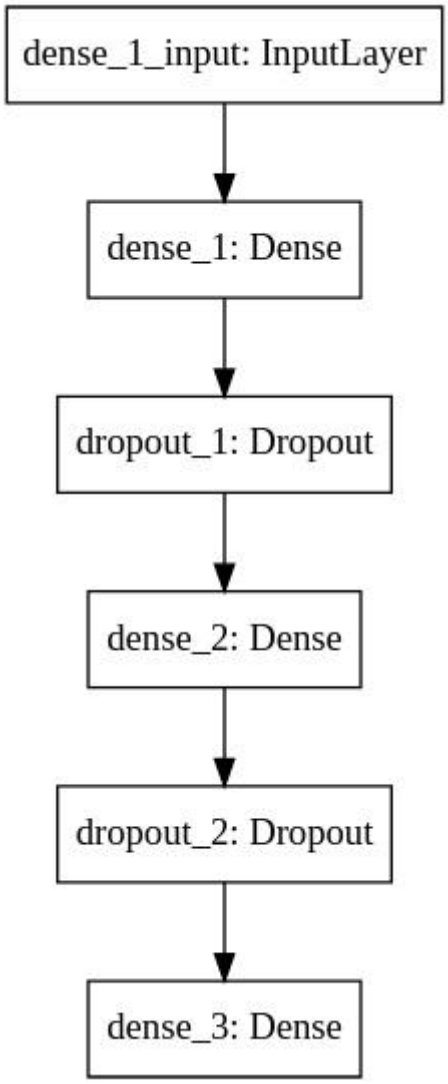

Figure 10 :un modèle séquentiel de réseau de neurones Keras [9]

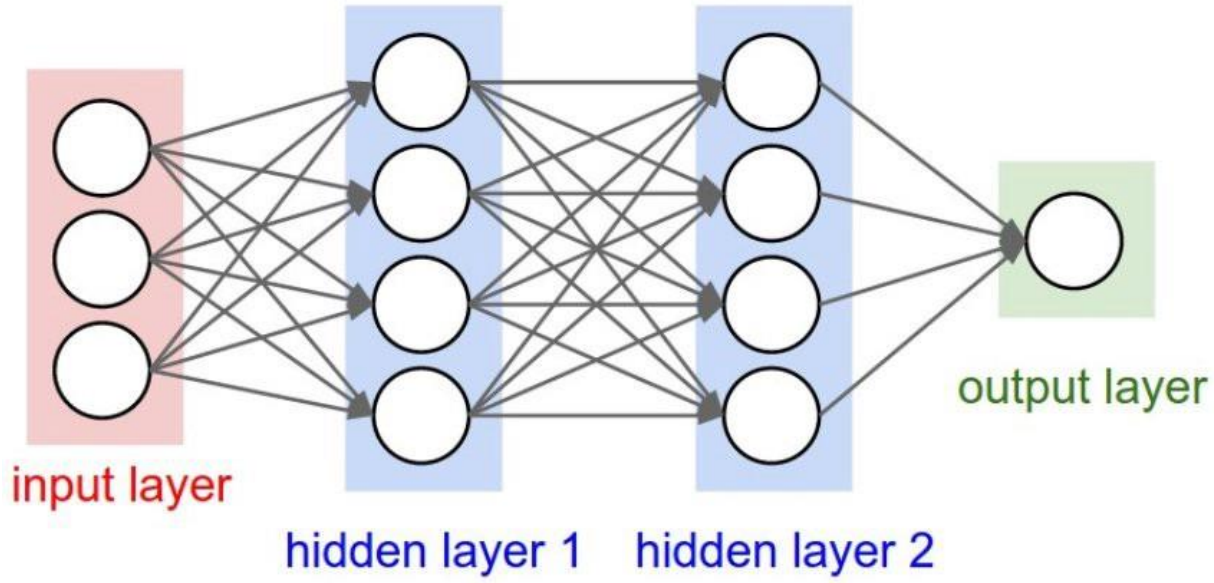

Figure 11: Modèle Séquentiel Architecture [10]

III.2 Les étapes du modèle séquentiel

III.2.1 Charger les données

La première étape consiste à définir les fonctions et les classes que nous avons l'intention d'utiliser. Nous utilisons la bibliothèque NumPy pour charger notre jeu de données et nous utiliserons deux classes de la bibliothèque Keras pour définir notre modèle. Puis nous pouvons charger le fichier sous forme de matrice.

III.2.2 Definer le model Keras

Les modèles dans Keras sont définis comme une séquence de couches.

Nous créons un modèle séquentiel et ajoutons des couches une par une jusqu'à ce que nous soyons satisfaits de notre architecture réseau.

La première chose à faire est de s'assurer que la couche en entrée a le bon nombre d'entités en entrée. Cela peut être spécifié lors de la création de la première couche avec l'argument input_dim**.**

Les couches entièrement connectées sont définies à l'aide de la classe Dense. Nous pouvons spécifier le nombre de neurones ou de nœuds dans la couche comme premier argument et spécifier la fonction d'activation à l'aide de l'argument d'activation.
Nous utiliserons la fonction d'activation de l'unité linéaire rectifiée appelée ReLU sur les deux premières couches et la fonction Sigmoïde dans la couche de sortie.

