
TABLES DES MATIÈRES

Liste des illustrations	vii
Liste des tableaux.....	x
Introduction.....	1
1. Généralités.....	2
1.1. Contexte	2
1.2. Chatbot : Définition	2
1.3. Bref historique	3
1.4. Application des chatbots.....	4
1.4.1. Le chatbot serviciel	4
1.4.2. Le chatbot expérientiel.....	4
1.4.3. Le chatbot commercial.....	5
1.4.4. Le chatbot entertainment	5
1.5. Avantages.....	6
1.6. Limites	6
2. Approche technique.....	7
2.1. Types de chatbot	7
2.1.1. Chatbot simple.....	7
2.1.2. Chatbot conversationnel	8
2.1.2.1. L'intelligence artificielle	8
2.1.2.2. Machine Learning	8
2.1.2.3. Deep Learning	11
2.1.2.4. Les réseaux de neurones artificiels	11
2.1.2.5. Réseau de neurones récurrents (RNR)	12
2.2. Fonctionnement du chatbot conversationnel.....	13
2.2.1. La compréhension du langage naturel.....	13
2.2.2. Traitement de la demande	14
2.2.3. Génération de réponse	14
2.2.4. Architecture	14
2.3. Les possibilités de création d'un chatbot.....	15
2.4. Graphisme et configurations	16
3. Problématique	16

4. Chatbot et éducation.....	17
4.1. Quelques cas d'application	17
4.2. Rôles	19
4.3. Fonctionnalités	19
4.4. Chatbot et Moodle	20
4.4.1. T-Bot.....	20
4.4.2. Q-Bot.....	22
4.4.3. Plugin chatbot.....	22
5. Définition de cas d'application.....	23
5.1. Descriptif.....	23
5.2. Analyse des besoins.....	23
5.2.1. Equipe de support.....	23
5.2.2. Professeur	25
5.2.3. Etudiant	25
5.3. Choix de cas.....	27
5.3.1. Indiquer quelle est la clé d'inscription au cours	27
5.3.2. Proposer un quiz pour l'étudiant	28
6. Choix du moteur chatbot.....	29
6.1. Critères de sélection	29
6.2. Analyse comparative	30
6.3. Dialogflow	31
6.3.1. Client SDK.....	32
6.3.2. Web demo integration.....	32
7. Design.....	34
7.1. Eléments communs aux deux cas d'applications	34
7.2. Cas d'utilisation 1 – Clé d'inscription.....	36
7.2.1. Diagramme d'activités	36
7.2.2. Création des intentions.....	37
7.2.3. Création d'entités.....	38
7.3. Cas d'utilisation 2 – Quiz pour étudiant.....	39
7.3.1. Diagramme d'activité	39
7.3.2. Création des intentions.....	40

7.3.3. Création d'entités.....	40
8. Développement	42
8.1. Architecture	42
8.2. Environnement de développement	43
8.2.1. Installation de Moodle.....	43
8.2.2. Architecture des dossiers.....	43
8.2.3. Git.....	44
8.2.4. PHPStorm	44
8.2.5. Ngrok.....	45
8.3. Développement de plugin Moodle.....	46
8.3.1. Bloc HTML.....	46
8.3.2. Service web	48
8.3.3. Configuration administrateur.....	51
8.4. <i>Business logic</i> du chatbot	52
9. Tests.....	53
9.1. Création de données Moodle	53
9.1.1. Utilisateurs.....	53
9.1.2. Cours.....	54
9.1.3. Quiz.....	54
9.1.4. Attribution des rôles et accès aux quiz.....	54
9.2. Simulations	55
10. Méthodologie de gestion de projet	58
10.1. Général / Product backlog.....	58
10.2. Rôles.....	58
10.3. Evénements	59
10.4. Sprints.....	59
10.4.1. Sprint 0	59
10.4.2. Sprint 1	60
10.4.3. Sprint 2	61
10.4.4. Sprint 3	62
10.5. Difficulté d'application / Remarques	63
11. Bilan.....	64

12. Conclusion.....	65
12.1. Résultat de la thèse.....	65
12.2. Recommandations	66
12.3. Perspectives d’avenir	66
13. Références.....	67
14. Annexes	73
Annexe I – Comparatif moteurs NLP	73
Annexe II – Faisabilité cas support.....	74
Annexe III – Utilité / Faisabilité cas étudiant	75
15. Déclaration de l’auteur.....	76

