

UNIVERSITE HASSAN II CASABLANCA

ECOLE NATIONALE SUPERIEUR D'ART ET METIERS

DEPARTEMENT INTELLIGENCE ARTIFICIELLE ET GENIE INFORMATIQUE

## **STAGE D'INITIATION (PROJET PERSONNEL)**

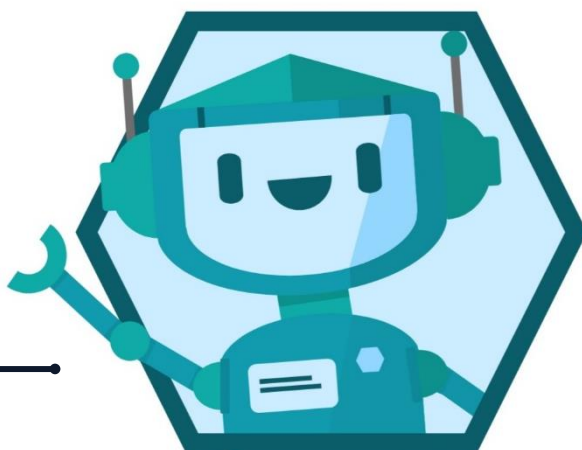
### **Sujet : CONCEPTION ET DÉVELOPPEMENT D'UN CHATBOT**

**Réaliser par :**

- **EZZAHRAOUI Haroun**
- **GATOU Amine**
- **LIFI El Mostafa**
- **BAYOUD Hamza**

**Encadrer par :**

- **BAHNASSE Ayoub**



**Année Universitaire 2020-2021**

# Remerciements

Avant d'ouvrir ce rapport, je profite de l'occasion pour remercier **ALLAH** pour la volonté, la force et la santé qu'il nous a donnée afin de réaliser ce travail.

Nous tenons à remercier toute l'équipe pédagogique du cycle ingénieur, Intelligence artificielle et génie informatique. En personne du chef de filière, Monsieur **MOUTACHAOUIK** nos enseignants qui nous ont accompagnés pendant cette année et qui ont partagé avec nous leurs expériences et qui nous ont encouragés sans cesse.

Nos remerciements sont également adressés à Monsieur **BAHNASSE Ayoub**, notre encadrant à la ENSAMC pour ses conseils et recommandations pertinentes durant toute la période de ce travail, nous m'avons fait le grand honneur d'accepter de nous diriger dans ce travail avec bienveillance et rigueur.

Nos remerciements vont aussi à l'encontre de nos chers parents qui n'ont ménagés aucun effort et qui ne cesseront de le faire pour nous soutenir à tout moment de notre vie.

Finalement, nous remercions l'ensemble des membres du jury qui nous ont fait l'honneur d'accepter d'évaluer et de juger notre travail.

# Résumé

Dans le cadre de notre formation à l'école nationale supérieure d'art et métiers Casablanca, on a eu l'opportunité de réaliser un stage d'observation dans l'entreprise de notre choix. Mais à cause de la pandémie que le monde fait face à elle, on n'a pas pu faire un stage externe dans une entreprise, c'est pour cela que nos enseignants et toute l'équipe pédagogiques de l'intelligence artificielle et génie informatique nous ont demandé des projets en nous encadrant à distance.

L'objectif de ce travail est la conception et la réalisation d'un service d'un chatbot qui permettra de répondre rapidement aux questions posées par l'utilisateur.

La lecture des rapports scientifiques, des ouvrages de référence et de la documentation technique des outils disponibles, nous ont permis d'acquérir des connaissances permettant de comprendre les procédures méthodiques et de les appliquer correctement. Cela comprenait la collecte et l'enregistrement des données de formation pour les modèles, et le développement et la formation du prototype.

Chers lecteurs, nous optons que ce modeste rapport vous fournira autant d'informations bénéfiques que possible sur notre projet.

# Abstract

As part of our training at the Casablanca National School of Arts and Crafts, we had the opportunity to do an observation internship in the company of our choice. But because of the pandemic that the world is facing, we were not able to do an external internship in a company, this is why our teachers and the entire teaching team at artificial intelligence and computer engineering have asked us for projects within us framing at a distance.

The objective of this work is the design and implementation of a chatbot service that will allow quick response to questions posed by the user.

Reading scientific reports, reference books and documentation technical tools available, allowed us to acquire knowledge allowing to understand methodical procedures and to apply them correctly. This included collecting and recording training data for models, and development and the formation of the prototype.

Dear readers, we opt that this modest report will provide you with so much information beneficial as possible on our project.

# Introduction

De sa nature, l'être humain a toujours tendance à chercher le meilleur, c'est pourquoi, celui-ci ne cesse de se développer afin d'atteindre ses objectifs et de réaliser ses rêves. En conséquence, cela permet au monde de connaître des évolutions dans tous les domaines et plus particulièrement dans le domaine de communication.

Aujourd'hui, il n'y a plus de domaines qui ne fassent usage de l'IA, qui offre tant de services pour l'entreprise que pour le personnel.

L'IA touche à tout ce qui va se rapprocher d'un comportement humain, et avec l'évolution exponentielle dans le domaine du Deep Learning (DL) d'une manière générale et dans le domaine du Traitement du Langage Humain (TLH) en particulier, il est devenu possible de créer des applications (chatBots) capables de nous comprendre et de nous répondre d'une manière instantanée et sans rupture.

Nous cherchons à développer un chatBot pour gérer le grand nombre des questions posée par l'utilisateur avec une disponibilité 24/7.

Chatbot est un projet qui se divise en trois grandes phases de Travail. Premièrement, une phase d'étude de technologies puis la conception et le développement de L'application et finalement la conception et le développement de l'interface utilisateur. Notre choix de créer un chatbot pour l'école se justifie de l'intérêt que nous avons porté aux problématiques que posait le sujet, mais également du fait qu'aussi bien dans des domaines éducatifs que de la recherche.

Les agents conversationnels et Assistants vocaux sont depuis 2016 incontournables pour de nombreux dispositifs et deviennent une Alternative très souvent envisagée à certaines interfaces. Comme on peut définir Chatbot comme suit :

Chatbot vient du terme « bot », diminutif de « robot ». Il désigne un programme conçu pour dialoguer avec des utilisateurs via une plateforme ou une application de messagerie. L'outil est aussi appelé « agent conversationnel », car son objectif est d'échanger avec des internautes. Le chatbot diffère donc en ce sens du bot et du robot.

# Table des matières :

<b>Remerciements.....</b>	<b>2</b>
<b>Résumé/Abstract.....</b>	<b>3</b>
<b>Introduction .....</b>	<b>4</b>
<b>Chapitre I : Présentation du cadre de projet</b>	
<b>I. Présentation de l'organisme d'accueil.....</b>	<b>11</b>
1. Présentation de l'école.....	11
2. Histoire.....	11
3. Missions.....	11
4. Admission à la formation ingénieur.....	11
5. Filière.....	12
6. Administration.....	12
<b>II. Projet.....</b>	<b>13</b>
1. Présentation du projet.....	13
1. <i>Etude de l'existant</i> .....	13
2. <i>Besoins fonctionnels</i> .....	13
3. <i>Besoins techniques</i> .....	13
2. Planification du projet.....	14
3. Conception.....	15
1. <i>Description du cahier de charge</i> .....	15
2. <i>Analyse des cas d'utilisation</i> .....	15
3. <i>Analyse des interactions</i> .....	16
4. <i>Modélisation orienté objet</i> .....	17

## Chapitre II : Etat de l'art du projet

I.	Etat de l'art et l'apprentissage profond.....	18
1.	Etat de l'art des Chatbots .....	18
1.	<i>Chatbot : Définition</i> .....	18
2.	<i>Une brève histoire de Chatbot</i> .....	18
3.	<i>Types et Architectures des Chatbots</i> .....	19
i.	<i>Architecture : Retrieval vs Generative</i> .....	21
2.	L'apprentissage profond .....	22
1.	<i>Réseau de neurones artificiels</i> .....	22
2.	<i>Fonctions d'activation</i> .....	23
i.	<i>Linéaires</i> .....	23
ii.	<i>Avec Seuil</i> .....	24
iii.	<i>Sigmoïde</i> .....	24
iv.	<i>Tangente hyperbolique</i> .....	25
v.	<i>Unités linéaires rectifiées</i> .....	25
vi.	<i>Softmax</i> .....	26
3.	<i>Réseau de neurones</i> .....	26
i.	<i>RNN</i> .....	27
ii.	<i>CNN</i> .....	31
3.	Traitement automatique du langage naturel .....	33
1.	Définition .....	33
2.	Principes défis du NLP .....	34
3.	Domaines du NLP .....	34
4.	Outils de base du NLP .....	34
5.	Représentation des mots .....	35
6.	Word Embeeding .....	36
7.	Word2Vec .....	36

## **Chapitre III : Réalisation par PYTHON**

<b>I. Réalisation par Python .....</b>	<b>38</b>
1. Langage de programmation .....	38
2. Environnement .....	38
3. Nettoyage de données .....	38
4. Corpus et base de données .....	39
5. Word Embeddings .....	39
6. Word2Vec .....	40
7. Model .....	40
8. Entraînement et Test .....	42
9. Architecture .....	42
10. Prototype et résultat .....	43

## **Chapitre IV : Réalisation par LARAVEL**

<b>I. Réalisation par Laravel .....</b>	<b>44</b>
1. Langage et environnement .....	44
2. Création de l'agent .....	45
3. Génération du JWT .....	48
4. Installation de Laravel .....	53
5. Installation de Botman .....	53
6. Création d'application Laravel .....	54
7. Ajout de Botman à Laravel .....	54
8. Prototype et résultats .....	57

<b>Conclusion .....</b>	<b>59</b>
-------------------------	-----------

<b>Bibliographie .....</b>	<b>60</b>
----------------------------	-----------

# Table des figures :

<b>Figure 1-</b> Diagramme de Gantt .....	12
<b>Figure 2-</b> Diagramme de cas d'utilisation .....	14
<b>Figure 3-</b> Diagramme Use Case .....	14
<b>Figure 4-</b> Diagramme de classes.....	15
<b>Figure 5-</b> Alan Turing (1912 - 1954).....	16
<b>Figure 6-</b> Test de TURING .....	17
<b>Figure 7-</b> Classification d'un chatbot .....	18
<b>Figure 8-</b> Architecture retrieval.....	19
<b>Figure 9-</b> Architecture générative .....	20
<b>Figure 10-</b> Un neurone biologique .....	20
<b>Figure 11.1-</b> Modélisation Mathématique d'un neurone artificiel .....	21
<b>Figure 11.2-</b> Fonction d'activation linéaire .....	21
<b>Figure 11.3-</b> Fonction d'activation avec seuil .....	22
<b>Figure 11.4-</b> Fonction d'activation sigmoïde .....	22
<b>Figure 11.5-</b> Fonction d'activation tangente hyperbolique.....	23
<b>Figure 11.6-</b> Unités linéaires rectifiées.....	23
<b>Figure 12.1-</b> Un réseau de neurones .....	24
<b>Figure 12.2-</b> Un réseau de neurones Profond .....	25
<b>Figure 13.1-</b> Réseau de neurones récurrent .....	25
<b>Figure 13.2-</b> Cellule LSTM .....	26
<b>Figure 13.2.1-</b> Forget gate layer.....	27
<b>Figure 13.2.2-</b> Input gate layer .....	27
<b>Figure 13.2.3-</b> Mise à jour du vecteur d'état de cellule du LSTM .....	28
<b>Figure 13.2.4-</b> Output gate" du LSTM .....	28
<b>Figure 13.3-</b> Gated Recurrent Unit .....	28
<b>Figure 14.1-</b> Une séquence CNN pour classer les chiffres manuscrits .....	29
<b>Figure 14.2-</b> Convolution 2D .....	30
<b>Figure 14.3-</b> Modèles du Word2Vec .....	35
<b>Figure 15.1-</b> nettoyage de données .....	36
<b>Figure 15.2-</b> Fichier YML contient l'ensemble des Q/A.....	37
<b>Figure 16-</b> Architecture du Model school.....	39
<b>Figure 16.1-</b> Modèle Encodeur Décodeur .....	39
<b>Figure 17-</b> Architecture de CHATBOT-ENSAM .....	40
<b>Figure 18-</b> Premier Prototype de CHATBOT-ENSAM .....	41
<b>Figure 19.1-</b> Dialogflow Panel .....	43
<b>Figure 19.2-</b> Création nouveau agent .....	43
<b>Figure 19.3-</b> Ajouter une entité.....	43
<b>Figure 19.4-</b> Créer des synonymes .....	44
<b>Figure 19.5-</b> Ajouter les intentions .....	44
<b>Figure 19.6-</b> Le nom de l'intention .....	44
<b>Figure 19.7-</b> Les phrases de l'entraînement.....	45
<b>Figure 19.8-</b> Les actions et les paramètres .....	45



<b>Figure 19.9-</b> Les repenses .....	46
<b>Figure 20-</b> ID de Projet .....	46
<b>Figure 21-</b> Google Cloud Platform .....	47
<b>Figure 21.1 -</b> OAuth 2.0 paramétrage .....	47
<b>Figure 21.2 -</b> Création des crédenciales .....	48
<b>Figure 21.3-</b> Les choix possible pour configurer OAuth .....	48
<b>Figure 21.4 -</b> Le nom du compte de service .....	48
<b>Figure 21.5 -</b> Les détails sur les méthodes durant la session .....	49
<b>Figure 21.6 -</b> Donnée l'accès au utilisateurs et développeurs .....	49
<b>Figure 21.7 -</b> Génération de API KEY .....	50
<b>Figure 21.8 -</b> OAuth Crédenciales .....	50
<b>Figure 21.9 -</b> Générer nouvelle key .....	50
<b>Figure 22-</b> Générer nouvelle key .....	51
<b>Figure 22.1 -</b> Générer nouvelle key .....	51
<b>Figure 23 -</b> Installation de BOTMAN .....	51
<b>Figure 24 -</b> Commande de création de projet laravel .....	52
<b>Figure 25 -</b> Commende d'intégration de BOTMAN dans le projet LARAVEL .....	52
<b>Figure 26 -</b> Création de nouveau Middleware pour assurer la connexion avec DialogflowV2 .....	53
<b>Figure 26.1 -</b> Le dossier routes .....	54
<b>Figure 26.2 -</b> Contenu de fichier Botman.php .....	54
<b>Figure 26.3 -</b> Le dossier Config .....	54
<b>Figure 26.4-</b> Modification de web.php .....	55
<b>Figure 26.5 -</b> Modification dans Botman widget .....	55
<b>Figure 27 -</b> La bulle de messagerie .....	55
<b>Figure 27.1 -</b> Premier test de ENSAM-Bot .....	56
<b>Figure 27.2 -</b> Test d'affichage des Message et repenses .....	56
<b>Figure 27.3 -</b> Test d'envoi des liens par message .....	56

## Liste des abréviations :

<b><i>UML:</i></b>	Unified Modeling Language
<b><i>MIT:</i></b>	Massachusetts Institute of Technology
<b><i>AIML:</i></b>	Artificial Intelligence Markup Language
<b><i>API:</i></b>	Application Programming Interface
<b><i>FAQ:</i></b>	Frequently Asked Questions
<b><i>ReLU:</i></b>	Rectified Linear Unit
<b><i>CNN:</i></b>	Convolutional Neural Networks
<b><i>RNN:</i></b>	Recurrent Neural Networks.
<b><i>LSTM:</i></b>	Long short-term memory
<b><i>GRU :</i></b>	Gated Recurrent Unit
<b><i>TALN :</i></b>	traitement automatique du langage naturel
<b><i>NLP:</i></b>	Natural Language Processing
<b><i>CBOW:</i></b>	Continuous Bag of Words
<b><i>PHP:</i></b>	Hypertext Preprocessor
<b><i>JWT:</i></b>	JSON Web Token
<b><i>JSON :</i></b>	JavaScript Object Notation

# **Chapitre I**

## **Présentation du cadre de projet**

### **I. Présentation de l'organisme d'accueil :**

#### **1. Présentation de l'école**

ENSAM (Ecole Nationale Supérieure des Arts et des Métiers) est une école d'ingénierie basée à Casablanca et à Meknès. Visant à former des ingénieurs d'État, l'instruction qui y est dispensée couvre les volets théoriques et pratiques pour une insertion optimale dans le monde du travail.

Le cursus ENSAM Casablanca est composé de deux années préparatoires et de trois années de formation proprement dite. Le système pédagogique de l'école repose sur les unités scolaires, agencées en semestre. En parallèle, à ENSAM Casablanca vous pourrez participer à beaucoup d'activités qui vous formeront sur le plan professionnel et personnel. Des stages d'application vont venir parfaire l'ensemble de votre apprentissage et vous confronter aux réalités travail industriel.

#### **2. Histoire**

L'École nationale supérieure d'arts et métiers - Casablanca a été créée en 2010, et a ouvert ses portes en septembre 2010, et implantée à l'Université **HASSAN II**.

#### **3. Missions**

L'ENSAM Casablanca a pour mission :

- Former des ingénieurs d'État Arts et Métiers ;
- Développer une activité de recherche à l'aide des moyens techniques dont dispose l'école.

L'ENSAM de Casablanca vise le développement d'une offre très large de formations. Partant de ce principe, et en incitant et motivant ses étudiants à organiser différents événements (Sociales, professionnels, culturels et sportifs), elle permet à ses futurs lauréats d'acquérir :

- Les compétences techniques et professionnelles de l'ingénieur.
- L'épanouissement de la personnalité.
- L'apprentissage de la responsabilité.
- L'adaptation à un milieu professionnel en constante évolution.

#### **4. Admission à la formation Ingénieur Arts et Métiers de Casablanca**

L'ENSAM présente 3 possibilités d'accès à la formation Ingénieur Arts et Métiers de Casablanca : en 1ère année, en 3ème année et en 4ème année.

### 5. Filière

Les filières accréditées (Année universitaire 2019/2020) au cycle d'ingénieur sont les suivantes :

- **Intelligence artificielle et génie informatique :**
  - Option 1 : Intelligence Artificielle & l'Industrie Connectée
  - Option 2 : Ingénierie Informatique
- **Génie industriel :**
  - Option 1 : Management des Systèmes Industriels
  - Option 2 : Gestion de Projets Technologiques
- **Génie mécanique :**
  - Option 1 : Construction d'Ouvrages Métalliques
  - Option 2 : Efficacité et Performance Energétique Industrielle
  - Option 3 : Conception de Produits Nouveaux
- **Management des systèmes électriques intelligents :**
  - Option 1 : Systèmes de Management de l'Energie
  - Option 2 : Systèmes Industriels Connectés

### 6. Administration

L'équipe administrative de L'ENSAM est composée de personnels qui travaillent en étroite collaboration afin de vous fournir une prestation de qualité.

Directeur :

- **MOUCHTACHI Ahmed**

Directeur adjoint chargé des affaires pédagogiques :

- **ABOURRICHE Abdelmjid**

Directeur adjoint chargé de la recherche et de la coopération :

- **LAARAJ Mohamed**

Secrétaire général :

- **BAKHRI Ahmed**

## II. Projet

### 1. Présentation du projet

Le but de ce stage de ce projet considéré comme un stage d'initiation est de concevoir et de réaliser une application Web liée au site web de l'école à base d'un ChatBot qui interagit avec un utilisateur pour introduire notre école ENSAM et dans le domaine d'orientation scolaire.

#### 1.1. Etude de l'existant

L'ENSAM Casablanca ne dispose actuellement d'aucune application web (Panneau dialogueur) qui facilite la communication entre l'utilisateur et un agent artificiel dans le cadre de bonne redirection. Ceci pose beaucoup de problème pour les employés les étudiants de l'école au niveau du temps pour faire cette tâche oralement ou bien manuellement afin de répandre rapidement aux questions posées par l'utilisateur en parfaite concordance avec la thématique choisie. Aujourd'hui, L'ENSAM gère les questions des étudiants ou bien les bacheliers oralement, ce qui augmente le risque d'accélérer la propagation de pandémie 'Covid-19' ou perdre le temps et les efforts de plus de la part des étudiants et les employés.

#### 1.2. Besoins fonctionnels

L'objectif de l'Analyse fonctionnelle du besoin est de préciser les fonctions de service attendues à partir des prestations définies. Cette étape est primordiale pour une bonne réalisation du projet. L'application à réaliser doit répondre aux besoins fonctionnels suivants :

L'objectif de l'Analyse fonctionnelle du besoin est de préciser les fonctions de service attendues à partir des prestations définies. Cette étape est primordiale pour une bonne réalisation du projet.

L'application à réaliser doit répondre aux besoins fonctionnels suivants :

- La consultation, l'ajout et la modification des réponses proposées par le bot ;
- Affichage des informations par thèmes ;
- Détermination avec précision les réponses souhaitées ;
- Optimisation de la durée de répondre aux questions demandées par l'utilisateur ;

#### 1.3. Besoins techniques

##### Performance :

-L'application doit offrir aux utilisateurs des réponses fiables et correctes

- L'application doit être présentable
- L'application doit assurer l'intégrité et la cohérence de données (pas de conflits).

##### Ergonomie :

- ❖ L'application doit offrir une interface conviviale qui facilite l'utilisation et réduit le temps d'apprentissage du fonctionnement.
- ❖ L'utilisateur de l'application doit accéder à l'information rapidement : le temps de recherche et de réponse doit être minimal.

Confidentialité : La manipulation des questions, et l'envoi des réponses devrait passer en toute sécurité.