III.2.3 Compiler le modèle Keras

Maintenant que le modèle est défini, nous pouvons le compiler.
La compilation du modèle utilise les bibliothèques numériques efficaces sous les couvertures (le soi-disant backend) telles que Theano ou TensorFlow. Le backend choisit automatiquement la meilleure façon de représenter le réseau pour la formation et de faire des prédictions à exécuter sur votre matériel, tel que CPU ou GPU ou même distribué.
Lors de la compilation, nous devons spécifier certaines propriétés supplémentaires requises lors de la formation du réseau. N'oubliez pas que former un réseau signifie trouver le meilleur ensemble de poids pour mapper les entrées aux sorties dans notre ensemble de données.
Nous devons spécifier la fonction de perte à utiliser pour évaluer un ensemble de poids, l'optimiseur est utilisé pour rechercher parmi différents poids pour le réseau et toutes les métriques facultatives que nous aimerions collecter et rapporter pendant la formation.

III.2.4 Adapter le modèle Keras

Nous avons défini notre modèle et l'avons compilé pour un calcul efficace.
Il est maintenant temps d'exécuter le modèle sur certaines données.
Nous pouvons entraîner ou ajuster notre modèle sur nos données chargées en appelant la fonction **fit ()** sur le modèle.
La formation se déroule sur des époques et chaque époque est divisée en lots.
Époque : un passage à travers toutes les lignes de l'ensemble de données d'apprentissage.
Lot : Un ou plusieurs échantillons pris en compte par le modèle dans une époque avant que les poids ne soient mis à jour.
Une époque est composée d'un ou plusieurs lots, en fonction de la taille de lot choisie et le modèle est adapté à de nombreuses époques.

Le processus d'apprentissage s'exécutera pendant un nombre fixe d'itérations à travers l'ensemble de données appelé epochs, que nous devons spécifier à l'aide de l argument epochs. Nous devons également définir le nombre de lignes de jeu de données prises en compte avant que les poids du modèle ne soient mis à jour à chaque époque, appelé taille de lot et défini à l'aide de l'argument batch_size.

III.2.5 Évaluer le modéle Keras

Nous avons formé notre réseau de neurones sur l'ensemble de données et nous pouvons évaluer les performances du réseau sur le même ensemble de données.
Cela ne nous donnera qu'une idée de la façon dont nous avons modélisé l'ensemble de données (par exemple, la précision du train), mais aucune idée de la façon dont l'algorithme pourrait fonctionner sur de nouvelles données. Nous avons fait cela pour plus de simplicité, mais idéalement, vous pourriez séparer vos données en ensembles de données d'entraînement et de test pour l'entraînement et l'évaluation de votre modèle.
Vous pouvez évaluer votre modèle sur votre ensemble de données d'entraînement à l'aide de la fonction **évaluer ()** sur votre modèle et lui transmettre les mêmes entrées et sorties que celles utilisées pour entraîner le modèle.
Cela générera une prédiction pour chaque paire d'entrée et de sortie et collectera les scores, y compris la perte moyenne et toutes les métriques que vous avez configurées, telles que la précision.

CONCLUSION

Dans ce chapitre, nous avons discuté des techniques que nous avons adoptées pour construire notre solution.

Chapitre 4: Etude ET Réalisation

INTRODUCTION

Ce chapitre décrit les étapes suivies pour mener à bien le projet selon la méthodologie CRISP-DM sous forme de sprints Scrum.

I Comprehension commerciale

Dans cette section, nous passons en revue la première étape du processus CRISP-DM qui peut être vu comme Sprint 0 selon le framework Scrum. Dans cette étape, nous allons définir la ligne directrice pour l'ensemble du projet.

I.1 Résultat souhaité du projet

La première étape consiste à définir l'objectif à court terme qui peut être élargi, et la liste des ressources nécessaires pour permettre l'accomplissement de cet objectif.

Objectifs d'un point de vue commercial Le projet à long terme consiste à développer un chat bot doté d'intelligence émotionnelle. Pour y parvenir, le composant principal de l'application est un module d'intelligence artificielle qui peut réaliser le langage naturel compréhension (NLU) de ce que dit l'utilisateur. Il est donc basé sur des techniques de traitement du langage naturel. Dans son projet, nous commençons par la tâche de comprendre les besoins des personnes curieuses pour l'aider.

Critères de réussite L'objectif est considéré comme atteint si nous parvenons à créer un modèle qui répondre à la besoins psychologiques de la personne curieux.

I.2 Situation actuelle

Après avoir défini l'objectif, dans cette section, nous définissons toutes les ressources nécessaires pour atteindre les objectifs du projet .Nous disposons de trois types de ressources : des données pour construire le modèle, un environnement de développement et un système matériel adéquat.
Sources de données Le tableau 11 montre les sites d'où nous obtiendrons nos données ainsi que sous quel format.

Table 11: les sites De Recherche

Sites	Data Description
Google	contient un grand nombre de sites que nous utilisons pour notre donnée.
Wikipedia	Wikipédia est une bonne source pour obtenir du texte librement accessible.

Ressources logicielles Le tableau 12 montre les bibliothèques logicielles requises pour créer le chat bot.

Table 12:les bibliothèques logicielles

Library	Description
Keras + Tensorflow	• TensorFlow est un outil open source écrit en Python pour la programmation d'apprentissage automatique, développé par Google. • Keras est une API open source qui repose sur Tensorflow, facilitant le travail avec Tensorflow, d'où son nom d'API de haut niveau.
Pandas	Bibliothèque open source pour la manipulation de données
Numpy and Spacy	Sont des bibliothèques Python pour le calcul numérique qui prennent en charge les opérations vectorielles

Configuration système Le tableau des ressources matérielles 13 montre la configuration système nécessaire pour construire le chat bot.