LISTE DES ILLUSTRATIONS

Figure 1 Un exemple de chatbot messenger	5
Figure 2 Flux conversationnel d'un chatbot simple	8
Figure 3 Fonction $f(x)=180x$ et marges d'erreurs	10
Figure 4 Réseau de neurones	12
Figure 5 Fonctionnement d'un modèle de neurones récurrents	12
Figure 6 Architecture d'un chatbot conversationnel	15
Figure 7 Personnalisation du T-Bot	21
Figure 8 Plateforme actuelle - Menus	27
Figure 9 Plateforme actuelle - Onglet utilisateurs	28
Figure 10 Plateforme actuelle - Méthode d'inscription	28
Figure 11 Logo officiel	30
Figure 12 Logo officiel	31
Figure 13 Logo officiel	31
Figure 14 Logo officiel	31
Figure 15 Web demo integration	33
Figure 16 Architecture avec client PHP	34
Figure 17 Diagramme de séquence du flux conversationnel	34
Figure 18 Moudy, son apparence	35
Figure 19 Méthode JQuery	35
Figure 20 Diagramme d'activité demande de clé d'inscription	36
Figure 21 Entrées pour l'intention demande_clef_d'inscription	37
Figure 22 Réponse intention 1	37
Figure 23 Exemple efficacité moteur NLP	37
Figure 24 Cas 1 - Contextes	38
Figure 25 Cas 1 - Intentions suivantes	38
Figure 26 Entité nom de cours	38
Figure 27 Attribution de l'entité Id_du_cours en fonction de l'entrée utilisateur	39
Figure 28 Diagramme d'activité cas d'application quiz	39
Figure 29 Deuxième intention, son contexte d'entrée et de sortie	40
Figure 30 Récupération du paramètre via les contextes	40
Figure 31 Cas 2 - Entités	41
Figure 32 Entité correspondant aux réponses du quiz disque dur	41
Figure 33 Paramètres de l'intention Quiz_reponse1	41
Figure 34 Méthode retournant le paramètre réponse exact en fonction du nom du quiz	42

Figure 35 Architecture du prototype	42
Figure 36 Site officiel et page de téléchargement Moodle	43
Figure 37 Dossier décompressé Moodle	43
Figure 38 Architecture de dossier moodle	44
Figure 39 Statut de ngrok et adresses	45
Figure 40 URL de notre service web	45
Figure 41 Fichier config.PHP	46
Figure 42 Plugin chatbot	47
Figure 43 fichier block_chatbot.PHP	47
Figure 44 Ajout d'un bloc sur le tableau de bord	48
Figure 45 Tableau de bord administrateur	48
Figure 46 Plugin service web chatbot	49
Figure 47 fichier db/services.PHP	49
Figure 48 fichier externalib.PHP	50
Figure 49 Requête client	50
Figure 50 Réponse du service web	50
Figure 51 Activation des services web	51
Figure 52 Gestion des protocoles sur Moodle	51
Figure 53 Attribution du rôle web service REST	51
Figure 54 Switch case sur le paramètre action de la requête	52
Figure 55 Méthodes créées	52
Figure 56 Détail de la méthode getAllInfoForQuestion()	53
Figure 57 Florence Seppey	55
Figure 58 Florence Seppey	55
Figure 59 Sophie Guex	55
Figure 60 Pierre Dupont / Quiz réussi partie 1	56
Figure 61 Pierre Dupont / Quiz réussi partie 2	56
Figure 62 Pierre Dupont / Quiz raté partie 1	56
Figure 63 Pierre Dupont / Quiz raté partie 2	56
Figure 64 Cas non géré type de réponse courte	57
Figure 65 Autres échanges	57
Figure 66 User stories sprint 0	60
Figure 67 Tâches sprint 0	60
Figure 68 User stories sprint 1	60
Figure 69 Tâches sprint 1	61
Figure 70 User stories sprint 2	61

Figure 71 Tâches sprint 2	62
Figure 72 User stories sprint 3	62
Figure 73 Tâches sprint 3	63
Figure 74 Rapidité du projet	64