2. Planification de projet

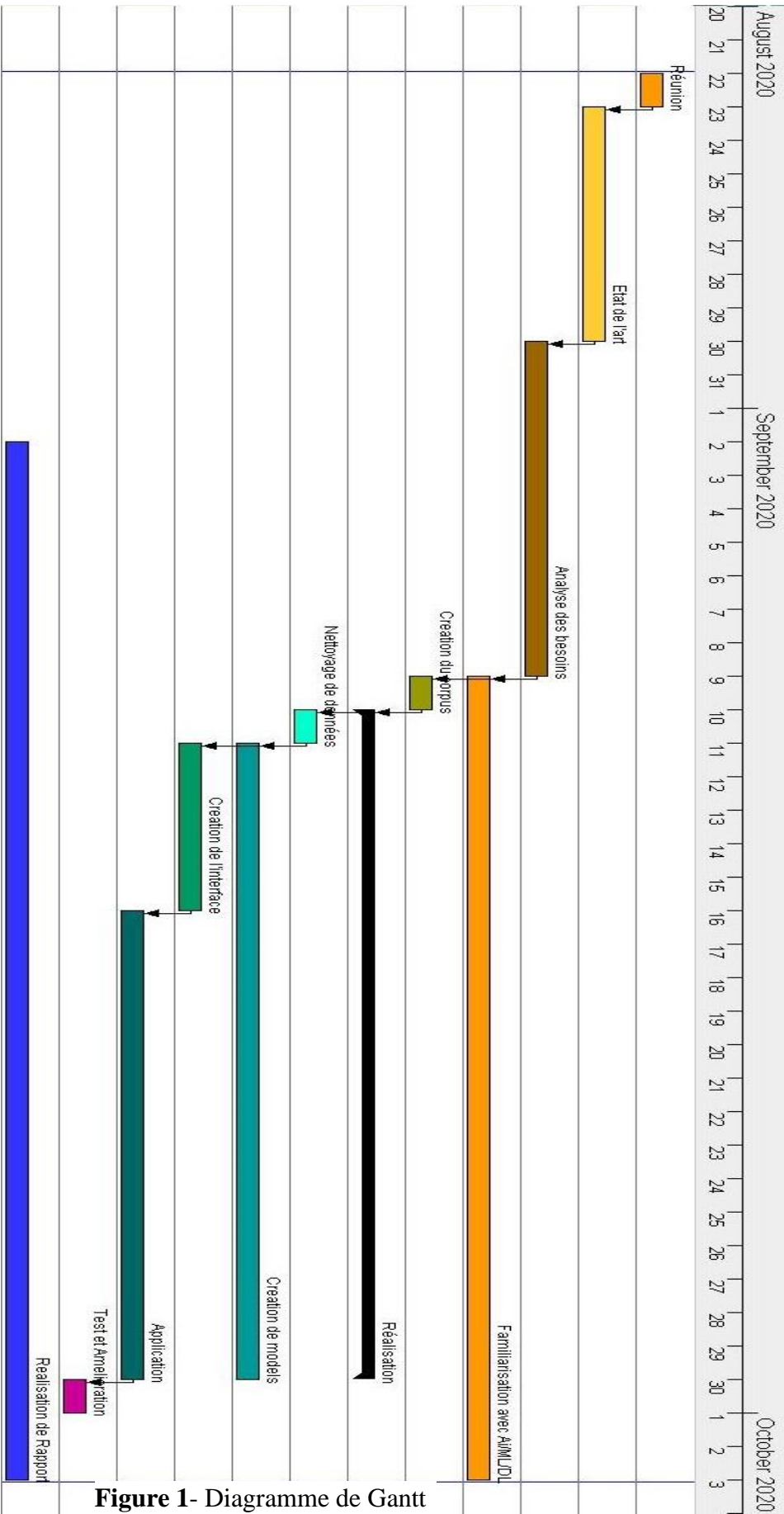


Figure 1- Diagramme de Gantt

### 3. Conception

Dans cette partie, nous découvrons la plate-forme proposée pour ce projet, à l'aide des diagrammes UML, on va bien détailler notre besoin et présenter les différents cas d'utilisation, l'architecture de notre système et les technologies utilisées, une abstraction du fonctionnement des modèles par le diagramme de classes et d'objets.

#### 3.1. Description de cahier de charge de l'application

Pour mieux répondre aux exigences fixées par le client (dans notre cas l'encadrant). Le projet vise à donner aux futurs utilisateurs la possibilité de poser des questions spécifiques, et de générer automatiquement la réponse à l'aide des méthodes d'apprentissage profond. L'adoption d'une telle approche dans la création des systèmes de conversation a ses inconvénients (les réponses générées peuvent être fausses). Les exigences couvrent ce problème en demandant d'impliquer les utilisateurs dans l'amélioration de la précision du système avec des critiques et des suggestions et signalisations.

Pour concrétiser ces besoins, nous avons proposé une application web liée au site de l'école appelée "BotENSAM".

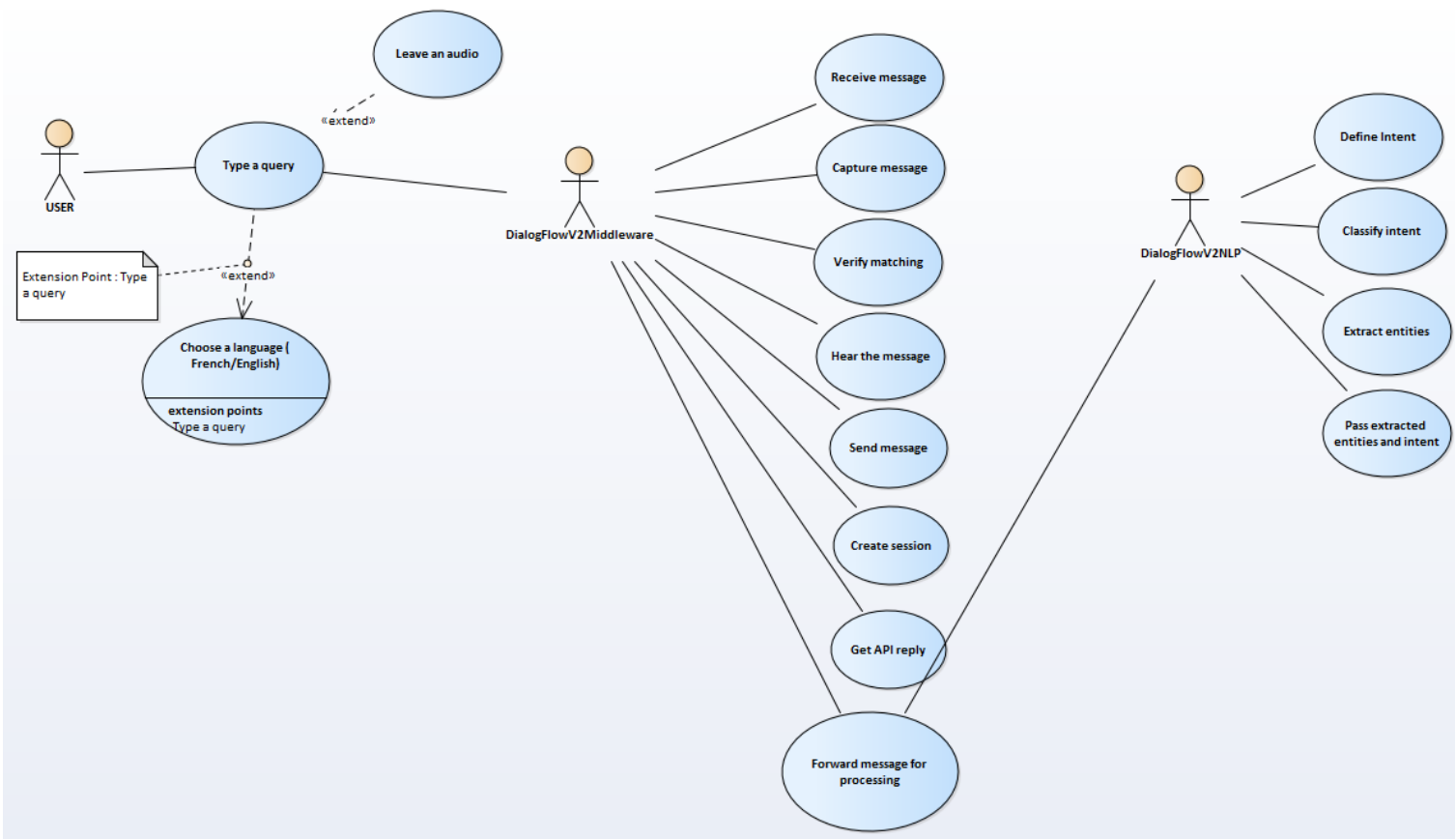
L'ajout, la modification et l'amélioration de chatbot doivent être effectués sans avoir besoin de mise à jour de tout le site. Pour ce faire, nous avons construit notre application avec une architecture (**Figure-17**) basée sur un Rest Api, réalisée avec la bibliothèque de python Flask, et une application web liée au site de l'école.

#### 3.2. Analyse des cas d'utilisation

Le diagramme de cas d'utilisation ci-dessus nous montre l'interaction entre l'utilisateur et le bot.

Composé de plusieurs acteurs (primaires et secondaires)

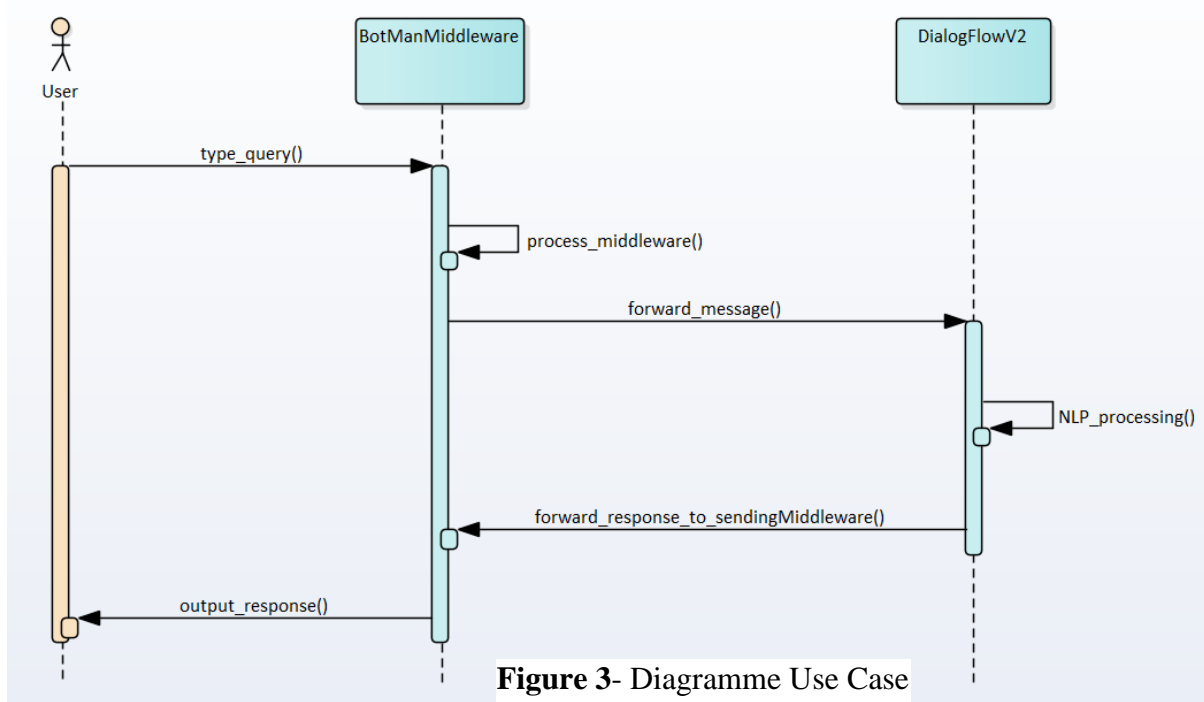
- **USER** : L'utilisateur ne fait que d'écrire des messages soit écrit soit vocal.
- **DialogFlowV2Middleware** : Acteur secondaire ; Reçoit le message envoyé par l'utilisateur et le capte et l'envoie au NLP algorithme de Dialogflow pour faire les traitements nécessaires avant de re-envoyer une réponse. Il peut aussi vérifier le pattern d'un message ou écouter un message vocal...
- **DialogFlowV2NLP** : Acteur secondaire ; Ce système permet d'entraîner le modèle, faire la reconnaissance des entités nommées, étiquetage partie du discours et le plus important la classification du texte...



**Figure 2-** Diagramme de cas d'utilisation

## 3.3. Analyse des interactions

Lorsque l'utilisateur envoie un message (requête), le middleware traite la demande reçue puis invoque l'algorithme de Dialogflow où s'exécute le NLP qui envoie la réponse correspondante au middleware qui finalement la re-envoie à l'utilisateur.



**Figure 3-** Diagramme Use Case



### 3.4. Modélisation orienté objet

La modélisation orientée-objet des systèmes informatiques est l'étape la plus importante dans le développement de ces systèmes qui travaillent avec ce paradigme de programmation, plusieurs facteurs doivent être respectés lors de l'élaboration du modèle :

- ❖ **Flexibilité et l'évolutivité** : le bon modèle est celui qui prévoit les prochaines modifications et laisse la porte ouverte aux nouveaux acteurs et cas d'utilisations.
- ❖ **Abstraction** : Chaque élément significatif de fonctionnalité d'un programme doit être implémenté à un seul endroit dans le code source. Lorsque des fonctions similaires sont exécutées par des morceaux de code distincts, il est généralement avantageux de les combiner en un seul en faisant abstraction des différentes parties.
- ❖ **Single responsibility principle** : Chaque module ou classe devrait avoir la responsabilité d'une seule partie des fonctionnalités fournies par le logiciel. Ceci est pour réduire la complexité du système et la dépendance entre les modules au maximum, et faciliter le débogage aux programmeurs.

Dans le diagramme de classe qu'on va proposer, on a pu respecter tous ces principes, ce qui a rendu l'écriture du code très facile et très collaborative.

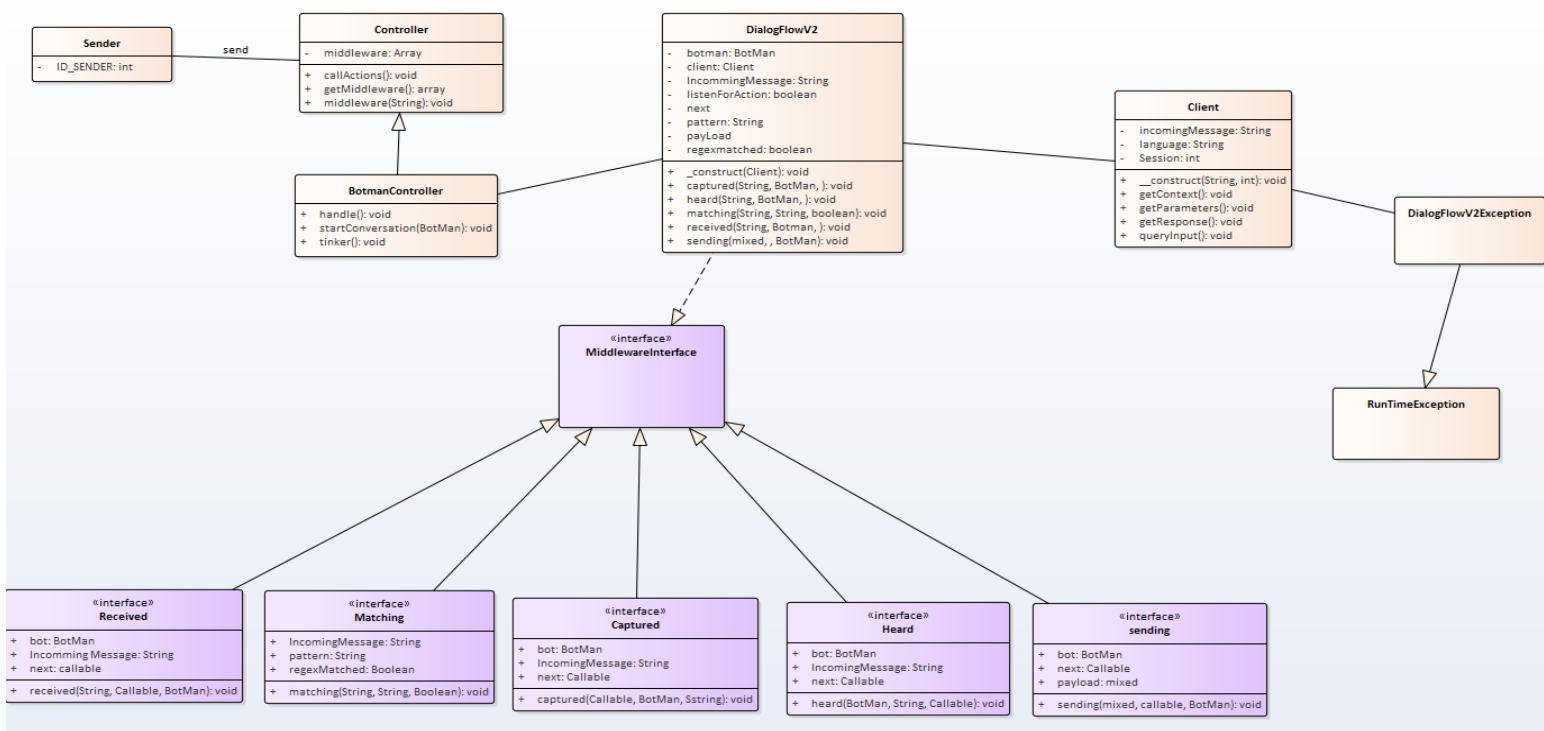


Figure 4- Diagramme de classes

Le diagramme de classe de notre projet admet plusieurs dépendant entre eux, notons par exemple le Controller de Botman qui fait appel à la classe « DialogflowV2 ». Cette dernière implémente l'interface des middlewares qui est subdivisé en plusieurs sous-interfaces (chaque type de middleware à sa propre interface), maintenant le Controller de Dialogflow peut redéfinir les fonctions puis l'utiliser dans le Controller de Botman.

# Chapitre II

## I. Etat de l'art et l'apprentissage profond

### 1. Etat de l'art des ChatBots

Chatbot est une technologie largement utilisé dans plusieurs domaines, allant du support technique au divertissement, comme le robot de recommandation des films. Ils sont fréquemment utilisés pour le service client de base et les systèmes de marketing sur les plateformes de réseautage social et les clients de messagerie instantanée tels que Telegram ou Messenger.

Dans ce chapitre on définit ce terme, et on prend en détails l'histoire et la théorie en arrière-plan des chatbots, les types, et les architectures.

#### 1.1.ChatBot : définition

Selon le Dictionnaire Oxford, un chatbot est défini comme suit :

chatbot(n.): *A computer program designed to simulate conversation with human users, especially over the Internet.*

Donc un Chatbot, également connu sous "Chatter Bot" ou Bot, est un programme informatique qui vise à imiter une conversation humaine, ce service est motorisé par des règles et de l'intelligence artificielle, est accessible via une interface de chat (interface utilisateur).

Le principe sous-jacent de chaque chatbot est d'interagir avec un utilisateur humain (dans la plupart des cas) via des messages texte et de réponse à l'utilisateur de manière appropriée.

#### 1.2. Une brève histoire de ChatBot

Aujourd'hui, selon Apple, plus 44.5% des smartphones dans les Etats-Unis ont utilisé l'assistant virtuel "SIRI", D'où vient cette montée dans l'utilisation des chatbots ?

Le premier qui a introduit la notion de l'imitation du comportement humain est le mathématicien anglais, "**ALAN TURING**" qui est considéré par tous les informaticiens comme le parrain de cette science. Dans son article "COMPUTING MACHINERY AND INTELLIGENCE" [\[1\]](#), TURING a posé la question pour savoir est-ce que les machines peuvent penser, il a reformulé cette question sous forme d'un jeu, où la machine gagne s'elle peut faire ce que nous (en tant qu'entités pensantes) pouvons faire ? Il a appelé ce jeu "**THE IMITATION GAME**".

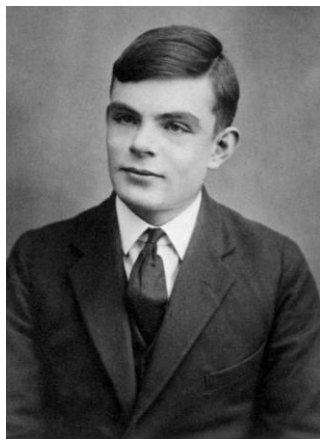
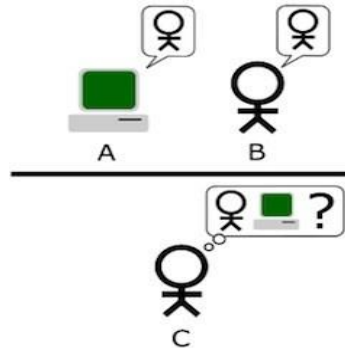


Figure 5- Alan Turing (1912 - 1954)

De cet article, le test de TURING est né, c'est un critère d'intelligence, qui vérifie si un programme informatique peut se comporter comme un humain dans une conversation écrite, bien qu'on ne puisse pas différencier entre l'homme et la machine. Dans ce test, un joueur essaie de déterminer laquelle des autres conversations est écrite par un ordinateur.



**Figure 6-** Test de TURING

En 1966, le chatbot passe pour la première fois à la concrétisation avec la création du chatbot "**ELIZA**", c'est le premier programme informatique de traitement du langage naturel créé en 1966 au *MIT Artificial Intelligence Laboratory* par « *Joseph Weizenbaum* ». Le but de la création d'un tel Bot était la super-facilité de la communication entre l'homme et la machine, **Eliza** est basée sur une méthodologie appelée "Pattern Matching" et de substitution qui a donnée aux utilisateurs une illusion de compréhension de la part du programme, mais n'avait pas de cadre intégrée pour contextualiser les événements.