Table 13 : Ressources Matérielles

Système	Description
Type de machine	Acer 64 Windows 10 et Asus 64 Windows 10
Configuration de la machine	Intel® Core™i5-8250U 1.6GHz Intel® Core™i5-6200U 2.8GHz
Taille de la mémoire (RAM)	8GO, 8GO

Comme pour tout projet data science, une étude sur l'état de l'art est à réaliser afin de se situer par rapport aux solutions existantes. Avant de construire le chat bot, nous allons voir quelles techniques et approches nous aideront à atteindre notre objectif.

I.3 Planification du projet

Après avoir défini l'objectif et les ressources nécessaires, nous présentons dans cette section les processus CRISP-DM en tant que ressorts Scrum du processus global. Nous fournissons un cas d'utilisation général, et un arriéré de produits.

Cas d'utilisation général La figure 12 montre le cas d'utilisation pour l'ensemble du processus.

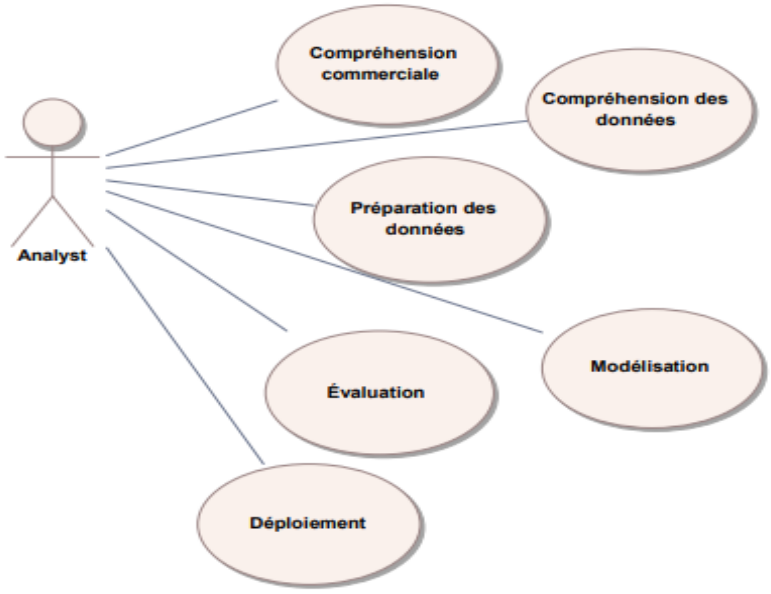

Figure 12: Cas d'utilisation Globale

I.4 Specification Des Exigences

Dans cette section, nous présentons les exigences et les acteurs impliqués dans le développement traiter.

Acteurs Il y a deux rôles pendant tout le processus :

• Analyste : effectuera toutes les tâches liées à la science des données, de la planification du projet au déploiement du modèle

• Curieux : testera l'application avec le modèle déployé

Exigence Le projet consiste à développer un prototype qui n'est pas à mettre à disposition usage public. Par conséquent, toutes les exigences sont décrites du point de vue de l'analyste :

• Définir un objectif réaliste sous les contraintes

• Avoir un inventaire clair des ressources et de la situation actuelle

• Créer un référentiel de données avec lequel travailler

• Disposer d'un ensemble de données prêt à l'emploi dans le format souhaité

• Pour trouver les paramètres optimaux pour réussir à construire le modèle

• Affiner le modèle jusqu'à ce que nous atteignions l'objectif souhaité défini dans l'étape objectif

• Trouver la stratégie adéquate pour déployer le modèle

Il n'y a pas de directives prédéfinies pour la construction du modèle. Par conséquent, pour atteindre la sortie souhaitée, l'ensemble du processus sera réexécuté jusqu'à ce que nous atteignions l'objectif.

I.5 Backlog

Dans cette section, nous présentons le backlog du produit et la répartition des tâches.

Table 14: Carnet de sprint

ID	User Story	Critères d'acceptation	Pr	Est
1	En tant qu'analyste, j'ai besoin d'acquérir des données provenant de sources spécifiées dans l' étape de rabotage	Quand j'ai un objectif et un inventaire des ressources	1	2
2	En tant qu'analyste, j'ai besoin d'examiner les données en visualisant son contenu	Quand j'ai un rapport sur l'organisation des données	2	2
3	En tant qu'Analyste, établir un rapport de qualité	Lorsque j'ai un rapport sur des données valides et fichiers non corrompus	3	2
4	En tant qu'analyste, je dois nettoyer les données acquises	Quand j'ai appliqué tout le nettoyage routines sur les données	4	5
5	En tant qu'analyste, j'ai besoin d'organiser les données dans les tableaux	quand j'ai un référentiel structuré de fichiers	5	3
6	En tant qu'analyste, je dois effectuer des données l'intégration	Quand je suis prêt à utiliser l'ensemble de données au format souhaité	6	3
7	En tant qu'analyste, j'ai besoin de former le modèle.	Lorsque la phase de formation est terminée et le modèle est prêt à l'emploi	7	5
8	En tant qu'analyste, je dois évaluer Maquette	Lorsque le modèle a montré la précision souhaitée sur des données invisibles	8	3
9	En tant qu'analyste, je dois déployer mon modèle en tant qu'application Web	Lorsque l'application Web est prête utiliser	9	5
10	En tant que curieux, je dois tester l'application	quand l'application fonctionne et je peut parler au chatbot	10	2

II Sprint 1: Compréhension des données

Cette section décrit le Sprint 1 qui est la phase de compréhension des données du processus CRISP DM.

II.1 Backlog du Sprint

Le tableau 15 montre le backlog de Sprint pour le Sprint 1.