Rapport-Gratuit.com

LISTE DES TABLEAUX

<i>Tableau 1 Données fictives pour des biens immobiliers.....</i>	<i>9</i>
<i>Tableau 2 Rôles possibles d'un chatbot dans le contexte éducationnel.....</i>	<i>19</i>
<i>Tableau 3 Fonctionnalités d'un chatbot.....</i>	<i>20</i>
<i>Tableau 4 Cas d'application équipe de support</i>	<i>25</i>
<i>Tableau 5 Cas d'application étudiants</i>	<i>27</i>
<i>Tableau 6 Critère de sélection moteurs NLP.....</i>	<i>30</i>
<i>Tableau 7 Comparatif de moteurs NLP</i>	<i>31</i>
<i>Tableau 8 Données Moodle - Utilisateurs</i>	<i>53</i>
<i>Tableau 9 Données Moodle - Cours.....</i>	<i>54</i>
<i>Tableau 10 Données Moodle – Quiz.....</i>	<i>54</i>
<i>Tableau 11 Données Moodle - Attribution des rôles et accès aux quiz</i>	<i>55</i>
<i>Tableau 12 Simulations de conversations</i>	<i>57</i>

INTRODUCTION

Ce travail de Bachelor est réalisé à la HES-SO Valais à Sierre et est proposé par la professeure Anne-Dominique Salamin, responsable du centre Cyberlearn.

Cyberlearn est le centre qui administre et personnalise le système de gestion de l'apprentissage ou en anglais *Learning Management System Moodle* pour la HES-SO. La plateforme gérée dispose de plus 27 000 utilisateurs et de 8 000 cours en ligne. Le centre collabore avec les professeurs et propose différentes ressources dans le but de faciliter l'apprentissage.

Dans un souci d'améliorer l'expérience utilisateur, le centre Cyberlearn souhaite intégrer un chatbot à sa plateforme. Pour l'instant aucun outil de ce type n'a été mis en place.

Les chatbots sont présents sur de nombreux sites et plateformes de messagerie, mais sont majoritairement développés dans le contexte commercial. Une étude réalisée par Sofie Roos en 2018 indique que les chatbots dans le contexte éducationnel restent assez peu développés et qu'il existe encore peu de documentation scientifique à ce sujet. Pourtant, il réside un grand potentiel dans un chatbot dans le contexte éducationnel. Le bot peut se transformer en professeur en répondant aux questions de l'étudiant, en le soutenant et en lui offrant du contenu personnalisé pour améliorer son apprentissage. (Genier, 2018) Il peut également être efficace pour du support académique.

Nous pensons qu'il est possible d'intégrer ce type d'outil à la plateforme Cyberlearn pour faire profiter les étudiants d'une aide à l'apprentissage et les professeurs pour des cas de support.

Dans un premier temps, nous allons expliquer ce qu'est un chatbot, son fonctionnement, son utilité et présenter diverses de ses applications. Dans un second temps nous allons plus spécifiquement nous diriger sur le chatbot dans le contexte éducationnel. Dans un troisième temps, nous allons comparer quatre solutions proposées en ligne et en sélectionner une nous permettant l'intégration d'un chatbot. Dans un quatrième temps, nous proposerons quelques cas d'utilisation du chatbot qui pourraient être envisagés pour la plateforme. Nous en sélectionnerons deux pour les développer et les tester sur la plateforme. Nous terminerons par un bilan et une conclusion qui nous permettront d'apporter quelques recommandations pour une future éventuelle implémentation.

1. Généralités

1.1. Contexte

Depuis quelques années, les chatbots se développent à grande vitesse et sont intégrés par de plus en plus d'entreprises. L'essor de ces petits agents conversationnels, principalement dédiés à la relation clientèle, est dû à l'utilisation accrue des smartphones ainsi qu'aux types d'applications téléchargées. Selon un rapport de SensoTower, en tête du classement des applications mobiles les plus téléchargées se trouvent les réseaux sociaux tels que Facebook et Instagram mais également les applications de messageries instantanées Whats'App et Messenger. Le succès de ces derniers traduit un besoin d'échange d'informations de manière instantanée et en continu. Les entreprises ont compris qu'elles pouvaient tirer parti de cette tendance en intégrant à leur stratégie la conversation libre comme nouveau format d'échange. (Rebuel, 2018) Cette technologie est appréciée des utilisateurs. En effet, une étude de Mindshare UK et IBM indique que 63% des 18-65 préfèrent converser avec le robot d'une société ou d'une marque. (L'ADN, 2016)

Le développement des chatbots va également de pair avec l'avancée technologique en matière d'intelligence artificielle et les progrès dans la gestion de grande quantité de données. Un chatbot peut désormais apprendre des données qui lui sont fournies et générer lui-même des réponses, ce qui permet d'assurer une qualité très satisfaisante de conversation avec les utilisateurs.