**Eliza** était un ChatBot qui joue le rôle d'un psychiatre, utilisée pour aider les patients à raconter leurs souffrances. Le scientifique de Stanford "**KENNETH COLBY**" avait une autre idée, en 1972, l'année où le film "**LE PARRAIN**" est sortie, Colby méta ses derniers retouches sur un programme qui simule comment un schizophrène paranoïde pense, il lui a donnée le nom **PARRY**. Les gens le considéraient comme le premier patient de la thérapeute **Eliza**.

Après le succès des deux premiers chatbots textuels, **JABBERwACKY** est devenu le premier chatbot à simuler la voix humaine. Le but de sa création était de permettre au chatbot de passer le test de TURING. Sur les traces de **JABBERWACKY**, le Chatbot nommé **Dr SBAITSO** a été créé en 1992. Le bot avait une interface utilisateur où le **Dr SBAITSO** a imité réponses d'un psychologue.

Artificial Linguistic Internet Computer Entity aka **ALICE**, est un chatbot développé par Richard Wallace. Il s'agit d'un programme de personnalité basé sur le langage de balisage de l'intelligence artificielle (AIML) qui a remporté le prix **Loebner** dans la catégorie l'ordinateur le plus humain. Ce prix est décerné lors des concours annuels de test de Turing. **ALICE** l'a remporté en 2000, 2001 et 2004.

Au début des années 2010, une montée des assistants virtuels a commencé tels que **Siri** d'Apple, **Cortana** de Microsoft, **Google assistant** de Google, **Alexa** d'Amazon et d'autres encore. Ces agents ont introduit sur le terrain le concept de conversation ainsi que le dialogue axé sur les objectifs. Un autre événement majeur dans le domaine des chatbots a été la sortie de Messenger Platform par Facebook Messenger en 2016 ce qui a permit la création d'agents conversationnels.

### 1.3.Types et Architectures des Chatbots

Les chatbots peuvent être classés en différents types en fonction des connaissances dont

ils disposent ou auxquels ils accèdent, de leur niveau d'interaction, de leur méthode de génération de réponse ou de l'objectif qu'ils souhaitent atteindre. La figure-7 illustre la classification d'un chatbot selon les quatre paramètres [2] : domaine de connaissances, service fourni, objectifs et génération de réponse.

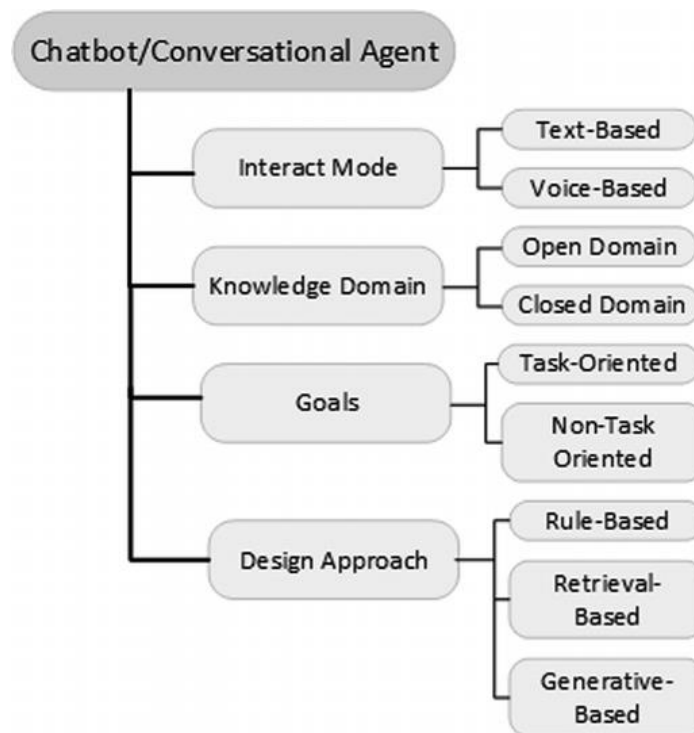


Figure 7- Classification d'un chatbot [2]

- ❖ **Domaine de connaissances** : Selon le domaine de connaissances, les chatbots peuvent être classés en deux catégories :
  - ➔ **Domaine ouvert** : Les bots peuvent parler des sujets généraux et répondre de manière appropriée.
  - ➔ **Domaine fermé** : Les bots se concentrent sur un domaine de connaissances particulier et ne peuvent pas répondre à d'autres questions qui sont hors de son domaine de connaissances.
- ❖ **Service fourni (Service Provided)** : Selon le service fourni, les bots peuvent être groupés en 3 catégories :
  - ➔ **Interpersonnel** : Les bots qui fournissent des services tels que la réservation de restaurant, la réservation de vol, et qui ne sont pas censés d'être des compagnons de l'utilisateur.
  - ➔ **Intrapersonnel** : Ils sont des compagnons de l'utilisateur, ils l'aident à gérer ses travaux, par exemple : Gérer le calendrier, stocker l'opinion de l'utilisateur, etc.
  - ➔ **Inter-agent** : Des bots qui ont la possibilité de communiquer entre eux.
- ❖ **Objectif (Goal)** : Ce paramètre classe les bots en fonction de l'objectif principal qu'ils vont atteindre :
  - ➔ **Informative** : Ces robots sont conçus pour fournir à l'utilisateur des informations préalablement stockées ou disponibles à partir d'une source fixe. Ils sont basés sur un algorithme de récupération d'informations soit dans la base de données ou bien les page FAQ.
  - ➔ **Conversationnel** : Ces bots discutent avec l'utilisateur, comme un être humain. Leur objectif est de répondre correctement aux questions de l'utilisateur.

- ➔ Basé sur la tâche : Les bots effectuent des tâches telles que : réserver un vol ou aider l'utilisateur à faire des courses dans un magasin. La plupart du temps, les actions nécessaires pour effectuer une tâche sont prédéterminées. Les bots sont intelligents dans le contexte de la demande d'informations et de comprendre la contribution de l'utilisateur.
- ❖ **Méthode de génération de réponse** : Cette classification prend en compte la méthode du traitement des entrées et de la génération des réponses :
  - ➔ Les systèmes intelligents : Les systèmes intelligents génèrent des réponses et utilisent le traitement automatique du langage naturel pour comprendre la requête. Ces systèmes sont utilisés lorsque le domaine est étroit et que de nombreuses données sont disponibles pour former un système.
  - ➔ Systèmes basés sur des règles : Les systèmes basés sur des règles utilisent l'appariement de motifs et s'avèrent rigides. Ceux-ci peuvent être utilisés lorsque le nombre de résultats possibles est fixe et que les scénarios sont imaginables en nombre.
  - ➔ Hybride : Ces systèmes sont un mélange de règles et d'apprentissage automatique.

### 1.3.1. Architecture : Retrieval vs Generative

**Retrieval-based** technique compte sur une large base de données des réponses possibles et les associent aux informations du message de l'utilisateur pour trouver la réponse la plus appropriée. Ces informations peuvent simplement être une expression régulière qui cherche des structures de phrases particulières, ou peuvent être le résultat d'un modèle d'apprentissage automatique. L'avantage de cette approche est que chaque réponse peut être contrôlée, et ainsi éviter les réponses inappropriées. [\[3\]](#)

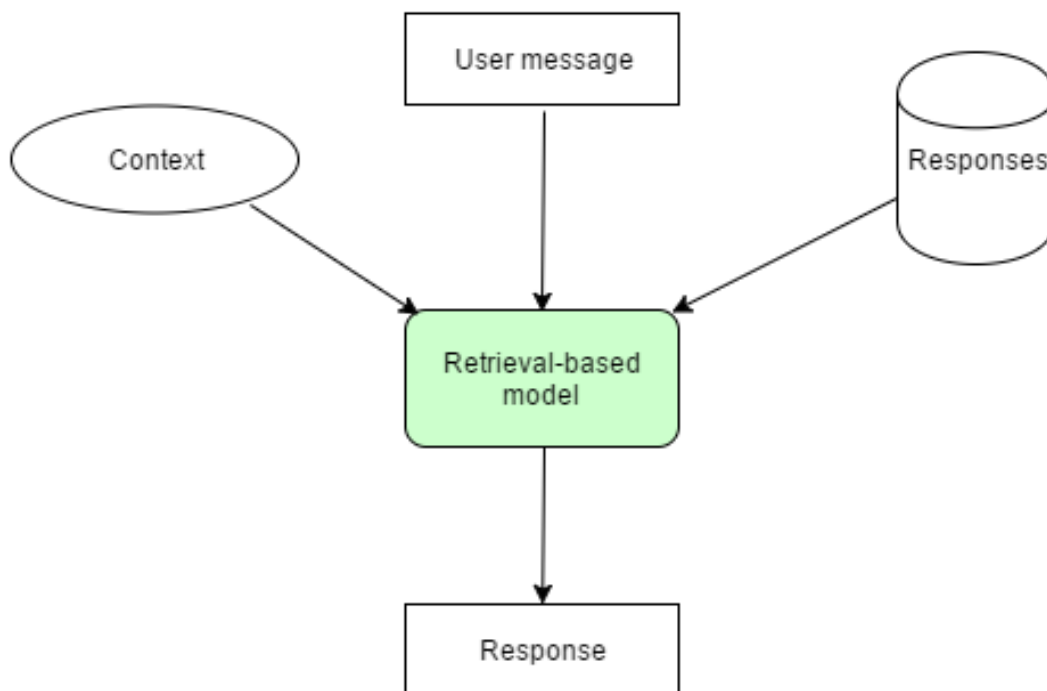


Figure 7- Architecture retrieval

**Generative-based** s'appuient sur des modèles génératifs pour produire de nouvelles réponses sans avoir besoin d'une base de données exhaustive d'exemples. Des nouvelles réponses peuvent être générées facilement à condition que le modèle soit correctement formé.

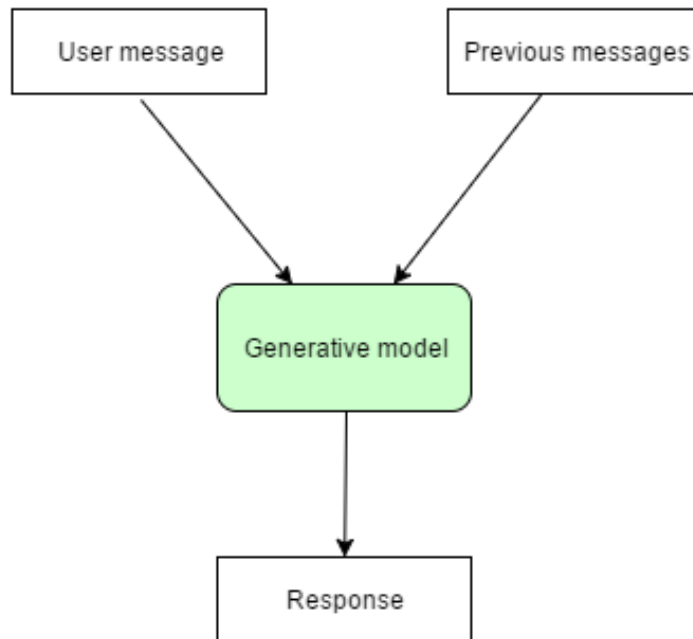


Figure 9- Architecture générative

## 2 L'apprentissage Profond

Le Deep Learning est un ensemble de méthodes d'apprentissage qui visent à modéliser des données avec des architectures complexes combinant différentes transformations non linéaires.

Dans ce chapitre, nous aborderons les bases de l'apprentissage profond.

### 2.1. Réseau de neurones artificiels

Nous traitons dans cette section les réseaux de neurones, qui grâce à elles fonctionne l'apprentissage profond. Afin de comprendre et exceller en matière d'apprentissage profond, il nous est impérativement nécessaire d'acquérir une parfaite maîtrise de ces réseaux.

- Neurone artificiel  
Un réseau de neurones est composé de nombreux neurones artificiels. À l'origine, le neurone artificiel est conçu sur le principe de formation et de fonctionnement des neurones biologiques. Un neurone biologique est connecté à plusieurs autres neurones grâce à un intermédiaire qui s'appelle **synapses** et qui est parmi les constituants du cerveau, il reçoit des signaux (impulsions électriques) par les **dendrites** et envoient l'information par les axones. Un traitement interne de ces signaux conduit à la génération d'une sortie, les sorties peuvent être mises en disposition pour déclencher d'autres neurones ou traverser un nerf pour activer certains muscles. [\[4\]](#)

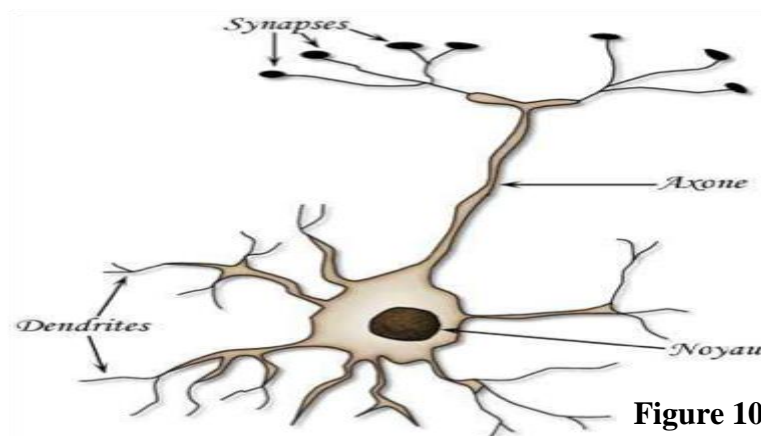
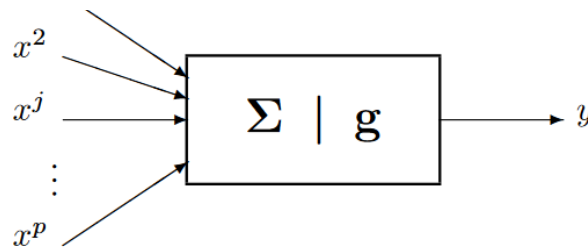


Figure 10- Un neurone biologique [\[5\]](#)

Maintenant, nous allons assimiler le fonctionnement d'un neurone artificiel par un modèle mathématique. Les valeurs d'entrée  $x_1, x_2, \dots, x_n$  sont fournies au neurone. Nous allons schématiser cette entrée comme un vecteur  $\mathbf{X}$ , tel que  $\mathbf{X} = x_1, x_2, \dots, x_n$ . Les entrées vont être pondérées par le vecteur de poids. Le vecteur de poids comporte les éléments  $w_1, w_2, \dots, w_n$ . C'est ce vecteur de poids que nous allons chercher à optimiser ses composantes afin d'aboutir à une sortie optimale. Nous appelons la sortie réelle  $y$ , et la sortie prévue du neurone comme  $\hat{y}$ . En plus des poids, les neurones possèdent un biais (disons  $b$ ), qui est un nombre réel. Afin d'ajouter la non linéarité au réseau on utilise la fonction d'activation. L'entrée de la fonction d'activation s'appelle  $Z$ , qui est égale à  $b + w_1x_1 + w_2x_2 + \dots + w_nx_n$ .

La sortie du neurone pourrait être formulée par [4] :

$$\hat{y} = f(Z) = f\left(b + \sum_{i=1}^n w_i x_i\right) = f(b + W^T \cdot X)$$



**Figure 11.1-** Modélisation Mathématique d'un neurone artificiel [6]

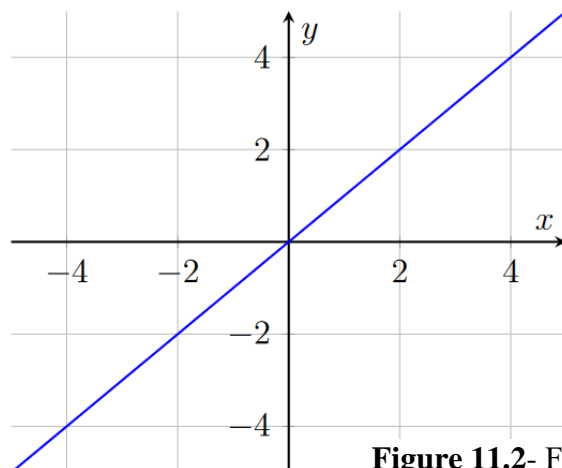
## 2.2. Fonctions d'activation

Il existe plusieurs types de fonctions d'activation. Chacune d'elles dépend de son cas d'utilisation.

### 2.2.1. Fonction d'activation linéaire

C'est une fonction simple de la forme :  $f(x) = ax$  [7]

Généralement, l'entée est transférée à la sortie avec une légère modification ou sans aucune modification. On reste ici dans une situation de proportionnalité.



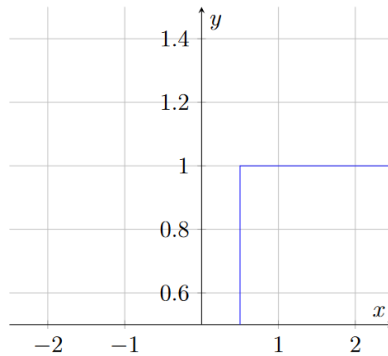
**Figure 11.2-** Fonction d'activation linéaire

### 2.2.2. Fonction d'activation avec seuil

La fonction d'activation avec seuil est définie comme :

$$f(x) = \begin{cases} 1 & \text{si } z \geq 0.5 \\ 0 & \text{sinon} \end{cases} \quad [8]$$

Cette fonction génère une valeur de 1 pour les valeurs entrantes de 0.5 ou plus et de 0 pour toutes les autres valeurs.



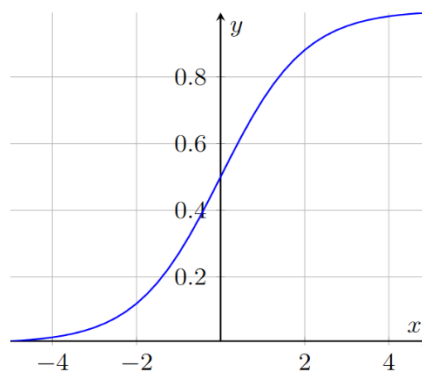
**Figure 11.3-** Fonction d'activation avec seuil

### 2.2.3. Fonction d'activation sigmoïde

La fonction sigmoïde est définie sous la forme mathématique suivante :

$$f(x) = \frac{1}{1 + e^{-x}}$$

Elle prend une valeur réelle en entrée et la transforme en une valeur réelle de sortie comprise entre 0 et 1



**Figure 11.4-** Fonction d'activation sigmoïde

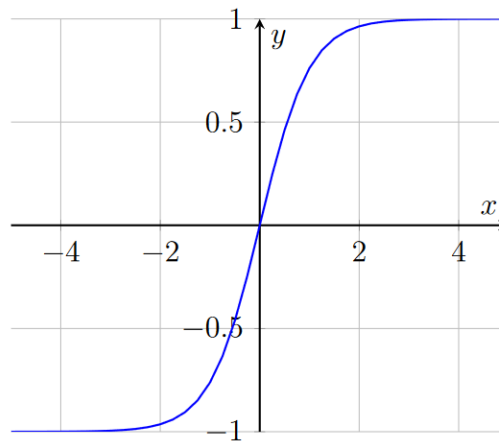


### 2.2.4. Fonction d'activation tangente hyperbolique

La fonction Tanh est également appelée "tangente hyperbolique" :

$$f(x) = \tanh(x)$$

Cette fonction ressemble à la fonction Sigmoid. La seule différence est que la fonction Tanh produit un résultat compris entre -1 et 1. La fonction Tanh est en général préférable à la fonction Sigmoid car elle est centrée sur zéro. Les grandes entrées négatives tendent vers -1 et les grandes entrées positives tendent vers 1.



**Figure 11.5-** Fonction d'activation tangente hyperbolique

### 2.2.5. Unités linéaires rectifiées

La fonction d'activation nommée unité de rectification linéaire (ou ReLU) est souvent un peu plus efficace qu'une fonction lisse de type sigmoïde, tout en étant bien plus simple à calculer.

$$f(x) = \max(0, x)$$

Cette fonction d'activation est non saturante et augmente considérablement la convergence du réseau. Mais la fonction ReLU n'est pas parfaite. Si la valeur d'entrée est négative, le neurone reste inactif, ainsi les poids ne sont pas mis à jour et le réseau n'apprend pas.



**Figure 11.6-** Unités linéaires rectifiées

### 2.2.6- Fonction d'activation Softmax

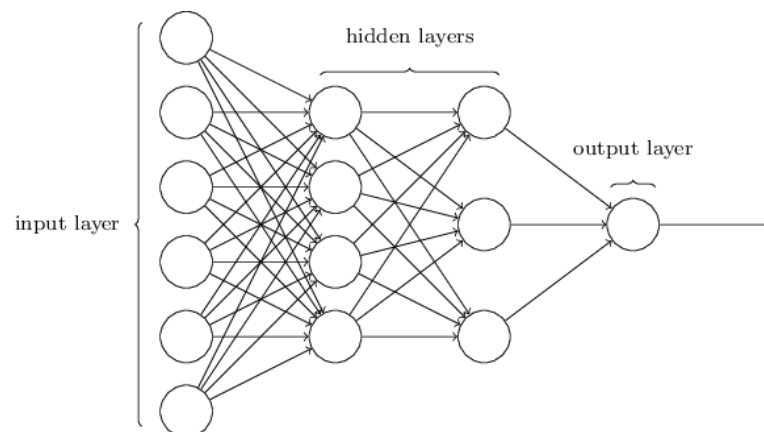
La dernière fonction d'activation que nous examinons dans cette partie est la fonction d'activation softmax. Elle se trouve généralement dans la couche de sortie d'un réseau de neurones. Cette fonction est surtout utilisée dans le cas des tâches de multiclass classification. Elle force la sortie du réseau de neurones à représenter la probabilité que l'entrée appartient à une classe définit. Pour obtenir cette probabilité, nous utilisons cette équation :

$$f(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Avec  $i$  représente l'indice du neurone de sortie en cours de calcul, et  $j$  représente les index de tous les neurones du niveau. La variable  $z$  désigne le vecteur des neurones de sortie. Il est important de noter que l'activation **softmax** est calculée différemment des autres fonctions d'activation. Lorsque **softmax** est la fonction utilisée, la sortie d'un neurone dépend des autres neurones de sortie.