Table 15:Backlog de sprint

ID	User Story	Tâches
1	En tant qu'analyste, j'ai besoin d'acquérir des données de spécifié dans l'étape de rabotage	Vérifier les ensembles de données prêts à l'emploi Extraire les données des pages wikipedia
2	En tant qu'analyste, j'ai besoin d'examiner les données en visualisant son contenu.	Classer les jeux de données par format Examiner les erreurs
3	En tant qu'analyste, établissez un rapport de qualité	Vérifiez les fichiers corrompus et supprimez eux

II.2 Cas d'utilisation du sprint

En figure 13, nous voyons le cas d'utilisation du sprint 1.

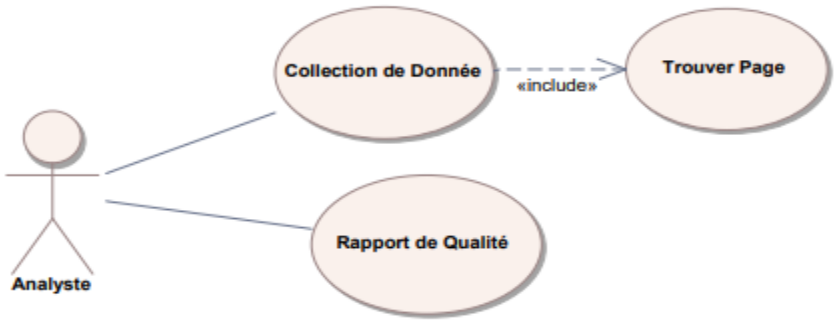

Figure 13:Cas d'utilisation Sprint 1

II.3 Mise en œuvre du sprint

Les données requises pour le processus sont composées de données prétraitées qui sont disponibles sous forme de jeu de données prêt à l'emploi.

Exploration des jeux de données Le premier défi rencontré lors de cette tâche est la diversité des formats d'où proviennent les données. Nous avons utilisé des jeux de données au format JSON, nous avons extrait des données de pages web, ce qui posait problème c'est la difficulté d'automatiser le processus puisque chaque page a sa propre structure, tout cela nous l'avons fait par inspection manuelle

Rapport de qualité L'avantage d'utiliser les ensembles de données disponibles dans différentes pages Web en recherchant et en récupérant les données car il existe de nombreux problèmes de grattage des données. La plus évidente est que ces ensembles de données ont été créés à des fins différentes des nôtres. Un autre problème est que dans chaque ensemble de données, il y avait plusieurs fichiers qui affichaient divers problèmes. D'autres problèmes comme un mauvais formatage.

II.4 Revue de sprint et retrospective

Revue de sprint Au cours de ce sprint, nous avons créé un jeu de données en collectant une quantité de données à partir de sites Web afin de les utiliser pour le reste du projet.

Rétrospectives Les difficultés rencontrées sont l'absence de jeu de données psychologiques et aussi principalement liées à la faiblesse des moyens de collecte des données. en plus, il y a peu de sites web disponibles pour la psychologie.

III Sprint 2: Preparation des données

Dans cette section, nous passons en revue la mise en œuvre de Sprint 2 qui couvre la phase de préparation des données du processus CRISP-DM.

III.1 Backlog de sprint

Le tableau 16 montre le backlog pour Sprint 2.

Table 16:Backlog

ID	User Story	Taches
4	En tant qu'analyste je dois nettoyer les données acquises	Trouvez des contenus indésirables tels que des mots hors vocabulaire et supprimez-les Déboîter le texte Supprimer les mots sans signification (mots vides) Trouvez des contenus indésirables tels que des mots hors vocabulaire et supprimez-les Déboîter le texte
5	En tant qu'analyste, j'ai besoin d'organiser les données dans des tableaux	Créez les colonnes avec les étiquettes adéquates, chargez les fichiers et attribuez à chaque ligne le bon texte dans la colonne de droite
6	En tant qu'analyste, je dois effectuer l'intégration de données	Créer l'ensemble de données en joignant les lignes des tables créées Stocker les données fusionnées créées sous forme de fichiers CSV

III.2 Cas d'utilisation de sprint

En figure 14, nous présentons le cas d'utilisation du Sprint 2.

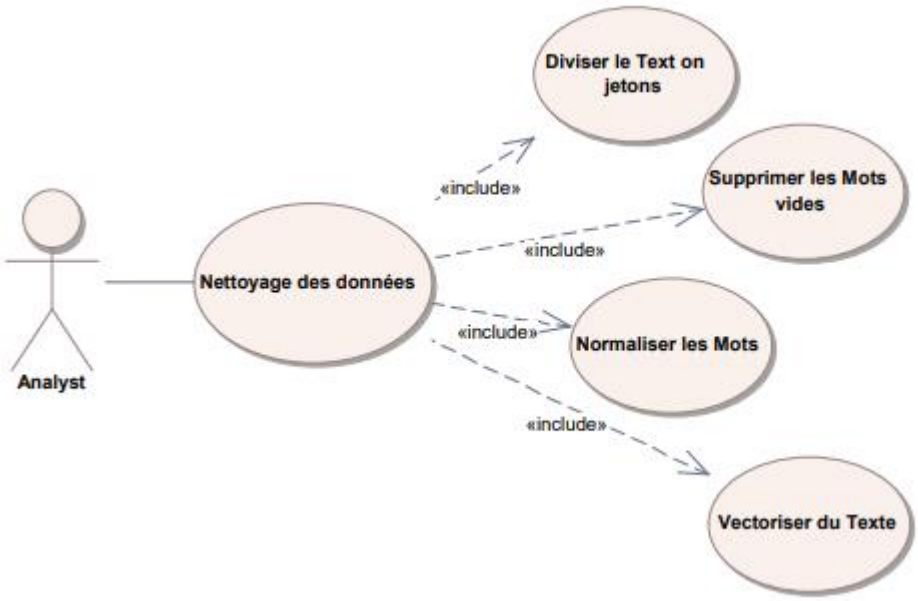

Figure 14:Cas d'utilisation :Préparation du données

III.3 Mise en œuvre du sprint

La mise en œuvre de Sprint 2 implique les étapes suivantes :

Nettoyage des données Pour nettoyer les données, qui sont du texte brut, nous avons appliqué une correspondance de modèle avec des expressions régulières. Plusieurs parties de l'ensemble de données contiennent plusieurs mots qui sont hors du vocabulaire, tels que des abréviations utilisées dans les discussions, et qui sont compris par les humains mais n'ont aucune entrée dans aucun dictionnaire.

En tableau 17, fournissez toutes les opérations que nous avons effectuées pour nettoyer le texte

Table 17:les opérations pour nettoyer le texte

Tâche	Techniques à résoudre
Tokenisation	Le texte brut est lu par ligne, donc pour le traitement, nous devions d'abord le diviser en jetons.
Supprimer les mots vides	Les mots vides sont des mots qui apparaissent très souvent dans le texte mais n'ont pas de sens sémantique exemple : "dans" "le", "a", "sur", "est", "tout". Ces mots n'ont pas de sens significatif, les supprimer est utile pour réduire l'effort de calcul.
Normaliser les mots	condenser toutes les formes d'un mot en une seule représentation de forme la plus simple
Vectoriser du texte	représenter chaque jeton d'une manière qu'une machine peut comprendre ,par le transformer en une vecteur ou un tableau numérique

III.4 Revue de sprint et retrospective

Revue de sprint L'objectif principal de ce sprint est de préparer un ensemble de données à utiliser pour la prochaine étape. Nous avons construit un ensemble de données structuré et étiqueté qui est facile à utiliser dans le réglage fin du modèle.

Rétrospectives Plusieurs expériences ont consisté à resélectionner les ensembles de données jusqu'à ce que nous remarquions un net progrès. Le rapport de ce projet décrit comment formater l'ensemble de données afin d'obtenir une grande précision et moins d'effort de calcul.

IV Sprint 3: Modélisation et evaluation

Dans cette section, nous montrons la mise en œuvre du Sprint 3, qui couvre les étapes de modélisation et d'évaluation du processus CRISP-DM.

IV.1 Backlog de sprint

Le tableau 18 répertorie le backlog de sprint pour le Sprint 3.

Table 18: Backlog

ID	User Story	Tâches
7	En tant qu'analyste, je dois former le modèle	Définir les hyper paramètres Ajustez les hyper paramètres si besoin
8	En tant qu'analyste, j'ai besoin d'évaluer le modèle	Exécuter le modèle sur un test Rapporter les résultats

IV.2 Sprint Use Case

Figure 15 montres le diagramme de cas d'utilisation de Sprint 3.

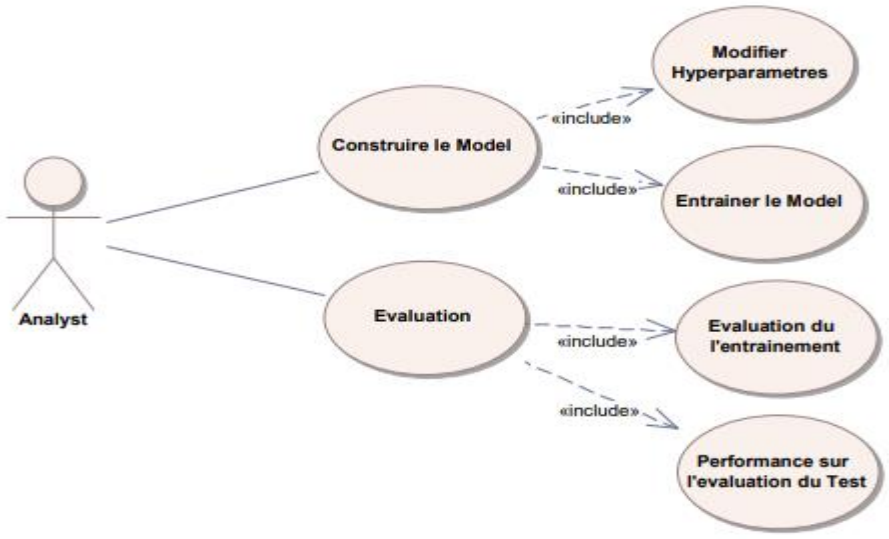

Figure 15:diagramme de cas d'utilisation

IV.3 Mise en œuvre du sprint

Définition des hyper paramètres Dans ce sprint, nous formons le classificateur de la couche de réglage fin pour la tâche de classification des intentions. Nous avons utilisé l'ensemble de données pour entraîner une classificatrice multi-étiquette avec les hyper paramètres suivants :

Table 19: Classifier Hyperparameters

Hyperparamètre	**Valeur**
Nombre d'époques	200
Optimiseur	Sgd
Taux d'apprentissage	0.01
Fonction de perte	Categorical cross entropy