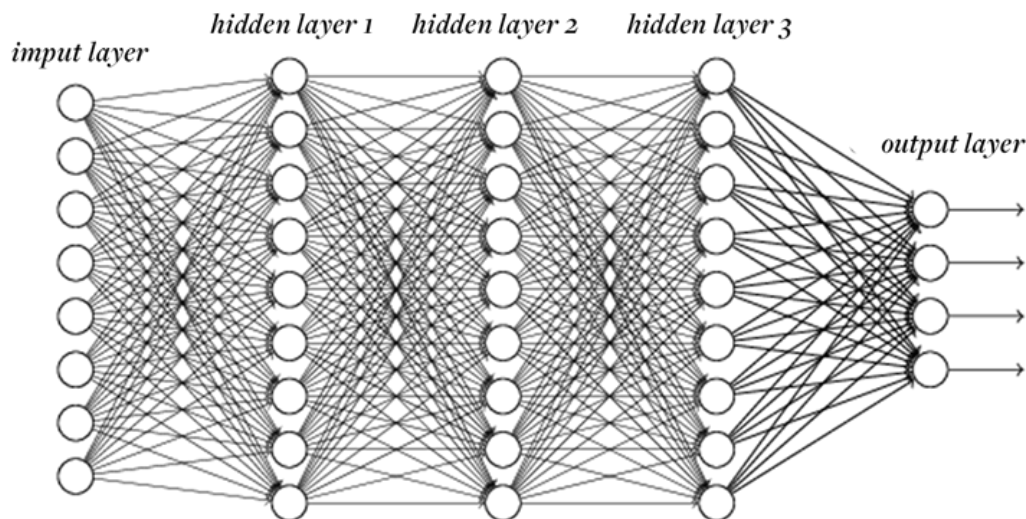
### 2.3. Réseau de neurones

Comme dans les cerveaux humains, les neurones isolés ne sont pas très puissants, mais lorsque plusieurs neurones se combinent, le fonctionnement du cerveau devient plus fort. De même, pour construire un réseau de neurones artificiels. Un réseau de neurones artificiel comporte différentes couches pour donner un sens aux informations qu'ils reçoivent comme entrées. La couche d'entrée reçoit diverses formes d'informations du monde extérieur. Ce sont des caractéristiques de l'entrée que le réseau vise à traiter ou à analyser. Les couches masques permettent la transition des données de l'unité d'entrée vers l'unité de sortie.



**Figure 12.1-** Un réseau de neurones

**Réseau de neurones profonds** Un réseau neuronal profond est un réseau avec un certain niveau de complexité. Habituellement, un réseau de neurones est considéré comme profond, s'il est composé d'au moins quatre couches (c'est-à-dire deux couches cachées, une couche de sortie, et la couche d'entrée pour avoir les quatre). Le nombre de neurones de chaque couche peut être différent. [\[9\]](#)

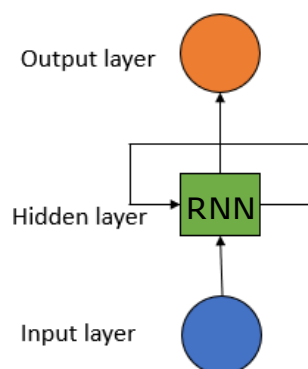


**Figure 12.2-** Un réseau de neurones Profond

Dans les prochaines sections deux types de réseaux de neurones sera présenté à savoir les réseaux de neurones convolutif (CNN) et les réseaux de neurones récurrents (RNN).

### 2.3.1 Les réseaux de neurones récurrents (RNN)

Les RNN ou "les Réseaux de Neurones Récurrents" sont un type spécial des réseaux de neurones spécifiquement adaptés aux données séquentielles. Ils sont appelés récurrents car ils effectuent la même tâche pour chaque élément de la séquence d'entrée, la sortie dépendant des calculs précédents et la couche cachée à l'instant  $t$  dépend de l'entrée à l'instant  $t$  et aussi de la couche cachée à l'instant  $t-1$ , ils ont la capacité de conserver les informations des données précédentes comme mémoire temporelle. Ils peuvent être considérés comme une chaîne de modules répétitifs de réseau de neurones. [\[10\]](#)



**Figure 13.1-** Réseau de neurones récurrent

En raison de leur mémoire temporelle, ils sont très utiles pour les données séquentielles ou chaque élément de la séquence est liée aux autres. Donc en théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues. Nous pouvons utiliser les réseaux de neurones récurrents pour résoudre les problèmes liés aux :

- Données avec un aspect chronologiques
- Données texte
- Données audio

## Chapitre II : Etat de l'art et l'apprentissage profond

Problème de dépendance à long terme Les informations transitent par le réseau de neuronal des neurones d'entrée aux neurones de sortie, tandis que l'erreur est calculée et propagée à travers le réseau pour mettre à jour les poids.

### Exploding gradient :

Le problème se produit principalement lorsque les paramètres du réseau et les hypers paramètres ne sont pas correctement définis. Les paramètres pourraient être des poids et des biais tandis que les hyperparamètres pourraient être le taux d'apprentissage, le nombre d'époques, le nombre de lots, etc. Dans les réseaux profonds de type RNN, les gradients d'erreur peuvent s'accumuler lors d'une mise à jour ce qui provoque une accumulation des gradients. À l'extrême, les valeurs des poids peuvent devenir grandes qu'elles débordent et donnent des valeurs de fausse. L'explosion se produit par croissance exponentielle en multipliant à plusieurs reprises les gradients à travers les couches du réseau qui ont des valeurs supérieures à 1,0.

### Vanishing gradient :

Maintenant, même si les RNN sont très puissants, ils souffrent d'un problème de Vanishing gradient ou bien gradient de fuite, En générale, l'ajout des couches masquées rendra le réseau capable d'apprendre des fonctions arbitraires plus complexes et, par conséquent, de mieux prédire les résultats futurs. Mais, pendant le calcul des gradients, il a tendance à devenir de plus en plus petit à mesure que nous continuons à reculer dans le réseau.

Donc avec la relation de mise à jour des poids

$$W_{new} = W_{old} - gradient * n$$

Le gradient sera très faible et Il n'y aura pas de différence entre  $W_{new}$  et  $W_{old}$ . Cela pose le problème de Gradient de fuit.

Ces deux problèmes peuvent être réduits en utilisant les unités de mémoire (LSTM) et des structures neuronales de type gated (GRU).

### Les réseaux de mémoire à court terme (LSTM) : [11]

Les réseaux LSTM sont un type spécial des RNN qui peuvent résoudre le problème des d dépendances à long terme en raison de sa structure interne. Alors qu'un RNN de base est composé d'une seule couche d'opération, les réseaux LSTM en utilisent quatre. En interne, ils peuvent effectuer trois opérations : oublier les informations, mettre à jour les informations et générer des informations.

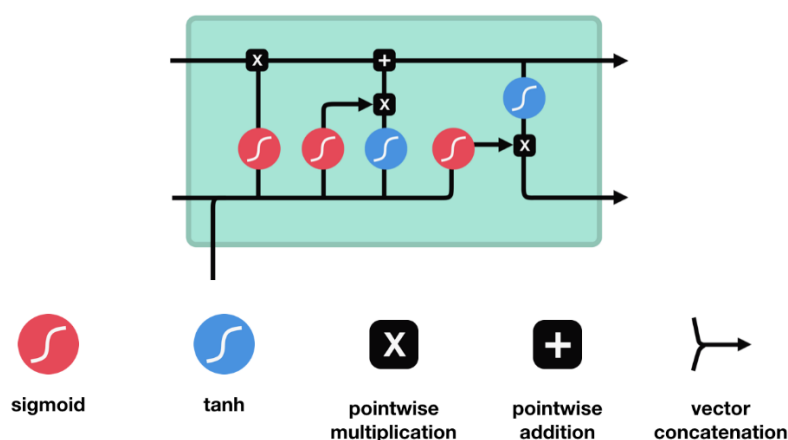


Figure 13.2- Cellule LSTM

Les réseaux neuronaux LSTM sont des RNNs qui permettent de limiter le Vanishing Gradient problème par un ensemble de 3 portes (gates).

### Point clé : “cell state”

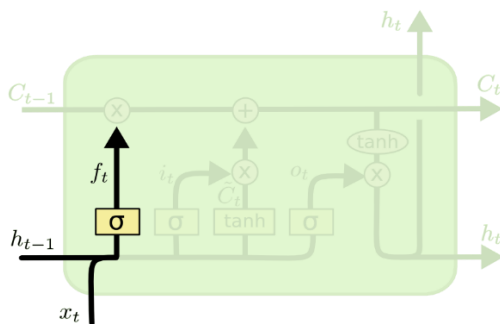
La ligne horizontale dans le schéma correspond à la propagation quasi à l'identique du vecteur d'état de cellule (cell state), ce qui assure la propagation de l'information initiale.

### Que signifie “portes” ?

Chaque “porte” correspond à un petit réseau neuronal avec sigmoïde comme fonction d'activation dont le but est de ramener les valeurs des vecteurs en entrée de la porte entre 0 et 1, puis en multipliant élément par élément (*element wise*) ces valeurs inférieures à 1 avec un autre vecteur, il s'agit de définir les valeurs de cet autre vecteur que l'on souhaite “laisser passer à travers la porte” et inversement les valeurs que l'on souhaite lifter (pour continuer à filer la métaphore, il s'agit en fait d'ouvrir plus ou moins la porte de manière personnalisée à chaque information/valeur d'un vecteur).

### Les 3 portes d'un LSTM : [12]

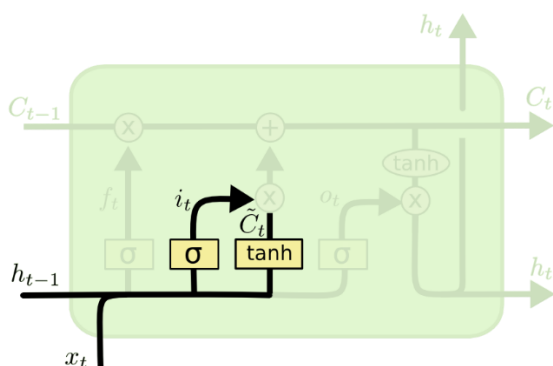
**Forget gate layer** : cette première porte apprend ce qu'elle doit oublier du vecteur d'état de cellule (cell state) en multipliant ce vecteur élément par élément par un vecteur de valeurs entre 0 et 1 obtenu suivant la fonction  $f_t$  suivante (transformation via un réseau neuronal linéaire à 1 couche avec fonction d'activation sigmoïde du vecteur obtenu par concaténation entre le vecteur d'état caché et le nouveau vecteur d'entrée).



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Figure 13.2.1- forget gate layer

**Input gate layer** : à présent que le LSTM a éliminé ce qu'il ne souhaitait pas se souvenir, il doit décider de ce qu'il veut garder, c'est-à-dire ajouter au vecteur d'état de cellule. Le schéma suivant explicite comment calculer ce qui doit lui être ajouté.



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Figure 13.2.2- Input gate layer

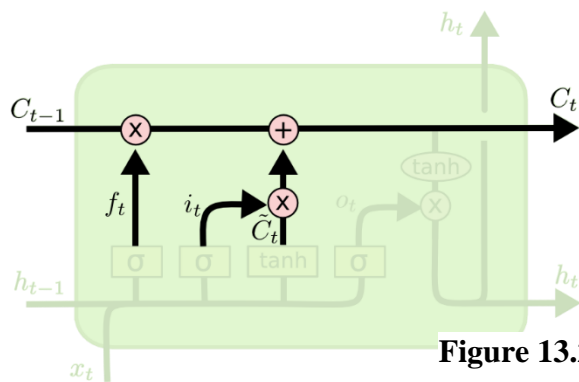
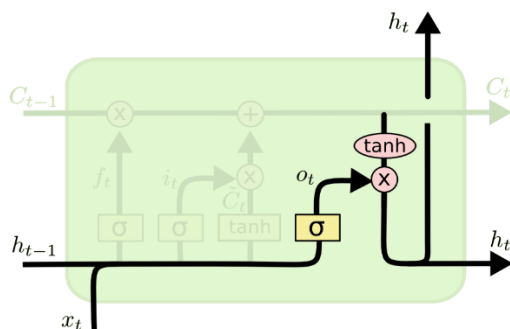


Figure 13.2.3- Mise à jour du vecteur d'état de cellule du LSTM

**Output gate :** il s'agit alors de calculer la sortie du LSTM (le nouveau vecteur d'état caché), ce qui revient en fait à sélectionner via une multiplication élément à élément une partie du vecteur d'état de cellule.



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Figure 13.2.4- output gate du LSTM

### GRU (Gated Recurrent Units) :

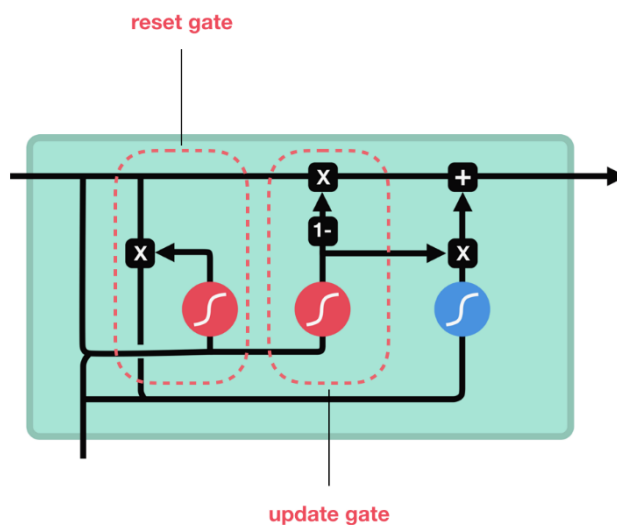


Figure 13.3- Gated Recurrent Unit

Les GRUs sont similaires aux LSTM mais ils n'ont pas de vecteur d'état de cellule et ils ont seulement deux portes (gates), une porte de réinitialisation  $r$  (reset gate) et une porte de mise à jour  $z$  (update gate) :

- La porte de réinitialisation  $r$  détermine la quantité d'information à oublier,
- La porte de mise à jour  $z$  définit la quantité d'information à conserver (elle fonctionne comme la porte forget et input d'un LSTM).

Le vecteur d'état caché calculé va s'écrire sous la forme  $(1-z)*h + z*h(t-1)$  avec les éléments du vecteur  $z$  appris et de valeurs entre 0 et 1 ( $h(t-1)$  est le vecteur d'état caché précédent et  $h$ , le vecteur d'état caché calculé). Il s'agit d'une *weighted average*.

- Les **GRU** s'entraînent plus rapidement et fonctionnent mieux que les LSTM sur moins de données de formation.
- Les **LSTM** nous donnent le plus de capacité de contrôle et donc de meilleurs résultats. Mais vient également avec plus de complexité et de coûts d'exploitation.

### 2.3.2. Réseaux de neurones convolutifs (CNN)

Les "Réseaux de neurones convolutifs" sont un type de réseau neuronal spécialisé pour le traitement de données qui sont formatées sous la forme d'une matrice. Par exemple : les données de séries chronologiques, qui peuvent être considérées comme une matrice 1-D, les images, qui peuvent être considérées comme une matrice 2-D de pixels. [13] L'idée derrière un réseau neuronal convolutif est de filtrer les images avant d'entraîner le réseau neuronal profond, après avoir filtré les caractéristiques de celle-ci peuvent être plus près imminente et on peut ensuite les repérer pour identifier quelque chose

Le nom "réseau neuronal convolutif" indique que le réseau utilise une opération mathématique appelée convolution. Donc un CNN (Convolutional Neural Network) est simplement un réseau de neurones qui utilise la convolution à la place de la matrice initiale dans au moins

Une de ses couches. Ce type de réseaux est composé principalement de deux parties, la partie convolution qui comprend l'opération mathématique "convolution", le "pooling", et le "flattennig", et la partie entièrement connectée.

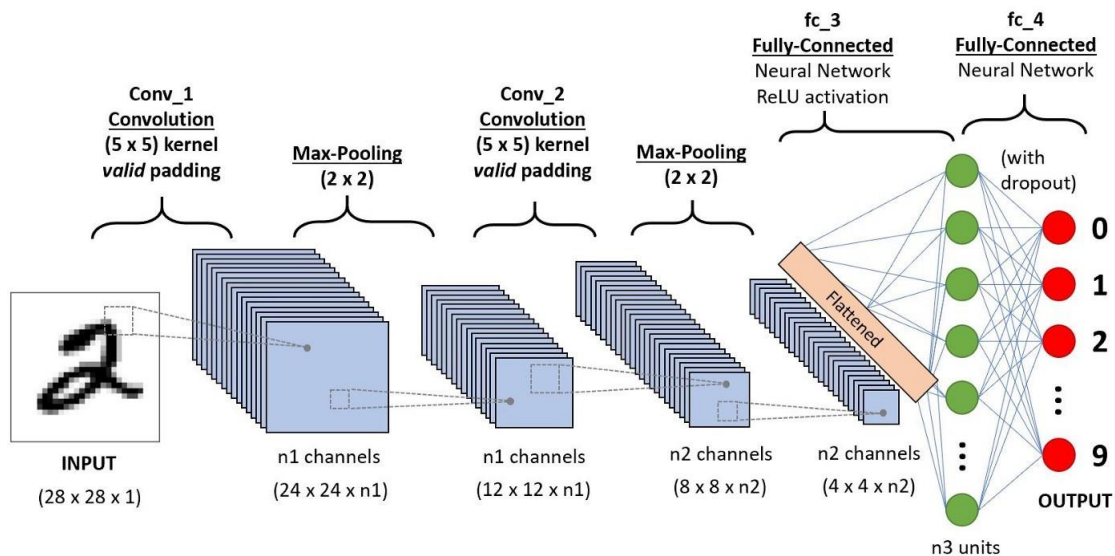


Figure 14.1- Une séquence CNN pour classer les chiffres manuscrits [14]

#### Couche Convulsive

Dans sa forme générale, la convolution est une opération sur deux fonctions  $f$  et  $g$  d'une variable réelle, qui produit une troisième qu'on la note  $S$  [13]

$$s(t) = f * g(t)$$

$S$  est la version modifiée (convoluté) de l'une des fonctions originales

$$s(t) = \int f(a)g(t-a)da$$

La version discrète de cette opération est dénotée par :

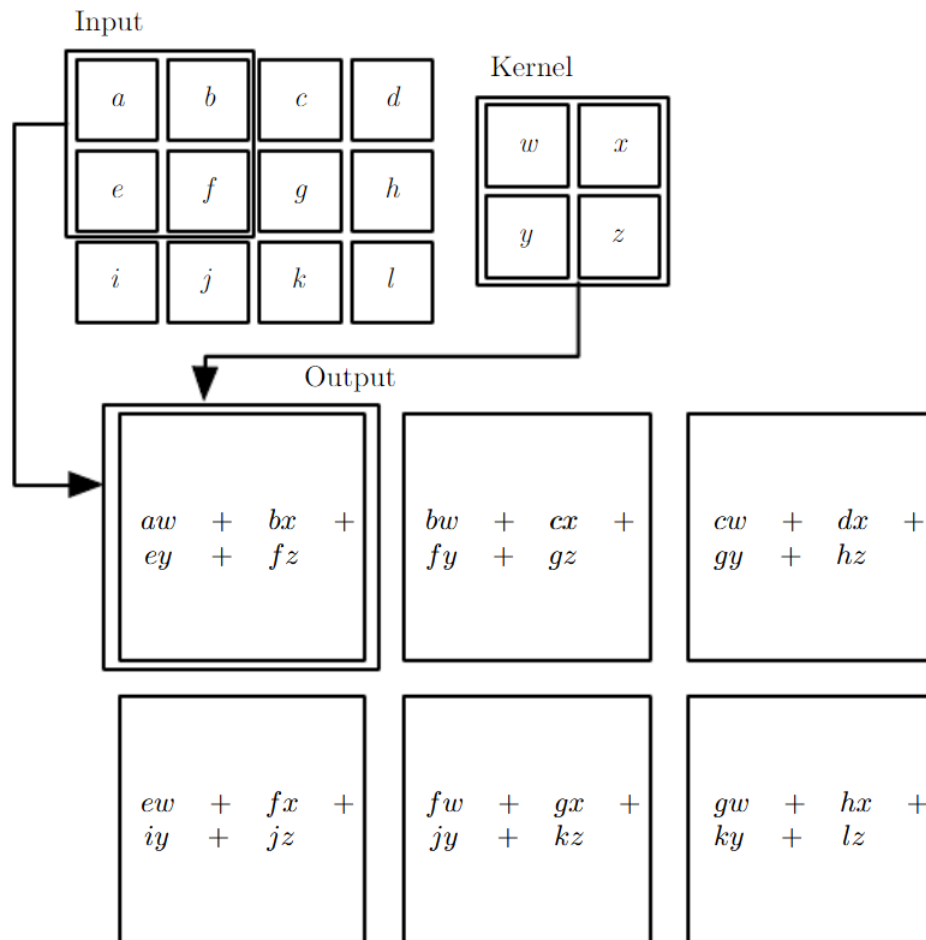
$$f * g(t) = \sum_a f(t)g(t-a)$$



Dans les applications d'apprentissage automatique, dont les signaux d'entrées sont multidimensionnels, on utilise la convolution 2D définie par : et le filtre  $K$  est un noyau de convolution appliqué à un signal 2D (ou une image)  $I$ .