Formation du modèle La Figure 16 montre la configuration du modèle avant de commencer le processus de formation.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 128)               58112
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 110)               7150
=================================================================
Total params: 73,518
Trainable params: 73,518
Non-trainable params: 0
_________________________________________________________________
```

Figure 16: Résumé du modèle

Évaluation La figure 17 montre l'évolution des valeurs de précision au cours des 200 époques, la figure 18 montre comment les valeurs de perte diminuent.

Figure 17:Train Accuracy

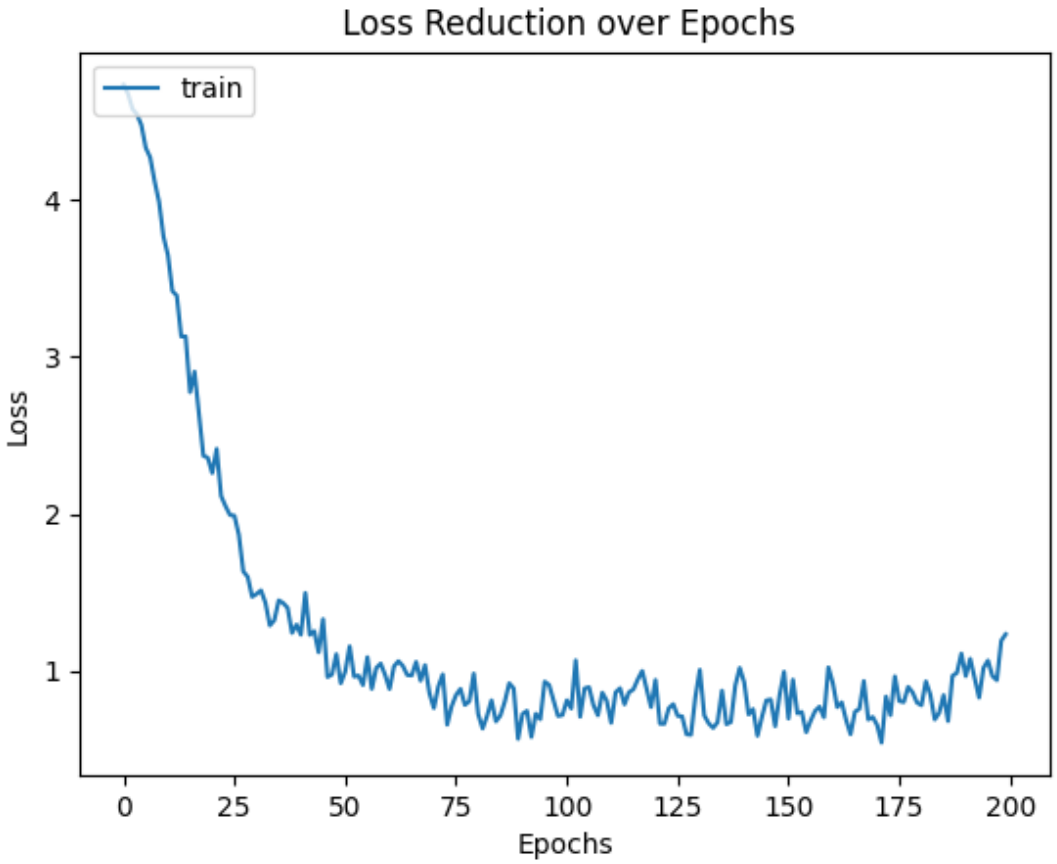

Figure 18:Loss Reduction over Epochs

IV.4 Revue Sprint ET Rétrospective

Revue de sprint L'apprentissage du modèle séquentiel n'a été possible qu'après plusieurs tentatives pour ajuster le jeu de données. À la fin, nous avons réussi à construire le model.

Rétrospectives Une formation réussie a été effectuée après avoir défini les bons hyper paramètres. La première fois, nous choisissons de nous entraîner pour optimizer=Adam, même sur une très petite partie des données.

V Sprint 4: Déploiement

Dans cette section, nous montrons l'implémentation de Sprint 4. Nous déployons le modèle construit dans le précédent en tant qu'application Web.

V.1 Backlog de sprint

Le tableau 20 montre le backlog pour Sprint 4.

Table 20:backlog

ID	User Story	Tâches
9	En tant qu'Analyste, je dois déployer mon modèle en tant qu'application Web	Créer le backend de l'application Créer le front-end
10	En tant qu'analyste, je dois tester l'application	Tester et évaluer l'application

V.2 Cas d'utilisation de sprint

La figure 19 montre le cas d'utilisation du Sprint 4.

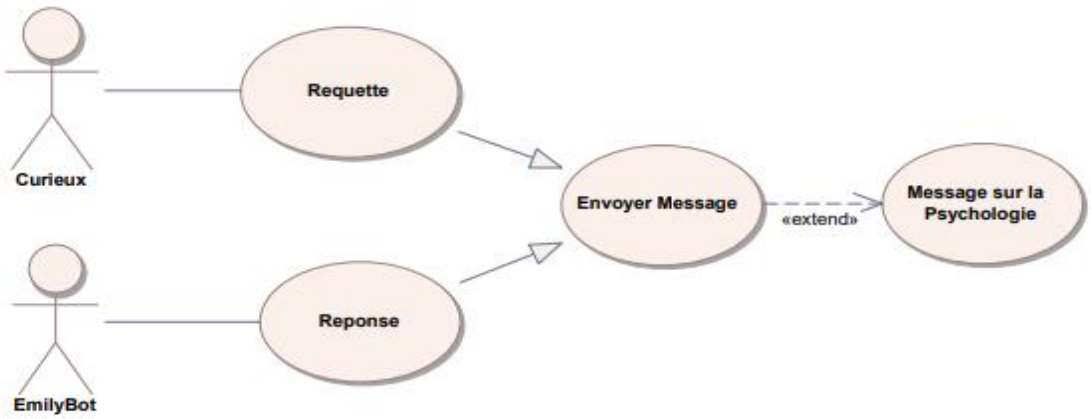

Figure 19:Cas d'utilisation: Chatbot

V.3 Mise en œuvre du sprint

La mise en œuvre de Sprint 4 implique les étapes suivantes :

Architecture de l'application Afin de tester le chat bot, nous construisons une application basée sur le framework Flask. L'application suit l'architecture modèle-vue-contrôleur (MVC) qui sépare les responsabilités du flux de travail. La figure 20 montre les architectures de l'application. Dans le tableau 21, nous décrivons chaque couche de l'application selon l'architecture MVC.

Table 21:Architecture MVC

Model	sont les données utilisées pour exécuter la logique métier. Dans ce projet, le modèle est l'entrée de l'utilisateur.
View	Est la partie qui rend les données au client. L'application utilise une page Web en tant que client
Controller	Est-ce la partie de l'application qui met à jour les données et contenu à rendre

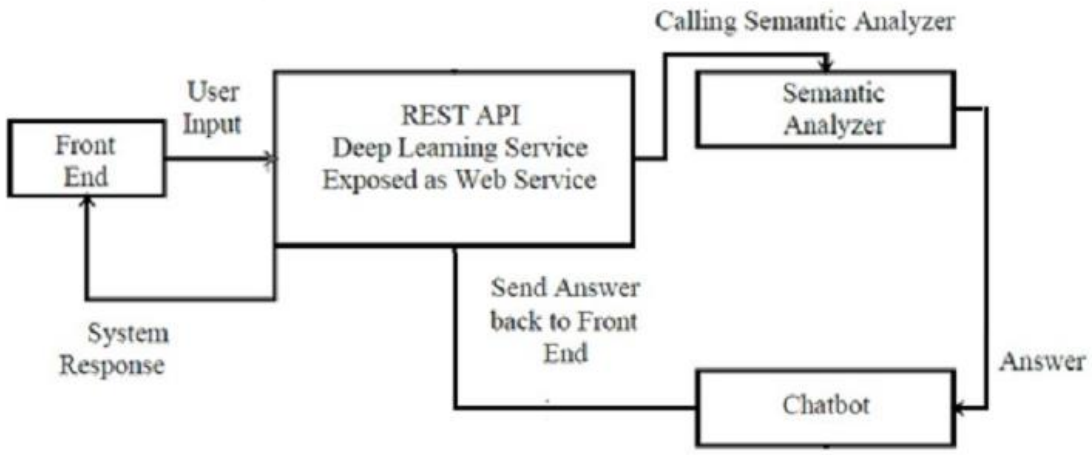

Figure 20:Web Application Architecture

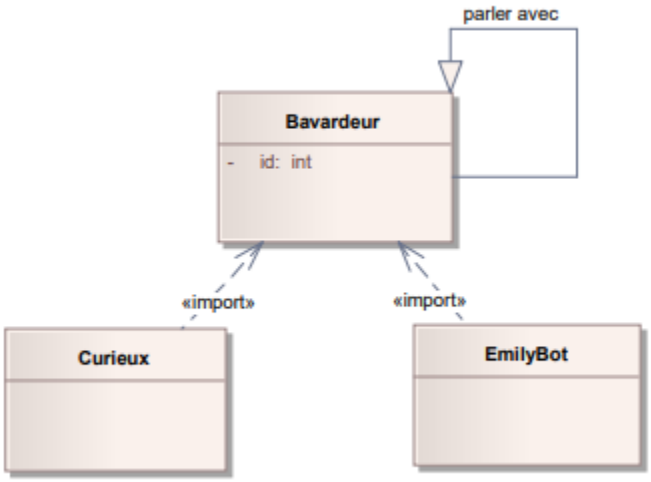

Figure 21:Diagramme De Class

La figure 22 montre l'interface de chat avec Emily de l'application Web:

Figure 22:Chat Avec Emily

V.4 Revue de sprint ET Rétrospective

Revue de sprint Dans ce sprint, nous avons construit une application Web pour démontrer les performances du chat bot. Pour les améliorations futures, l'architecture sera conservée, et le modèle sera mis à jour après un réapprentissage avec des données et des moyens de calcul plus importants.

Rétrospectives L'application a bien fonctionné, mais même si elle a été formée sur un très petit ensemble de données, le temps de réponse est lent. Si l'utilisateur pose une question, cela peut prendre jusqu'à une minute avant que l'ordinateur ne réponde.

CONCLUSION

Ce chapitre décrit les étapes de réalisation du projet selon la méthodologie CRISP-DM. Nous commençons par présenter la compréhension métier, puis la partie data, qui comprend la compréhension. Ensuite, nous passerons au modèle et à son évaluation. Pour conclure le chapitre, nous donnerons un aperçu de l'application finale.

CONCLUSION

Le projet consistait à créer un système basé sur l'intelligence artificielle et plus particulièrement, le traitement du langage naturel. L'idée principale est d'analyser l'entrée de l'utilisateur et de détecter son intention. Un tel système a l'avantage de cibler le besoin de l'utilisateur au moment de la saisie. Tout d'abord, nous avons présenté le contexte du projet en présentant l'entreprise d'accueil du stage, en définissant l'énoncé du problème et en donnant un bref aperçu de notre solution. il est à noter que nous avons présenté une méthodologie de gestion de projet adéquate, afin de guider l'ensemble du processus de création du projet.

Ensuite, nous avons effectué une revue de concepts théoriques importants ainsi qu'une étude approfondie sur l'état de l'art des systèmes de recommandation et de l'intelligence artificielle, notamment en PNL afin d'identifier la position de notre projet par rapport aux solutions déjà disponibles. Selon la méthodologie adoptée, le projet doit être réalisé dans un cycle de six étapes majeures qui peuvent être regroupées en deux parties à savoir une partie liée à la compréhension et aux données commerciales et à la mise en œuvre comprenant la construction du modèle, son évaluation et son déploiement dans un environnement réel. cas d'utilisation mondial. Avant de construire le modèle, nous devions collecter les bonnes données et les préparer pour une utilisation dans l'étape de construction du modèle. Cette étape impliquait la construction d'une ontologie du domaine d'intérêt, qui est la psychologie. L'étape suivante consistait à collecter les données par les pages web. Une fois les données collectées, une étape de pré-traitement a été réalisée afin de les utiliser pour construire le modèle. La construction du modèle a été réalisée à l'aide d'un modèle pré-entraîné. Après s'être assuré que le modèle atteignait un niveau de précision acceptable, nous l'avons déployé en tant que chatbot dans une application Web.

Plusieurs améliorations peuvent être apportées

• En raison de la limitation de la puissance de calcul, un peu de données collectées ont été utilisées lors de l'apprentissage du modèle. Par conséquent, afin de donner au modèle plus de capacités, un environnement d'exécution adéquat doit être mis en

place. La solution de pointe est pour demander des services cloud GPU sans restrictions de temps/d'utilisation.