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m,j-n).$$

Le filtre  $K$  est un noyau de convolution appliqué à un signal 2D (ou une image)  $I$ . Comme c'est montré dans la figure.... L'extraction des caractéristiques est faite en glissant un filtre sur l'image 2-D d'entrée.



**Figure 14.2-** Convolution 2D [\[13\]](#)

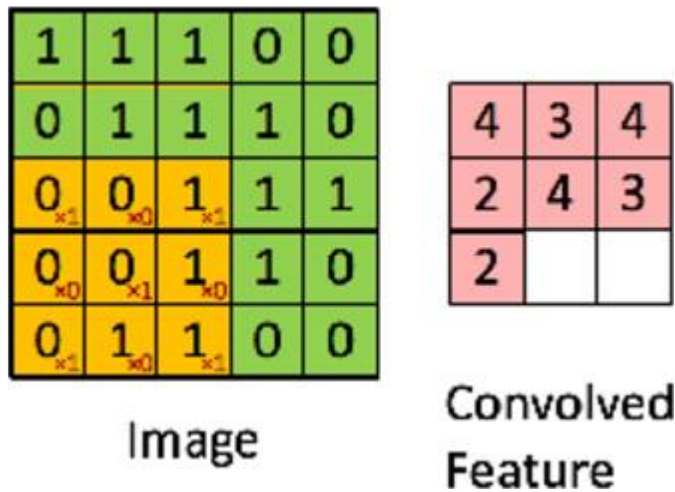
Exemple : Considérons une image 5 x 5 dont les valeurs des pixels ne sont que de 0 et 1, est un filtre  $K$  de taille 3 x 3 :

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1



A convolution de l'image 5 x 5 et de la matrice 3 x 3 est calculée comme indiqué dans la figure.



### Couche de regroupement

Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Elle est presque toujours insérée entre deux couches convolutives successives d'une architecture de réseau de neurones convolutifs pour réduire le surapprentissage. Le pooling peut être de différents types :

- Max Pooling
- Average Pooling
- Sum Pooling

**Couche entièrement connectée** La couche entièrement connectée est la couche finale d'un CNN. Le terme "entièrement connectée" implique que les neurones de cette couche ont des connexions vers toutes les sorties de la couche précédente. Le but de la couche entièrement connectée est d'utiliser les caractéristiques extraites par le réseau pour la classification d'images, la segmentation d'images etc.

## 3. Traitement automatique du langage naturel

Le traitement automatique du langage naturel est une branche de l'intelligence artificielle qui aide les ordinateurs à comprendre, interpréter et manipuler le langage humain, TALN dérivée plusieurs disciplines, y compris la science et le langage informatique dans sa recherche afin de remplir le vide entre la communication humaine et la compréhension de l'ordinateur. Ce chapitre fournit les bases du traitement automatique du langage naturel en relation avec l'apprentissage profond.

### 3.1. Définition

Le traitement automatique du langage naturel (TLN, ou NLP en anglais) est la capacité pour un programme informatique de comprendre le langage humain tel qu'il est parlé. Il fait partie des technologies d'intelligence artificielle. [15] L'objectif du traitement automatique du langage naturel (TALN) est la conception de logiciels, capables de traiter de façon automatique des données linguistiques, c'est-à-dire des données exprimées dans une langue (dite "naturelle"). Ces données linguistiques peuvent être des textes écrits, ou bien des dialogues écrits ou oraux, ou encore des unités linguistiques détaillée inférieure à ce que l'on appelle habituellement des textes (par exemple : des phrases, des énoncées, des groupes de mots ou simplement des mots isolés).

### 3.2. Principaux défis du NLP

Parmi les problèmes les plus fréquents du traitement des langues naturelles, on trouve en premier rang le problème de l'ambiguïté (on dit qu'il y a ambiguïté lorsqu'il y a plus d'une interprétation pour une structure linguistique donnée). L'être humain a une capacité de surpasser les ambiguïtés de la parole en utilisant ses connaissances ce qui n'est pas le cas pour les machines qui ne sont pas encore fiables. Les applications du TALN sont souvent confrontées aux ambiguïtés à tous les niveaux d'analyse.

### 3.3. Domaines d'application du NLP

Parmi les diverses applications possibles, on peut citer principalement :

- La traduction automatique
- Questions answering
- Vérification orthographique
- Le résumé automatique des textes
- Analyse des informations des sites Web, des documents. . .
- . . .

### 3.4. Outils de base du NLP

Parmi les outils de base du NLP, on trouve :

#### Tokenisation

La tokenisation est le processus de segmentation ou de fractionnement d'une chaîne, du texte en une liste de tokens. [16] On peut considérer un mot comme une partie d'une phrase et une phrase comme une partie d'un paragraphe Exemple :

"Hello what is ENSAM Casablanca?" → ["Hello", "what", "is", "ENSAM", "Casablanca", "?"]

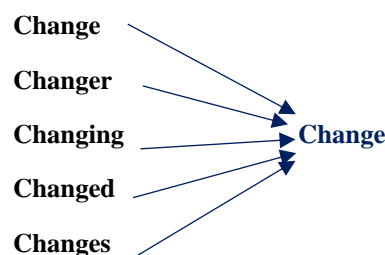
#### Anti-dictionnaire

Ce sont des mots qui se produisent très souvent dans un fichier texte et s'ils sont exclus les performances peuvent être considérablement améliorées. Exemple :

["Hello", "what", "is", "ENSAM", "Casablanca", "?"] → "Hello what is ENSAM Casablanca?"

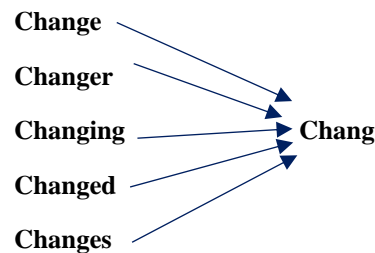
#### Lemmatisation

La lemmatisation est le processus de regroupement des différentes formes fléchies d'un mot afin qu'elles puissent être analysées comme un seul élément. Il regroupe des mots selon leur lemme. [17] Exemple :



### Racination

La racine d'un mot correspond à la partie du mot restante une fois que l'on a supprimé son préfixe et son suffixe, à savoir son radical. Elle est aussi parfois connue sous le nom de stemm d'un mot. Exemple :



### 3.5. Représentation des mots

Cette partie montre la façon dont nous représentons les mots en vecteurs de nombres réels pour être utilisés en tant qu'entrées d'un modèle neuronal.

#### Représentation One-Hot

La représentation One-Hot, L'approche la plus courante pour convertir des mots sous forme de vecteur, chaque vecteur d'un mot contient le chiffre « 1 » à l'index de ce mot et des « 0 » partout ailleurs avec la taille de ce vecteur est « N », N est la taille de notre vocabulaire. [18] Considérons les deux phrases suivantes :

[“what is ENSAM Casablanca?”, “Where is ENSAM Casablanca?”]

Tokénisons les phrases, ignorons la ponctuation donnera un vocabulaire de taille=5

[“what”, “is”, “ENSAM”, “Casablanca”, “Where”]

Chaque mot avec un vecteur One-Hot à 5 dimensions :

	What	is	ENSAM	Casablanca	Where
$V_{\text{What}}$	1	0	0	0	0
$V_{\text{is}}$	0	1	0	0	0
$V_{\text{ENSAM}}$	0	0	1	0	0
$V_{\text{Casablanca}}$	0	0	0	1	0
$V_{\text{Where}}$	0	0	0	0	1

Nous représentons chaque mot comme un vecteur indépendant. Ainsi, cette représentation ne nous donne aucune notion directe de similarité. [19]

#### Matrice de co-occurrence

En utilisant cette technique, nous comptons le nombre de fois que chaque mot apparaît dans une fenêtre d'une taille particulière autour du mot qui nous intéresse. [19] Considérons les deux phrases suivantes :

[“what is ENSAM Casablanca?”, “Where is ENSAM Casablanca?”]

En Utilisant une fenêtre de taille 1 :

	What	is	ENSAM	Casablanca	Where
What	0	1	0	0	0
is	1	0	2	0	0
ENSAM	0	2	0	2	0
Casablanca	0	0	2	0	0
Where	0	1	0	0	0

Alors notre Matrice de co-occurrence est :

$$X = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

### 3.6 Plongement de mot (Word Embedding)

Avant le plongement de mots (Word embedding en anglais), la représentation One-Hot(simples) était la plus couramment utilisée pour la représentation des textes. Cependant, la représentation One-Hot ne capture pas la relation entre les mots. Le plongement de mots est une solution qui permet de pallier ce problème. Le plongement de mots est basé sur l'hypothèse que les mots utilisés dans les mêmes contextes tendent à avoir des significations similaires et donc prennent en compte des relations sémantiques entre les mots. Les méthodes de plongements de mots peuvent être catégorisées en deux types : basées sur le comptage et basées sur les réseaux de neurones. Ces deux types diffèrent lors de la construction des vecteurs de mots ainsi que par le contexte qu'elles prennent en compte. Les méthodes basées sur le comptage utilisent les documents comme contexte et capturent la similarité sémantique entre documents alors que celles basées sur les réseaux de neurones utilisent les mots voisins comme contexte pour détecter la similarité mot à mot. [\[20\]](#)

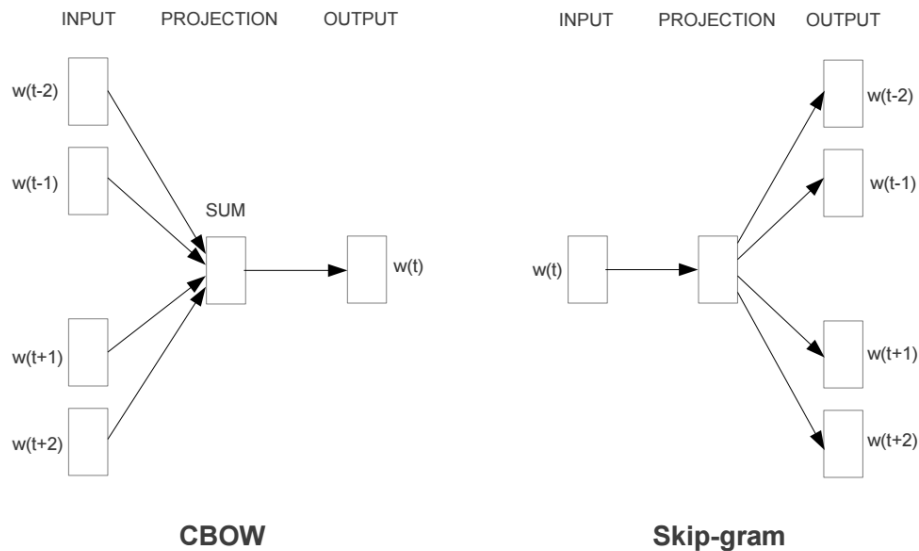
Plusieurs approches neuronales ont été proposées dans la littérature pour la construction des plongements de mots, les plus populaires sont le Word2vec, GloVe et FastText ;

#### Word2vec

Word2vec (Mikolov, Chen et al., 2013 ; Mikolov, Le, Sutskever, 2013) est l'une des techniques les plus populaires pour apprendre le plongement de mot. Les modèles Word2vec utilisent un réseau de neurones à une couche cachée visant à construire une représentation vectorielle des mots.

Il exploite pour cela la notion de contexte d'un mot ; Chaque mot est déterminé grâce aux mots qui apparaissent autour de lui. La représentation du mot est donc apprise en prenant en compte son contexte, permettant ainsi de capturer des liens sémantiques forts entre les mots.

Word2vec est composé de deux modèles en réalité : Sac-de-mots continus CBOW et skip-gram



**Figure 14.3-** Modèles du Word2Vec [\[21\]](#)

### 1. CBOW

Le modèle CBOW prédit le mot cible en fonction des mots de contexte dans une fenêtre spécifique. La couche d'entrée contient les mots de contexte et la couche de sortie contient le mot cible.

### 2. Skip Gram

Contrairement au modèle CBOW qui est formé pour prédire un mot cible en fonction des mots de contexte, le modèle Skip-gram est formé pour prédire les mots de contexte en fonction du mot cible. La couche d'entrée contient le mot cible et la couche de sortie contient les mots de contexte

## GloVe

GloVe est un algorithme d'apprentissage non supervisé pour générer des représentations vectorielles pour les mots. Il a été construit à partir de deux méthodes principales sont la factorisation matricielle globale et la fenêtre de contexte local.

## FastText

FastText est une autre méthode de plongement de mots qui est une extension du modèle word2vec. Au lieu d'apprendre directement des vecteurs de mots, fastText représente chaque mot comme un n-gramme de caractères. Par exemple, prenez le mot « artificiel » avec  $n=3$ , la représentation fastText de ce mot est  $\{ar, art, rtí, tí, íf, íc, íc, íc, íc, íc\}$ .

FastText permet de prendre en compte la structure interne des mots. De plus fastText fonctionne bien avec les mots rares. Donc, même si un mot n'a pas été vu pendant l'entraînement, il peut être décomposé en n-grammes pour obtenir son vecteur de plongement. Word2vec et GloVe ne parviennent pas à fournir des représentations vectorielles pour les mots qui ne figurent pas dans le dictionnaire du modèle. C'est le grand avantage que possède cette technique.

# Chapitre III

## Réalisation par PYTHON

### 1. Langage de programmation



Python est le langage de programmation open source le plus employé par les informaticiens, à cause de sa simplicité et sa facilité d'utilisation, il est simple à coder, vous n'êtes pas obligé d'ajouter des points-virgules ou des accolades n'importe où. En python, vous pouvez écrire de petits codes pour effectuer de grandes tâches. Par conséquent, vous gagnez du temps même lors de l'écriture du code

### 2. Environnement



Spyder 3 (nommé Pydee dans ses premières versions) est un environnement de développement pour Python. Libre (Licence MIT) et multiplateforme (Windows, Mac OS, GNU/Linux), il intègre de nombreuses bibliothèques d'usage scientifique : Matplotlib, NumPy, SciPy et IPython.

### 3. Nettoyage de donnée

Le nettoyage des données est un processus qui vise à identifier et corriger les données altérées, inexacts ou non pertinentes. Cette étape fondamentale du traitement des données améliore la cohérence, fiabilité et valeur des données.

C'est une étape essentielle avant que les données soient prêtes à être analysées. Les données textuelles qu'ils ont déjà grattées sont hautement non structurées et bruyantes par nature, les fautes de frappe, caractères HTML, la mauvaise grammaire, l'utilisation d'argot, la présence de contenu indésirable comme les URL, les mots vides, les emojis, etc. Python propose plusieurs bibliothèques qui aide à nettoyer le maximum les données ;

Alors il existe plusieurs méthodes du nettoyage des données :

1. La Bibliothèque Beautiful Soup nous donne la possibilité de supprimer les caractères HTML.
2. La bibliothèque redonne la possibilité de remplacer les caractères de votre choix avec le vide en utilisant la fonction **re.sub()**.

```
def clean_text(text_to_clean):
    res = text_to_clean.lower()
    res = re.sub(r"i'm", "i am", res)
    res = re.sub(r"he's", "he is", res)
    res = re.sub(r"she's", "she is", res)
    res = re.sub(r"it's", "it is", res)
    res = re.sub(r"that's", "that is", res)
    res = re.sub(r"what's", "what is", res)
    res = re.sub(r"where's", "where is", res)
    res = re.sub(r"how's", "how is", res)
    res = re.sub(r"\\ll", " will", res)
    res = re.sub(r"\\ve", " have", res)
    res = re.sub(r"\\re", " are", res)
    res = re.sub(r"\\d", " would", res)
    res = re.sub(r"\\re", " are", res)
    res = re.sub(r"won't", "will not", res)
    res = re.sub(r"can't", "cannot", res)
    res = re.sub(r"n't", " not", res)
    res = re.sub(r"n'", "ng", res)
    res = re.sub(r"bout", "about", res)
    res = re.sub(r"'til", "until", res)
    res = re.sub(r"[-()\\\"#/@;:<>{}~+=~|.!?,]", "", res)
    return res
```

Figure 15.1- nettoyage de données

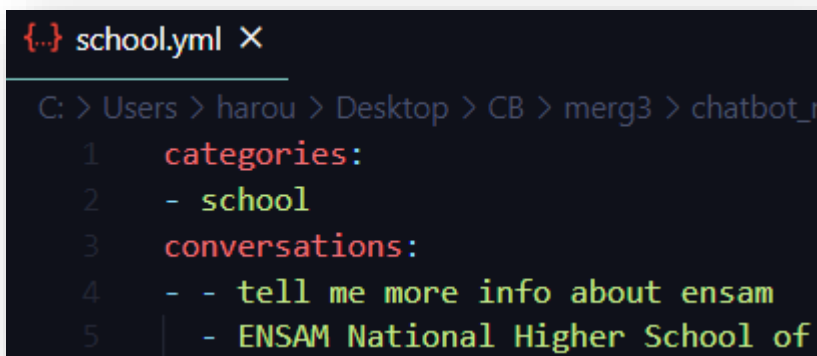
Changement de décodage des données

- Suppression des ponctuations
- Suppression des liens
- Suppression des hashtags
- Suppression des mentions

La dernière étape est la suppression des emojis, et pour cela on doit créer une fonction `remove_emojis` qui englobe plusieurs représentations des fameux emojis et on l'applique sur nos données. Les résultats finals sont des données bien structurés, propres prêts à être traitées par notre modèle neuronale.

### 4. Corpus et base de données

Concernant notre thème, on a créé un corpus sous fichier **YML** qui contient 30 (Questions / Réponses)

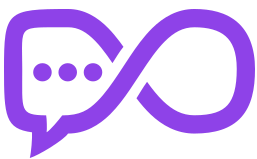


```
{...} school.yml X
C: > Users > harou > Desktop > CB > merg3 > chatbot_r
1 categories:
2 - school
3 conversations:
4 - - tell me more info about ensam
5 - - ENSAM National Higher School of
```

Figure 15.2- fichier YML contient l'ensemble des Q/A

- ⇒ La taille de notre corpus est très petite, donc on a besoin d'un moyen qui va nous aider à enrichir notre base de données, la solution est d'utiliser **chatette** ou créer un programme similaire à lui.

#### ➤ Chatette



Chatette est un programme qui permet d'enrichir les bases de données, il utilise les synonymes pour générer plusieurs exemples d'une question donnée. Chatito ou Chatette prend en entrée les phrases qu'on souhaite les traiter et les liste des synonymes, et il génère automatiquement un fichier JSON, qui contient les nouvelles phrases avec tous les synonymes.

### 5. Word Embeddings :

Bibliothèque Gensim Python

Gensim est une bibliothèque Python pour la modélisation de sujets, l'indexation de documents et la recherche de similitudes en utilisant de grands corpus. Le public cible est la communauté qui s'intéresse au traitement du langage naturel (TLN) et de la recherche d'informations. Gensim prend en charge une implémentation de l'incorporation de mots pour l'apprentissage de nouveaux vecteurs de mots à partir de texte.

### 6. Word2Vec

La première méthode de plongements de mots, utilisée dans ce projet, est Word2Vec, C'est un algorithme d'apprentissage d'un mot incorporé à partir d'un corpus de texte. Gensim fournit la classe Word2Vec pour travailler avec un modèle Word2Vec. Apprendre un mot incorporé à partir du texte implique de charger et d'organiser le texte en phrases et de les fournir au constructeur d'une nouvelle instance de Word2Vec (). Plus précisément, chaque phrase doit être symbolisée, préparée et par la suite divisée en mots.

Word2Vec possède différents paramètres, dont les plus importants sont :

- Taille (Size) : (par défaut 100) Dimensionnalité des vecteurs de mots ;
- Fenêtre (Window) : (par défaut 5) La distance maximale entre un mot cible et les mots autour du mot cible ;
- Min count : (par défaut 5) le nombre de mots entourant le mot cible : les mots dont l'occurrence est inférieure à ce nombre seront ignorés.
- Workers : (par défaut 3) Le nombre de threads à utiliser pendant la formation.
- SG : (par défaut 0 ou CBOW) L'algorithme d'apprentissage, soit 0 CBOW ou 1 SG.

Une fois le modèle formé, il est accessible via l'attribut « wv ». Il s'agit du véritable modèle de vecteur de mots dans lequel les requêtes peuvent être effectuées. Enfin, un modèle entraîné peut ensuite être enregistré dans un fichier en appelant la fonction save-word2vec-format (). Le modèle est enregistré soit sous format binaire ou texte. Le modèle enregistré peut ensuite être à nouveau chargé en appelant la fonction Word2Vec.load ().

Avantages

- Faciles à entraîner.
- Rapide / efficace.