• L'ajout de la prise en charge de la figure de style peut rendre l'application plus humaine, ce qui peut contribuer à la confiance de l'utilisateur dans le système. Si le système peut capturer si l'utilisateur utilise la métaphore, il peut aider le système à identifier des besoins plus profonds.

Bibliographie

[1][Enligne].Available:https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FThe-Steps-of-a-KDD-process_fig7_220073492&psig=AOvVaw07shs1WSv1kqNRokvXBZzL&ust=1631794609261000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCLCJgeT6gPMCFQAAAAAdAAAAABAD

[2] [En ligne].Available:https://fr.wikipedia.org/wiki/SEMMA

[3] [En ligne].Available:https://www.datascience-pm.com/semma/

[4][Enligne].Available:https://www.google.com/search?q=semma+DEFINITION&rlz=1C1GCEA_enTN930TN930&sxsrf=AOaemvK4KG8pXUnALAbV-3NP4P0253V6Mg:1631707338143&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjulJHC94DzAhUH26QKHS59A_EQ_AUoAXoECAIQAw&biw=1366&bih=625&dpr=1#imgrc=ZY0-47nuUj-xvM&imgdii=SZLocy5a5K1qFM

[5] [En ligne].Available:https://www.ibm.com/docs/fr/spss-modeler/SaaS?topic=dm-crisp-help-overview

[6][Enligne].Available:https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.solutions-numeriques.com%2Fexpertise-la-methode-crisp-une-solution-pour-reussir-vos-projets-big-data-alianor-sibai-mc2i-groupe%2F&psig=AOvVaw311Y7tqBhhYB20QLFFQ-KB&ust=1631801056815000&source=images&cd=vfe&ved=0CAwQjhxqFwoTCJiS6uiSgfMCFQAAAAAdAAAAABAP

[7] [En ligne].Available:Azevedo et al. “KDD semma and CRISP-DM: A parallel overview”. In: Jan. 2008,pp. 182–185

[8][Enligne].Available:https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FAn-example-of-word-vectors-embedding-in-two-dimensional-

space_fig2_349915332&psig=AOvVaw2bGwkKluTj5kRZmdhC5s5V&ust=1632179703950000&source=images&cd=vfe&ved=0CAwQjhxqFwoTCIiqsq2VjPMCFQAAAAAdAAAAABAD

[9][Enligne].Available:https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.kdnuggets.com%2F2018%2F06%2Fbasic-keras-neural-network-sequential-model.html&psig=AOvVaw1IEoQtNGJfjpC6IlMswR7p&ust=1632229840477000&source=images&cd=vfe&ved=0CA0Q3YkBahcKEwiI46eX0I3zAhUAAAAAHQAAAAAQIw

[10] [En ligne].Available:https://towardsdatascience.com/building-a-deep-learning-model-using-keras-1548ca149d37

[11] [En ligne].Available:Sanjay Churiwala Gopinath Rebala Ajay Ravi. An Introduction to Machine Learning. Springer, 2019. isbn: 978-3-030-15729-6. url: http://gen.lib.rus.ec/book/

index.php?md5=ab0e27673dc00812c19cac58b8c0801f.

[12] [En ligne].Available:Ajay Shrestha and Ausif Mahmood. “Review of Deep Learning Algorithms and Architectures”. In: IEEE Access PP (Apr. 2019), pp. 1–1. doi: 10.1109/ACCESS.

2019.2912200.

[13] [En ligne].Available:Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. http://www.deeplearningbook.org. MIT Press, 2016.

[14] [En ligne].Available:Musab Coşkun et al. “AN OVERVIEW OF POPULAR DEEP LEARNING METHODS”. In: European Journal of Technic 7 (Dec. 2017), pp. 165–176. doi: 10.23884/ejt.2017.7.2.11.

[15] [En ligne].Available:Zhe Wang et al. “Chinese Poetry Generation with Planning based Neural Network”. In: CoRR abs/1610.09889 (2016). arXiv: 1610.09889. url: http://arxiv.org/

abs/1610.09889.