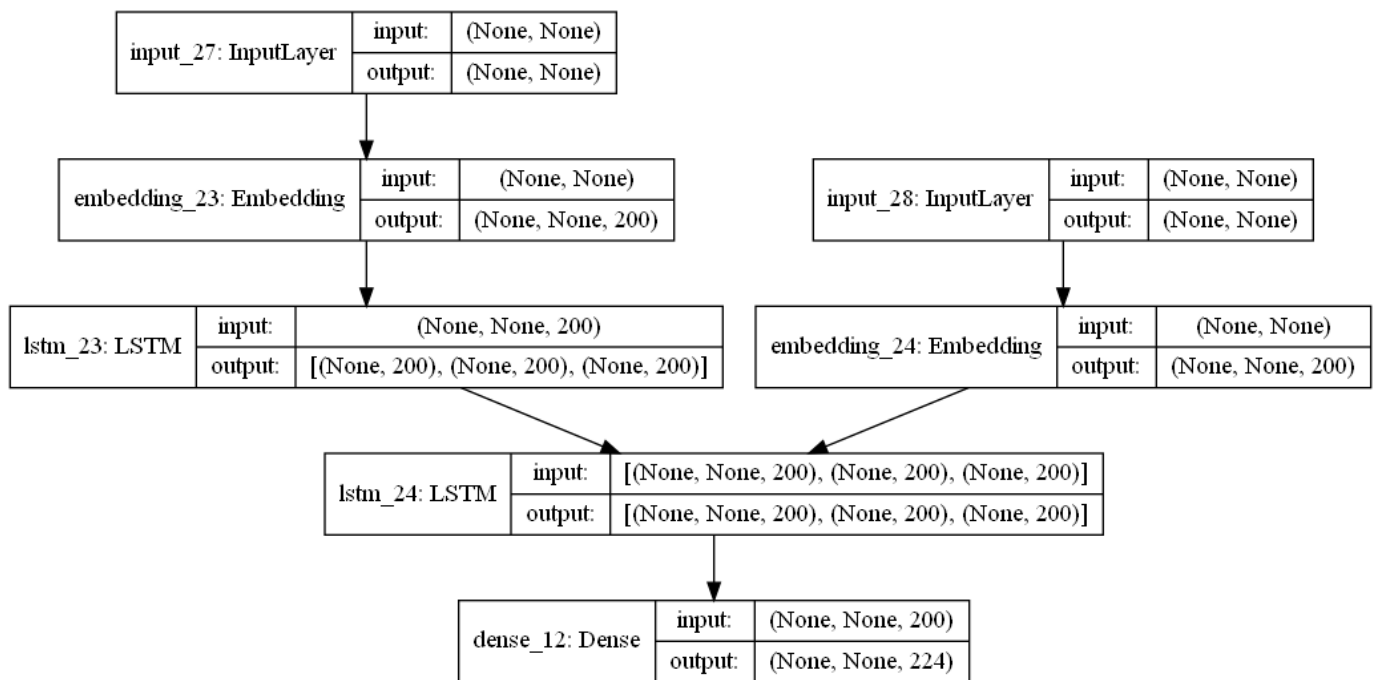
Inconvénients

- Modèles opaques : les dimensions sont difficilement interprétables
- Plus difficile de contrôler les paramètres utilisés pour entraîner les modèles.

### 7. Model

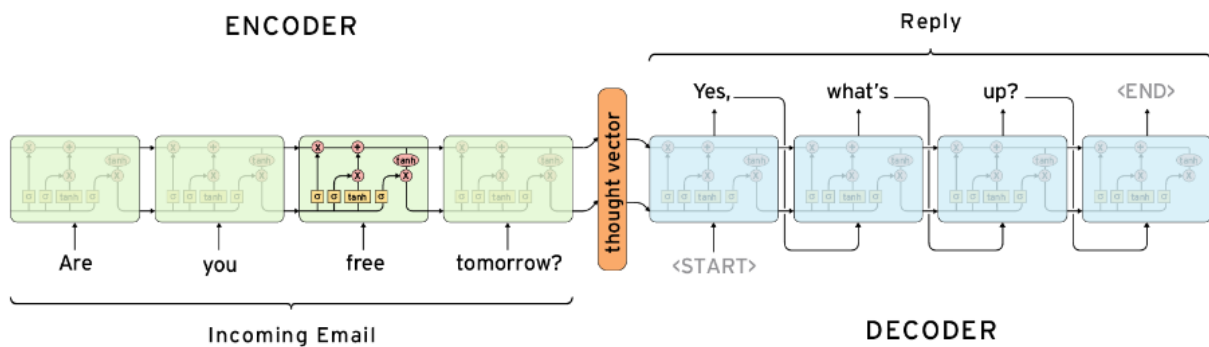
La base de données de school est trop petite avec un total de **X** Q/A mais il a avantage. Dans la base de données school plusieurs questions ont la même réponse. Les réponses uniques sont égales à **Y**. Pour résoudre le problème de la base de données qui est trop petite, on a décidé de travailler avec un modèle de classification basée sur Long short Term Memory Term (**LSTM**). Donc on est obligé de garder la liste des questions, et mettre les réponses sous la forme des classes numéroté de 1 à **Y**, car le corpus contient juste **Y** réponses unique sur les **X** questions. Par la suite on représente la classe de chaque réponse par un vecteur One Hot de taille **Y**. L'architecture utilisée prend le vecteur de la question en entrée, et il le transfère à une matrice de 2 dimensions dans la couche embeddings, en utilisant les représentations des mots déjà traités.





**Figure 11-** Architecture du Model school

L'idée principale derrière ce modèle est qu'il contient un encodeur (LSTM) et un décodeur (LSTM). L'un pour « comprendre » la séquence d'entrée et l'autre pour « décoder » le « vecteur d'états » et construire une séquence de sortie. Donc, pour cette approche, nous avons ajouté les deux jetons *-start-* et *-end-* à chaque paire du Corpus.



**Figure 16.1-** Modèle Encodeur Décodeur

#### Modèle Encodeur Décodeur implémenté

Donc, la prédiction est faite par les étapes suivantes

- (1) Encoder la séquence d'entrée en vecteurs d'état.
- (2) utiliser ces vecteurs comme états initiales du décodeur.
- (3) prédire le premier mot après le jeton de début de la réponse *-start-*.
- (4) Répétez jusqu'à ce que nous générions le caractère de fin de séquence ou que nous atteignons la limite de caractères.

### 8. Entraînement et Test

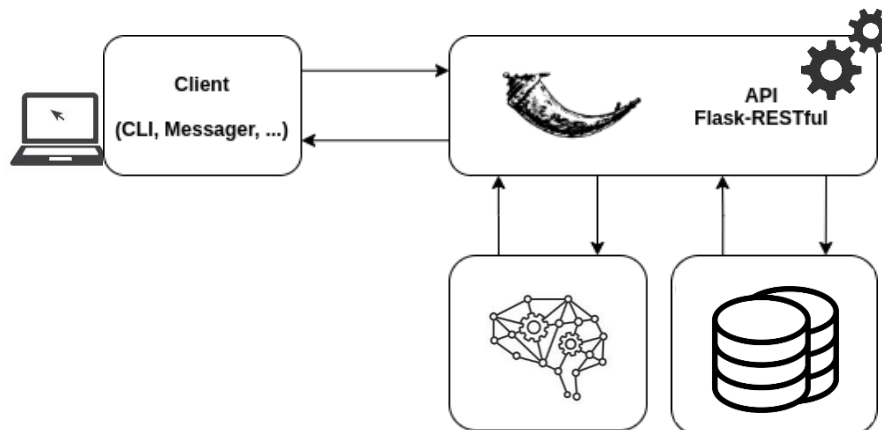
Pour compiler le modèle on a besoin de deux paramètres essentiels qui sont l'**optimizer**<sup>1</sup> et **loss**<sup>2</sup>, on a choisi pour le premier **RMSPROP** et le deuxième **sparse-categorical-crossentropy**. La dernière étape est d'indiquer le nombre des **époches**<sup>3</sup> et le **batch-size**<sup>4</sup>.

### 9. Architecture



Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires.

L'architecture adoptée pour ce projet est la tendance, qui sépare ce que les utilisateurs voient et comment elle est présentée et chargée, le Frontend, du traitement des données et de la gestion des demandes, le Backend. La figure 17 présente les deux composants dont nous parlons :

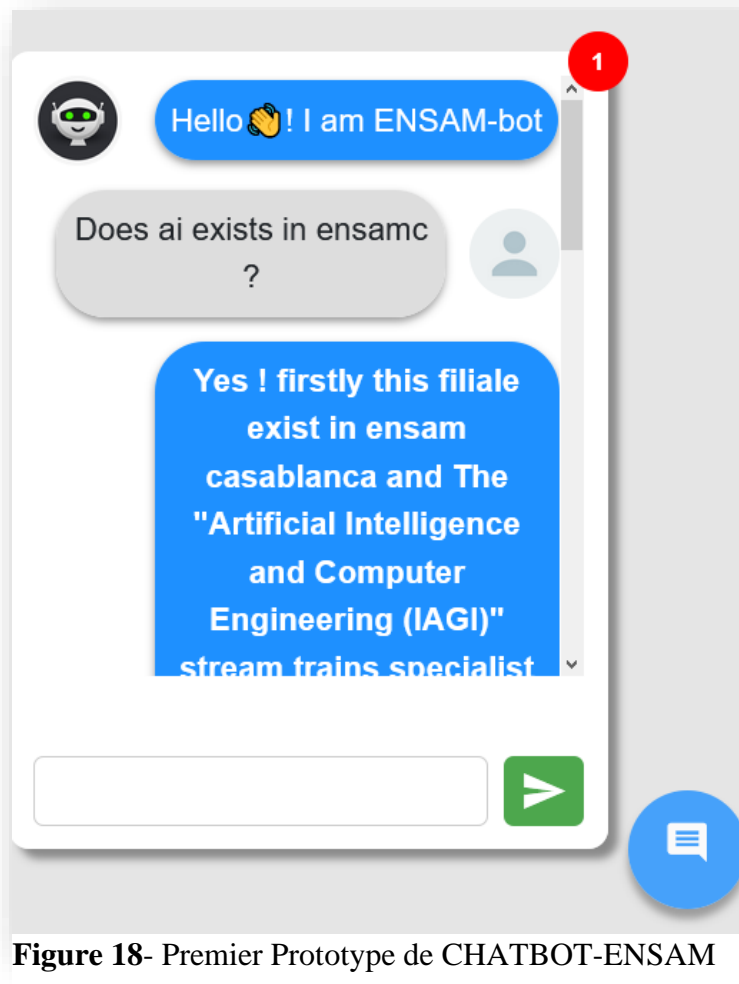


**Figure 17-** Architecture de CHATBOT-ENSAM

- 
1. Les optimiseurs sont des algorithmes ou des méthodes utilisés pour mettre à jour les poids du réseau neuronal.
  2. L'erreur que nous essayons de minimiser lors de l'apprentissage du modèle. Plus la perte est faible, plus nos prévisions sont proches des vraies étiquettes.
  3. Signifie combien de fois le modèle traite les données.
  4. Mis à jour plusieurs fois au cours d'une même époque.

## 10. Prototype et Résultat

Concernant l'interface de notre application, nous avons essayé de rendre l'interface utilisateur le plus simple possible afin de permettre aux utilisateurs de comprendre plus rapidement comment elle fonctionne. La figure 18 a présente la bulle de la messagerie codée par HTML-CSS-JS.



# Chapitre IV

## Réalisation par LARVEL

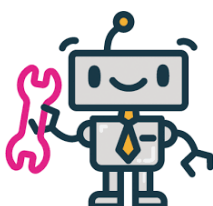
### 1. Langage et environnement



**PHP** "Hypertext Preprocessor", plus connu sous son sigle PHP (acronyme récursif), créé en 1994 par « Rasmus Lerdorf » est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande.



**Composer** est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin. Le développement a débuté en avril 2011 et a donné lieu à une première version sortie le 1<sup>er</sup> mars 2012. Développé au début par Nils Adermann et Jordi Boggiano. Le logiciel Composer est à l'initiative d'un portage en PHP du logiciel Libzypp satsolver d'Open Suse. Le logiciel Composer est fortement inspiré du logiciel npm pour Node.js et de bundler pour Ruby.



**BotMan** est une bibliothèque PHP indépendante du Framework conçue pour simplifier la tâche de développement de bots innovants pour plusieurs plates-formes de messagerie, notamment Slack, Telegram, Microsoft Bot Framework, Nexmo, HipChat, Facebook Messenger, WeChat et bien d'autres.



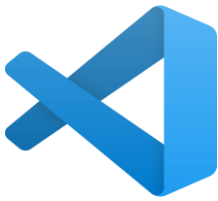
**Dialogflow** est une plate-forme de compréhension du langage naturel qui facilite la conception et l'intégration d'une interface utilisateur conversationnelle dans votre application mobile, application Web, appareil, robot, système de réponse vocale interactif, etc. En utilisant Dialogflow, vous pouvez proposer aux utilisateurs de nouvelles façons intéressantes d'interagir avec votre produit.

Dialogflow peut analyser plusieurs types d'entrées de vos clients, y compris des entrées de texte ou audio (comme à partir d'un téléphone ou d'un enregistrement vocal). Il peut également répondre à vos clients de plusieurs manières, soit par texte, soit par discours synthétique



**Laravel** est un framework d'application Web avec une syntaxe expressive et élégante. Un framework Web fournit une structure et un point de départ pour créer votre application, vous permettant de vous concentrer sur la création de quelque chose d'extraordinaire pendant que nous transpirons les détails.

Laravel s'efforce de fournir une expérience de développeur incroyable, tout en fournissant des fonctionnalités puissantes telles que l'injection de dépendances approfondie, une couche d'abstraction de base de données expressive, des files d'attente et des tâches planifiées, des tests unitaires et d'intégration



**Visual Studio Code** est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

## 2. Création de notre agent sur Dialogflow : [23]

Après avoir créé un compte sur Dialogflow, accédant à la liste des agents puis on clique sur créer un nouvel agent :

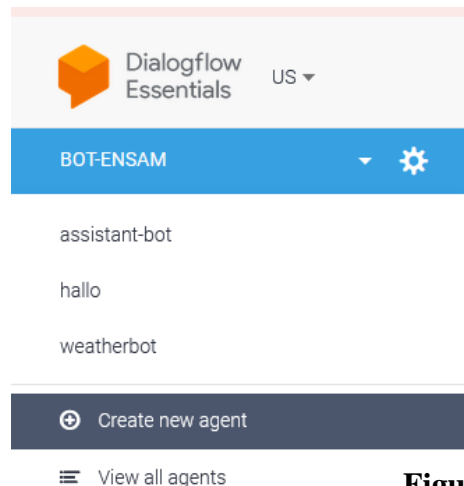


Figure 19.1- dialogflow panel

On doit donner le nom de notre agent et spécifier time zone pour la synchronisation des messages.

CREATE

DEFAULT LANGUAGE ⓘ

English – en

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT0:00) Africa/Casablanca

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

Create a new Google project

Enables Cloud functions, Actions on Google and permissions management.

Figure 19.2- Création nouveau agent

Etape suivante consiste à créer des entités ;

Les entités sont les mots clés qui aident l'agent à comprendre les phrases.

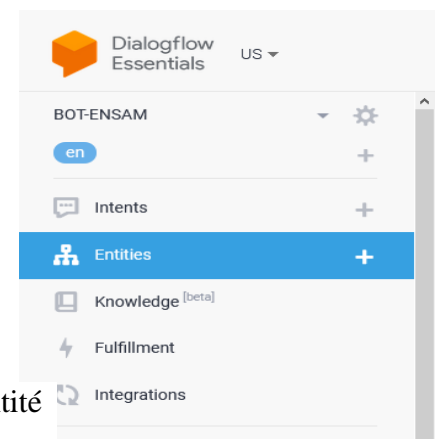


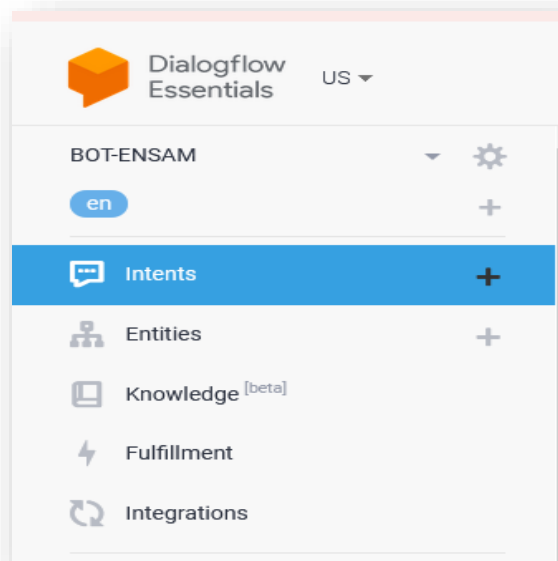
Figure 19.3- Ajouter une entité

On donne le nom de l'entité puis la valeur de référence et ses synonymes.

The screenshot shows the 'Create synonyms' interface in Dialogflow. At the top, the entity name 'ensamc\_ai' is displayed next to a 'SAVE' button. Below this, there are four checkboxes: 'Define synonyms' (checked), 'Regexp entity', 'Allow automated expansion', and 'Fuzzy matching'. A text input field contains the word 'ai', and a list of synonyms is shown: 'ai, artificial intelligence, genie informatique, intelligence artificiel et genie informatique, computer science, ai&gi, ai-gi, artificial intelligence and computer science, computer engineering, aigi, iagi, giai'.

**Figure 19.4-** Créer des synonymes

Pour utiliser les entités on doit créer les intentions pour notre agent



**Figure 19.5-** Ajouter les intentions

On donne le nom de l'intention et le contexte ;

The screenshot shows the 'Add intent' interface in Dialogflow. At the top, the entity name 'ensamc\_ai' is displayed next to a 'SAVE' button. Below this, there is a section for 'Contexts'. It includes a text input field for 'Add input context' and a row of context tags: '5 ensamc\_ai', '5 ai', and 'Add output context'.

**Figure 19.6-** le nom de l'intention

Puis les phrases de l’entrainement, les paramètres d’action et les repenses.

Training phrases ?

” Add user expression

” more infos

” tell me about aigi

” artificial intelligence exists in ensam casablanca

” ai

” does it exist ai in ensam

” what is Ai&gi

” what is ai ?

” does it exist ai in ensam casablanca

” tell me about ai filiale

Comme nous montre la figure, l’agent sait directement les mots clés

Figure 19.7- les phrases de l’entrainement

Ce paramètre de référence on va l’utiliser dans notre code backend pour bien gérer la conversation.

Action and parameters

ask\_ensamai

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
<input type="checkbox"/>	ensamc_ai	@ensamc_ai	Sensamc_ai	<input type="checkbox"/>
<input type="checkbox"/>	ensamc	@ensamc	Sensamc	<input type="checkbox"/>
<input type="checkbox"/>	infos	@infos	Sinfos	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

Figure 19.8- les actions et les paramètres

L'agent affichera la réponse selon l'entité détectée dans la conversation ;

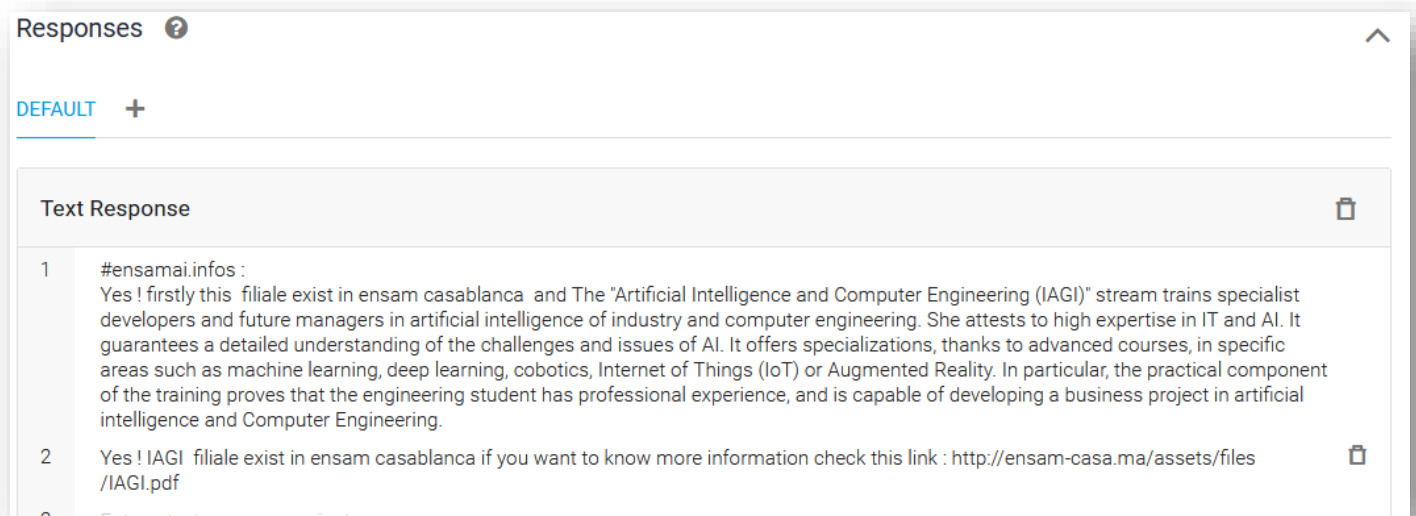


Figure 19.9- les réponses

### 3. Générer le JWT

Dans les paramètres de notre agent on clique sur l'ID de Projet :

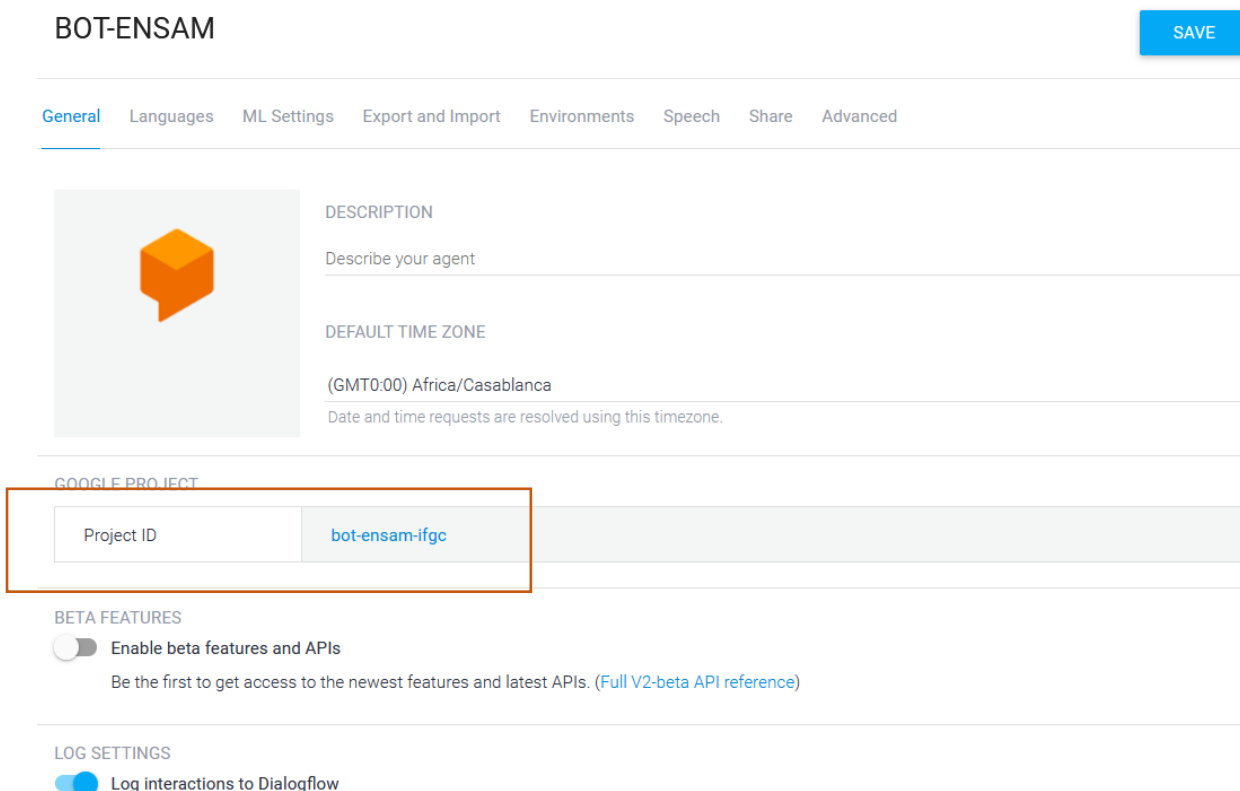
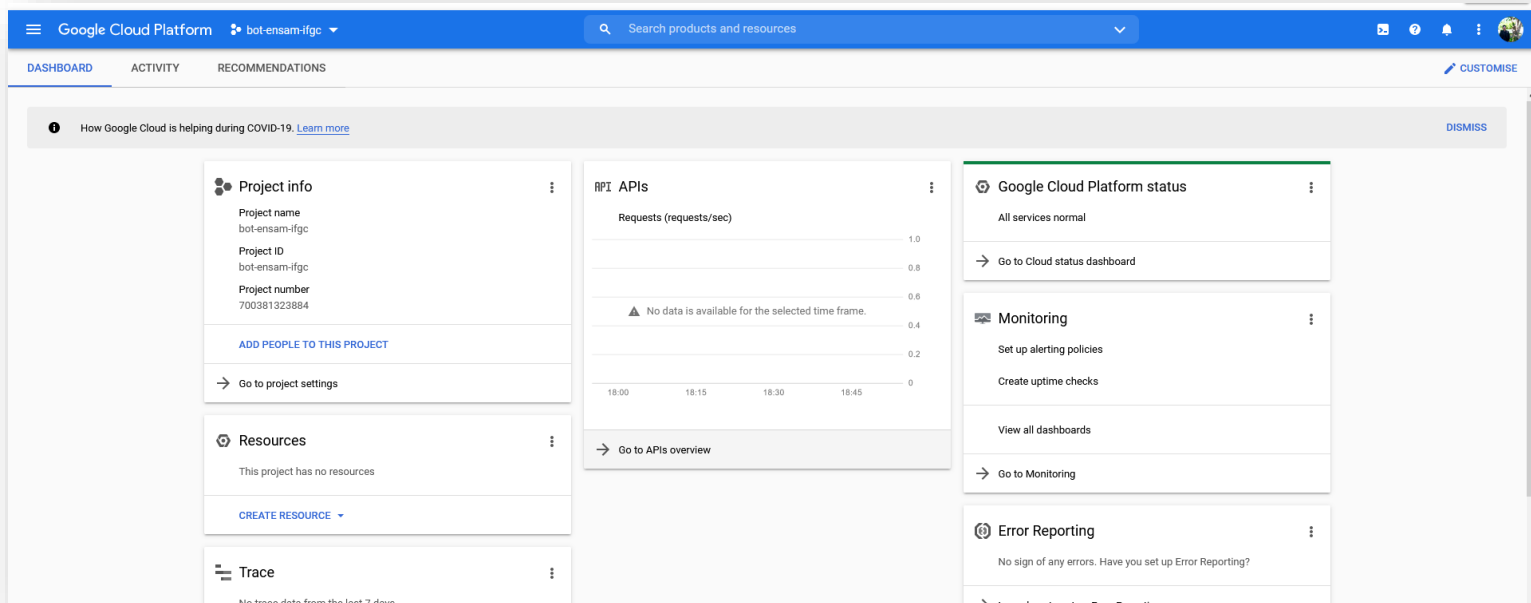


Figure 20- ID de Projet

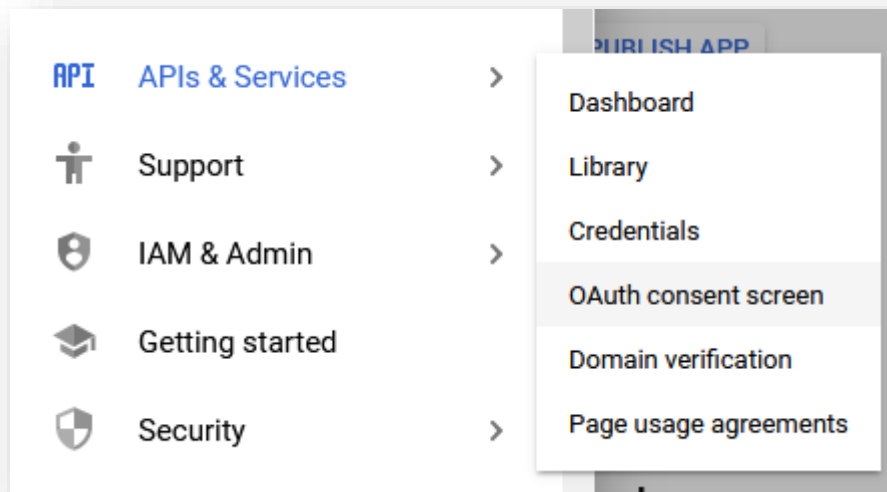


Le lien nous dirige vers Google Cloud Platform,



**Figure 21-** Google Cloud Platform

On va créer « *OAuth consent screen* » ;

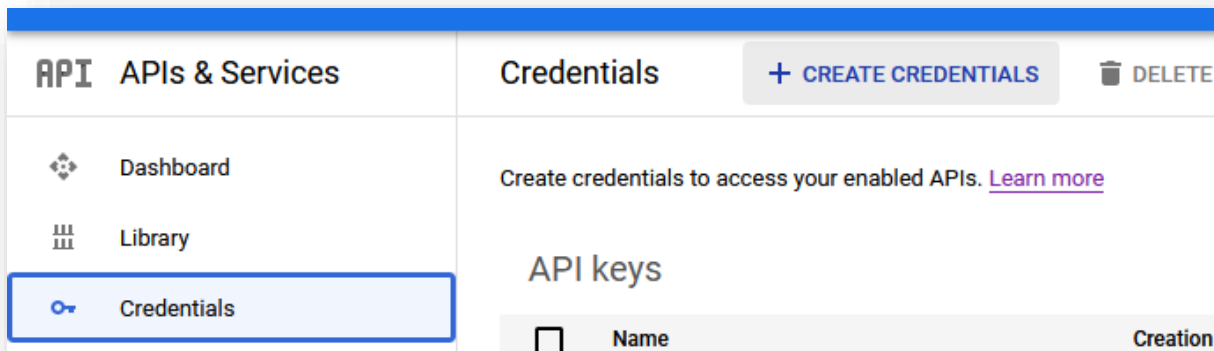


**Figure 21.1-** OAuth 2.0 paramétrage

Pour assurer la sécurité de l'application on donne le nom et le mail du développeur de l'application et définir le scope d'API

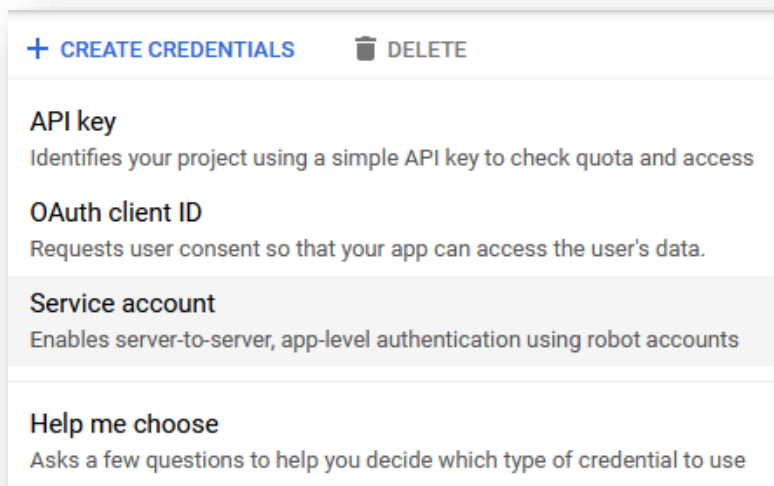
Etape suivante et de créer la certification et accréditation ;

Dans le volé crédenciales :



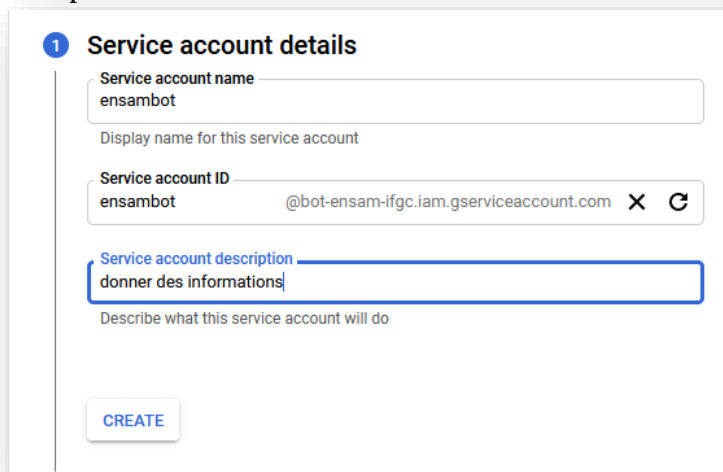
**Figure 21.2-** Création des crédenciales

On va donner les services pour chaque compte de développeurs d'API :



**Figure 21.3-** les choix possible pour configurer OAuth

Donner le nom et ID du compte de service :



**Figure 21.4-** le nom du compte de service

Accorder l'accès au projet pour ce compte de service ;

The screenshot shows the 'Grant this service account access to the project (optional)' step in the Google Cloud IAM console. It lists two roles: 'Dialogflow API Admin' and 'Dialogflow API Client'. Each role has a description of its permissions and a link to 'Add condition'. A '+ ADD ANOTHER ROLE' button is at the bottom. A 'CONTINUE' button is at the bottom right.

**Service account details**

**2 Grant this service account access to the project (optional)**

Grant this service account access to bot-ensam-ifgc so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role	Condition
Dialogflow API Admin Can query for intent; read & write session properties; read & write agent properties.	<a href="#">Add condition</a>
Dialogflow API Client Can call all methods on sessions and conversations resources as well as their descendants.	<a href="#">Add condition</a>

[+ ADD ANOTHER ROLE](#)

[CONTINUE](#)

**3 Grant users access to this service account (optional)**

Donner comme rôle :

- Dialogflow API Admin
- Dialogflow API Client

**Figure 21.5-** les détails sur les méthodes durant la session

Accorder aux utilisateurs l'accès à ce compte de service ;

The screenshot shows the 'Grant users access to this service account (optional)' step. It has two sections: 'Service account users role' with one user 'haroun.ezzahraoui.8@gmail.com' and 'Service account admins role' with four users: 'lifimostafa@gmail.com', 'haroun.ezzahraoui.8@gmail.com', 'amine.gatou98@gmail.com', and 'hamza10bayoud@gmail.com'. There are 'DONE' and 'CANCEL' buttons at the bottom.

**3 Grant users access to this service account (optional)**

Grant access to users or groups that need to perform actions as this service account. [Learn more](#)

**Service account users role**  
haroun.ezzahraoui.8@gmail.com

Grant users the permissions to deploy jobs and VMs with this service account

**Service account admins role**  
lifimostafa@gmail.com haroun.ezzahraoui.8@gmail.com amine.gatou98@gmail.com hamza10bayoud@gmail.com

Grant users the permission to administer this service account

[DONE](#) [CANCEL](#)

Accorder le service aux comptes des coéquipière dans le projet

**Figure 21.6-** Donnée l'accès au utilisateurs et développeurs

Par la suite on clique sur API Key pour le générer ;

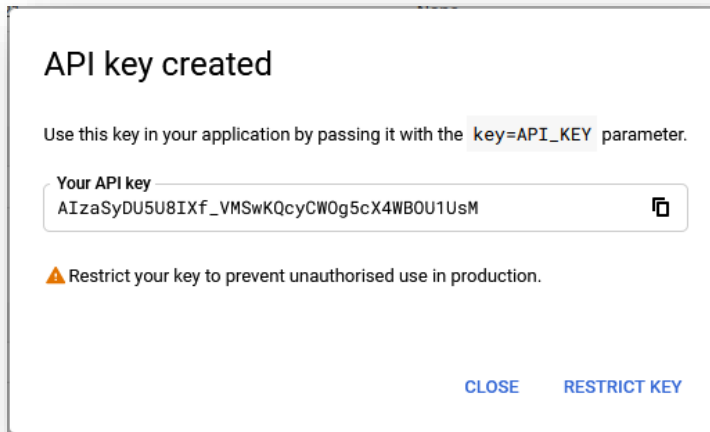


Figure 21.7- Génération de API KEY

Maintenant pour télécharger le fichier JSON on accède au compte de service

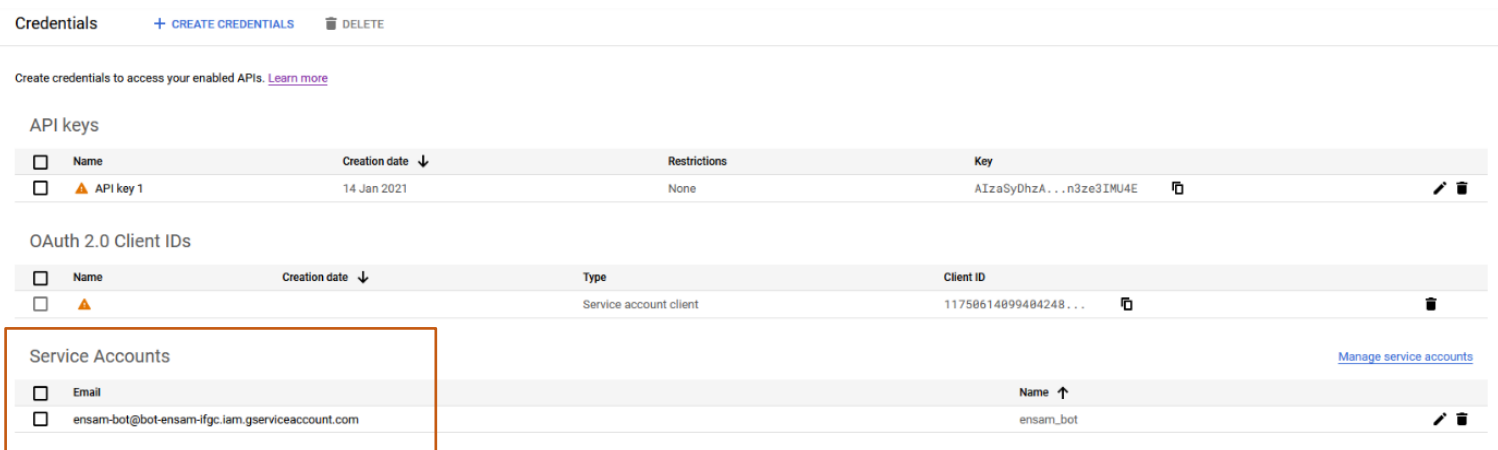


Figure 21.8- OAuth Crédenciales

Puis « KEYS » et ajouter une clé :

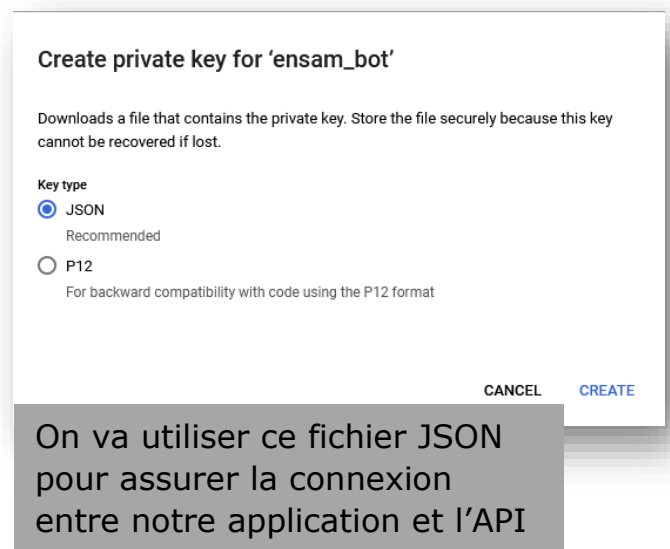
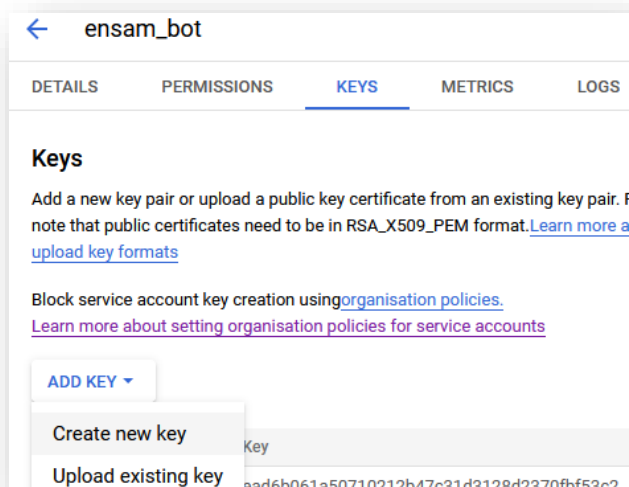
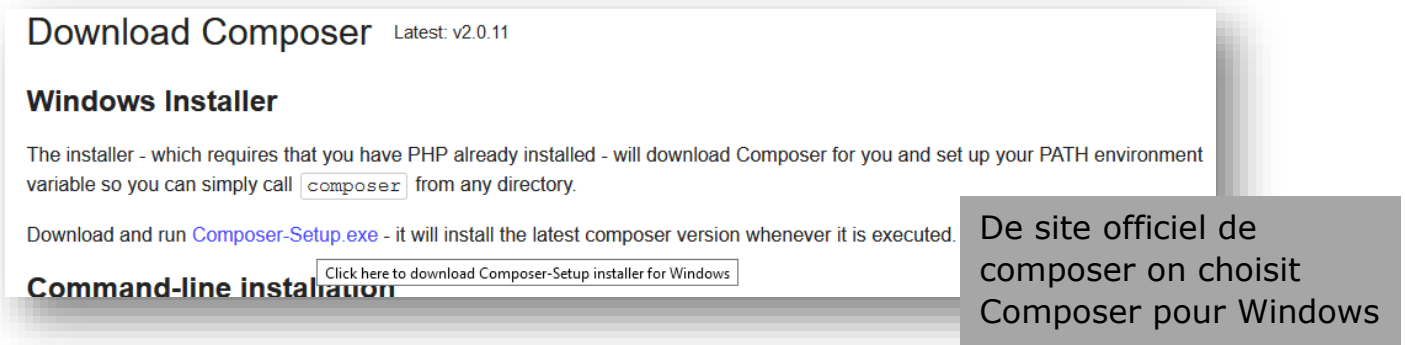


Figure 21.9- Générer nouvelle key

### 4. Installation de Laravel :

Premièrement on installe Composer :



**Figure 22-** Site Officielle de Composer

On installe LARAVEL via Composer sur l'invite de commande ;

```
D:\testlara
λ composer global require laravel/installer
Changed current directory to C:/Users/harou/AppData/Roaming/Composer
Using version ^4.1 for laravel/installer
./composer.json has been updated
```

**Figure 22.1-** Installation de LARAVEL

### 5. Installation de Botman : [22]

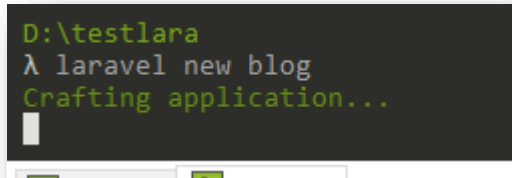
L'installation de Botman se fait par deux méthodes on installe directement BotmanStudio ou via Composer ;

```
D:\testlara
λ composer global require "botman/installer"
Changed current directory to C:/Users/harou/AppData/Roaming/Composer
Using version ^1.0 for botman/installer
./composer.json has been updated
Running composer update botman/installer
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating autoload files
14 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
```

**Figure 23 -** Installation de BOTMAN

## 6. Création d'Application Laravel :

Créant notre application laravel nommé blog

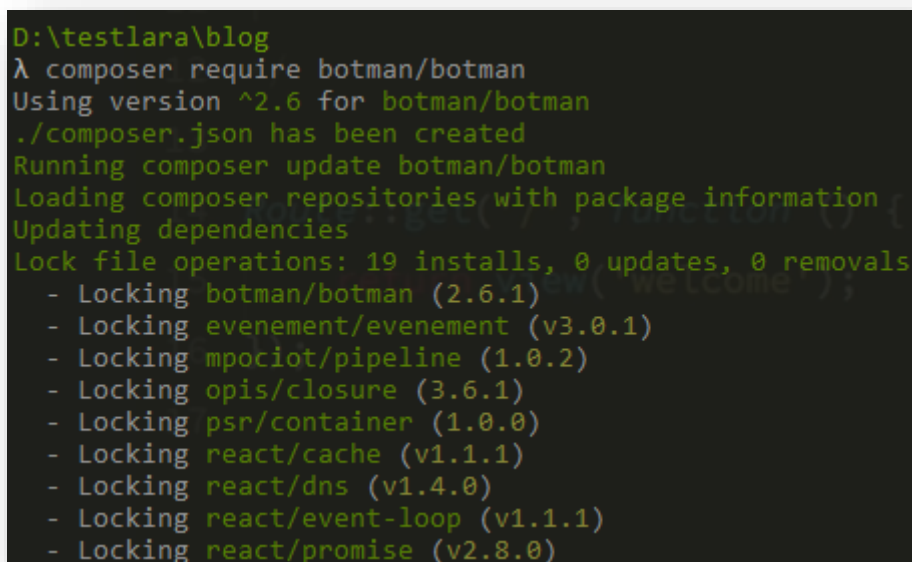


```
D:\testlara
λ laravel new blog
Crafting application...
```

**Figure 24** – Commende de création de projet laravel

## 7. Ajouter Botman à l'Application Laravel :

Pour ajouter Botman à l'application laravel on va utiliser Composer pour installer Botman exigences ;

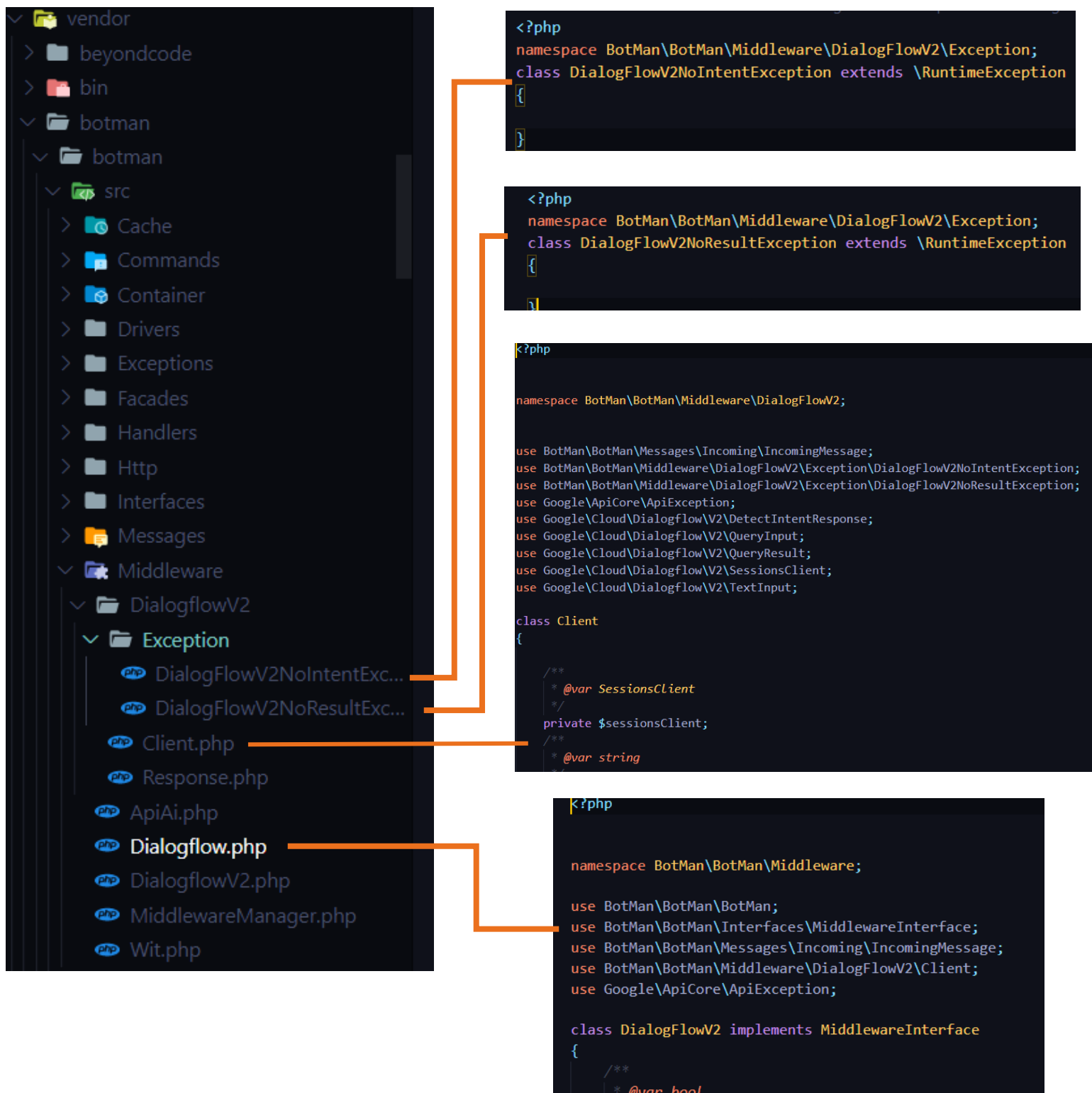


```
D:\testlara\blog
λ composer require botman/botman
Using version ^2.6 for botman/botman
./composer.json has been created
Running composer update botman/botman
Loading composer repositories with package information
Updating dependencies
Lock file operations: 19 installs, 0 updates, 0 removals
- Locking botman/botman (2.6.1)
- Locking evenement/evenement (v3.0.1)
- Locking mpociot/pipeline (1.0.2)
- Locking opis/closure (3.6.1)
- Locking psr/container (1.0.0)
- Locking react/cache (v1.1.1)
- Locking react/dns (v1.4.0)
- Locking react/event-loop (v1.1.1)
- Locking react/promise (v2.8.0)
```

**Figure 25** – Commende d'intégration de BOTMAN dans le projet LARAVEL

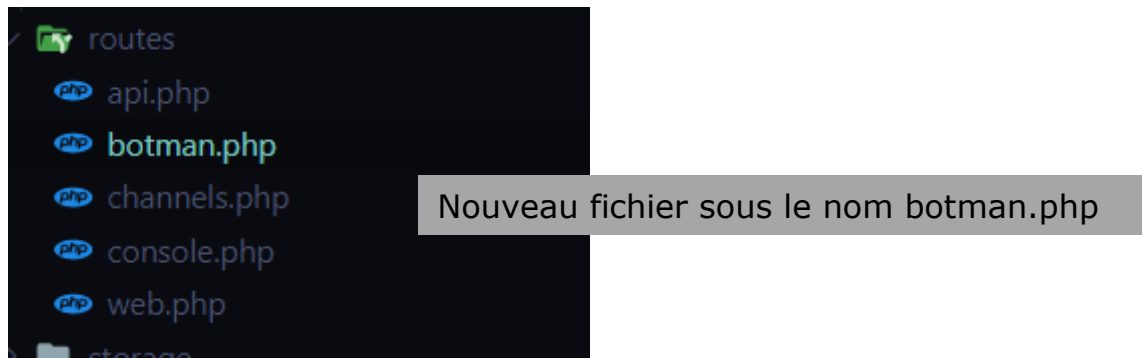
### ➤ Création des Middlewares personnalisés

Les services de google ne sont plus accessible via AccessToken, c'est pour cette raison on a créé JWT qui fait l'appel à la connexion via l'OAuth 2.0, dans ce cadre on est obligé de de créer nos middlewares personnalisés ;



**Figure 26** – Création de nouveau Middleware pour assurer la connexion avec Dialogflow V2

Après l'installation des exigences on va ajouter un fichier manuellement,  
Dans **routes** :



**Figure 26.1** – Le dossier routes

Dans ce fichier on va donner les fonctions d'écoutes « **hears** » pour récupérer les paramètres d'action spécifiés dans Dialogflow et on utilise un middleware « **received** » pour récupérer les messages et faire l'appel à la fonction « **reply** » pour envoyer la réponse au client ;



**Figure 26.2** – Contenu de fichier Botman.php

Créant un fichier de configuration pour le pilote et le cache,  
dans le dossier « **config** » ;

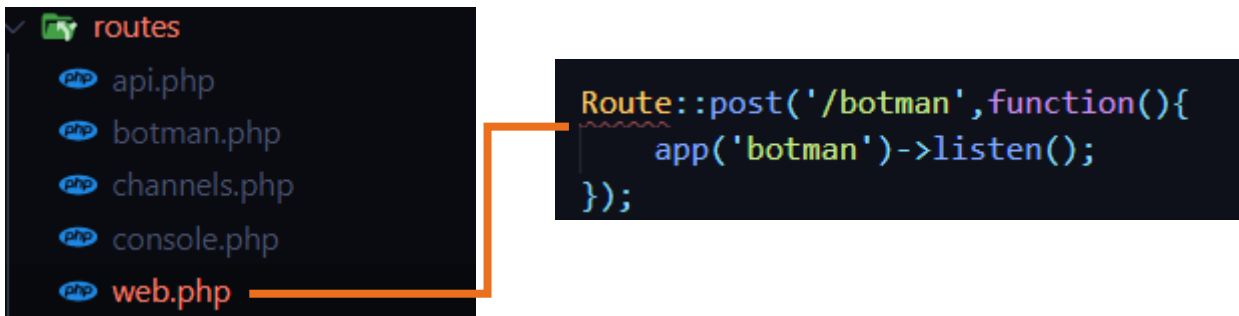


**Figure 26.3** –Le dossier Config



Ajoutant des routes pour les requêtes de Botman ;

Dans le dossier « **routes** »



**Figure 26.4** – Modification de web.php

Ajoutant un widget de Botman dans le fichier welcome.blade.php ;

Dans le dossier « **resources** »

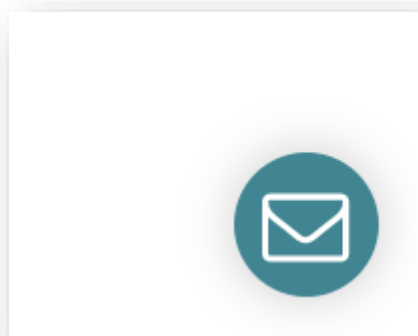


**Figure 26.5** – Modification dans Botman widget

## 8. Prototype et résultats :

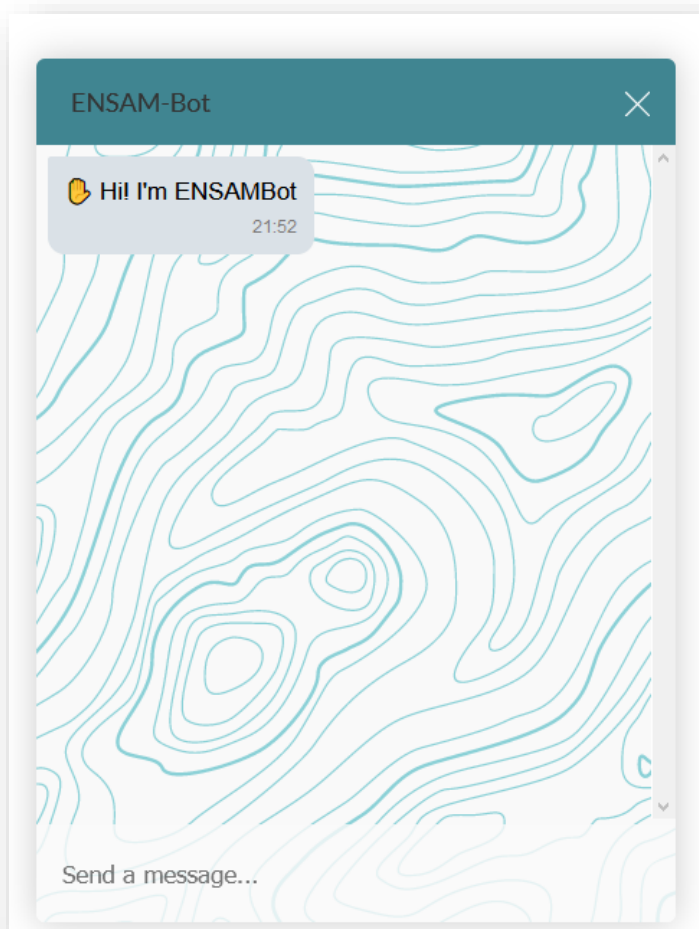
Concernant l'interface de notre application web, nous avons essayé de rendre l'interface utilisateur le plus simple possible, par l'utilisation de Botman Template afin de permettre aux utilisateurs de comprendre plus rapidement comment elle fonctionne.

Sur notre site le popup suivant apparaît :

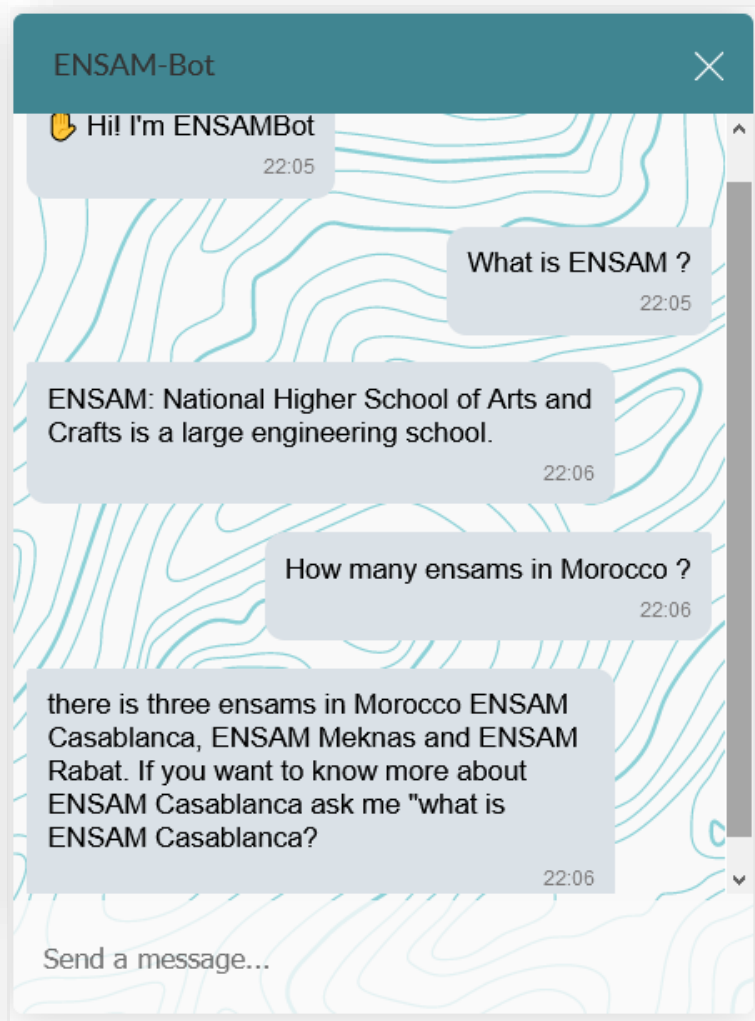


**Figure 27**– La bulle de messagerie

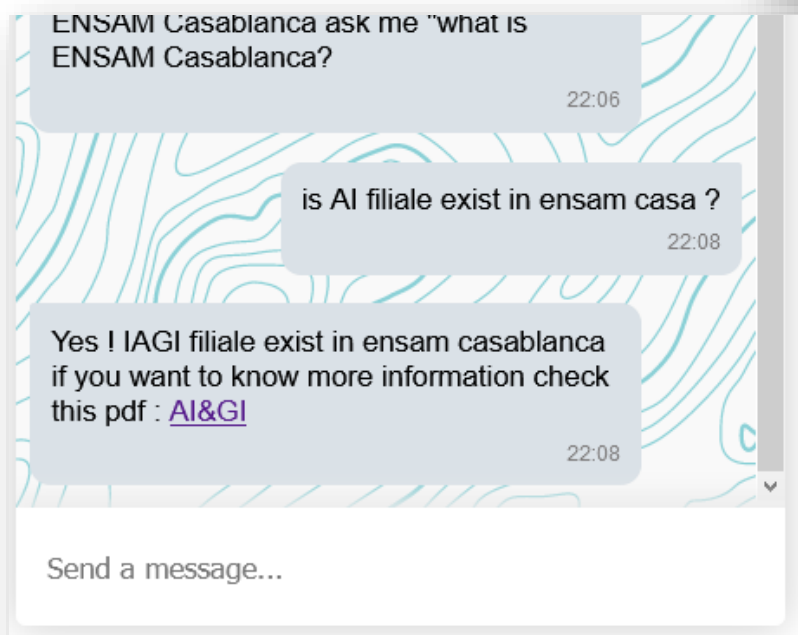
Si l'utilisateur clique sur le popup directement affiche la boîte de communication suivante :



**Figure 27.1**– Premier test de ENSAM-Bot



**Figure 27.2**– Test d’affichage des Message et repenses



**Figure 27.3**– Test d’envoi des liens par message

# Conclusion

Le domaine des agents conversationnels virtuels est appelé à prendre un rôle encore plus important en raison du développement des entreprises et leurs orientations vers la digitalisation. Ce projet nous a permis de découvrir ce monde assez riche et large, et construire une idée générale et une vision propre sur ce métier.

Durant la préparation de notre projet de stage, nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et les formations qu'on a suivies sur des plateformes éducatives surtout au niveau de la partie Deep Learning, dans le but de concevoir et réaliser un chatbot d'orientation avec la langue anglaise.

L'application des techniques d'apprentissage en profondeur, et une interface utilisateur conviviale, ont été les principaux défis que nous avons essayé de passer pour fournir les résultats convoités dans le dernier chapitre.

L'ensemble de notre travail va être exécutable sur le site web de l'école à l'adresse "<http://ensam-casa.ma/>", et le code est visionnable sur un dépôt GitHub créé à cet effet.

# Bibliographie / Webographie

- [1] A. M. TURING. ICOMPUTING MACHINERY AND INTELLIGENCE. Mind, LIX (236) :433–460, 10 1950.
- [2] Eleni Adamopoulou and Lefteris Moussiades. An overview of chatbot technology. In Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis, editors, Artificial Intelligence Applications and Innovations, pages 373–383, Cham, 2020. Springer International Publishing.
- [3] Bhagwat, Vyas Ajay, ” Deep Learning for Chatbots” (2018). Master’s Projects. 630. scholarworks. DOI: <https://doi.org/10.31979/etd.9hrt-u93z> .
- [4] M Sewak. Deep reinforcement learning: Frontiers of artificial intelligence.
- [5] Gérald PETITJEAN. INTRODUCTION AUX RESEAUX DE NEURONES.  
<http://master-ivi.univ-lille1.fr/fichiers/Cours/rdf-semaine-8-neurones.pdf>
- [6] Wikistat. Réseaux de neurones, 2018.
- [7] Florent SIMON. Deep Learning, les fonctions d’activation.
- [8] Racine Ly. Les fonctions d’activation.
- [9] <https://www.techopedia.com/definition/32902/deep-neural-network>
- [10] Denny Britz. 2015. <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>
- [11] <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>
- [12] <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.  
<http://www.deeplearningbook.org>.
- [14] Sumit Saha. A comprehensive guide to convolutional neural networks. 2018.  
<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [15] Margaret Rouse. Traitement du langage naturel (tln ou nlp).  
<https://www.lemagit.fr/definition/Traitement-du-langage-naturel-TLN>.
- [16] GeeksForGeeks. <https://www.lemagit.fr/definition/Traitement-du-langage-naturel-TLN>.
- [17] GeeksForGeeks. <https://www.geeksforgeeks.org/python-lemmatization-with-nltk/>
- [18] D. Rao and B. McMahan. Natural language processing with pytorch. 2019
- [19] G. Genthial R. Mundra F. Chaubard, M. Fang and R. Socher. Natural Language Processing with Deep Learning. 2017.
- [20] Faneva RAMIANDRISOA Michel RAJOELINA. Intégration des plongements de mots dans les méthodes, supervisées et non supervisées, d’extraction automatique de mots clés.  
<https://atlas.irit.fr/PIE/VSSST/Actes-VSSST2018-Toulouse/Ramiandrisoa.pdf>.
- [21] Greg Corrado Jeffrey Dean Tomas Mikolov, Kai Chen. Efficient estimation of word representations in vector space.  
<https://arxiv.org/pdf/1301.3781.pdf>.
- [22] <https://botman.io/2.0/welcome>
- [23] <https://cloud.google.com/dialogflow/>