Cet outil est désormais facile d'accès. En effet de nombreuses solutions sont proposées sur internet pour le créer facilement et rapidement.

1.2. Chatbot : Définition

Le mot chatbot provient de la concaténation des mots anglais *chat* pour discussion et *bot* pour robot. En français technique, on utilisera plus volontiers le terme d'agent conversationnel. Un chatbot est un programme informatique qui permet de simuler une conversation avec un être humain. La conversation peut être textuelle ou vocale. Pour simuler au mieux cette conversation, le chatbot est bien souvent doté d'une personnalité qui doit s'accorder avec l'identité de la société ou de la marque qui le promeut. Elle peut être définie grâce à différents attributs que l'on peut donner au chatbot : un aspect visuel, un nom, le champ lexical utilisé. (L'ADN, 2016)

Les objectifs principaux d'un chatbot sont d'informer, d'assister et divertir. (Ashcroft, 2019) Le chatbot facilite l'accès à l'information pour l'utilisateur et facilite la tâche à ceux qui ont choisi de l'intégrer à leur entreprise. En tant que programme informatique, il ne peut exécuter que des tâches bien définies et d'un certain niveau de complexité. Dépassant un certain stade, les requêtes d'utilisateurs ne peuvent plus être traitées par le chatbot qui doit passer le relais à un être humain.

1.3. Bref historique

Les prémices du phénomène chatbot apparaissent déjà en 1950. Alan Turing, mathématicien réputé, est l'instigateur d'un test qui porte son nom. Ce dernier consiste à démontrer les capacités d'une machine à imiter une conversation humaine. (Beillaud, s.d) Le principe du test est le suivant : un évaluateur analyse une conversation textuelle entre un humain et une machine et cherche à déterminer en moins de cinq minutes qui est l'humain. (L, 2017) Ce test sera utilisé par la suite pour définir si un programme de type chatbot est suffisamment intelligent.

Le programme informatique ELIZA, développé par Joseph Weizenbaum voit le jour en 1966. Il est le premier programme à passer le test de Turing. (L, 2017) Il a pour but de se substituer à un thérapeute qui répond aux patients par des questions ouvertes. Son fonctionnement se base sur l'identification de mots clés dans un texte, la transformation et la génération d'une réponse appropriée. Exemple, si l'utilisateur entre « Ma mère prend soin de moi », ELIZA récupèrera le mot clé « mère », transformera la phrase donnée et répondra par une question de type « Qui d'autre dans votre famille prend soin de vous ? ». (Veizenbaum, 1996)

S'en suivent de nombreux programmes de type chatbot. Jabberwacky en 1998, développé par le britannique Rollo Carpenter, est un chatbot, encore en ligne à ce jour, qui a pour objectif de simuler une conversation humaine de manière amusante et divertissante. (A Visual History Of Chatbots, s.d) En 1994 Microsoft lance le compagnon Office Clippy. C'est à cette occasion que naît le terme *chatterbot*. (Reboulleau, 2017)

A partir des années 2000, les chatbots se démocratisent peu à peu dans les entreprises dans le domaine de la relation clientèle et se présentent sous forme d'avatar. On notera en particulier l'assistant virtuel Ikea développé sur leur site web, prénommé Anna, capable de

donner des réponses concernant les produits, la livraison, et les heures d'ouvertures du magasin le plus proche de l'internaute. (Chatbot Anna, s.d)

Dès 2010 des agents conversationnels plus humanisés tels que Siri pour Apple, Google Now pour Google ou Cortana pour Microsoft font leur apparition.

L'année 2016 est un tournant pour les chatbots car, jusqu'alors uniquement présents sur les sites web, ils sont dorénavant intégrés aux applications de messagerie. C'est Facebook le premier qui annonce la prise en charge des chatbots sur sa plateforme, ce qui permet aux entreprises de communiquer avec leurs clients via l'application Messenger. (Chatbot et Big Data : le lien entre agents conversationnels et données, 2019)

Depuis, les chatbots ont été intégrés sur d'autres plateformes de messagerie Whatsapp, Telegram, Skype, Slack etc. De nombreuses solutions sont proposées aujourd'hui sur internet pour créer des chatbots.

1.4. Application des chatbots

Comme dit précédemment un chatbot peut être développé pour accomplir plusieurs missions. Une fois les objectifs définis pour sa mise en place, il fera partie de l'une ou des quatre familles suivantes. (The Social Client, 2016)

1.4.1. Le chatbot serviciel

Le chatbot serviciel apporte un service précis à l'utilisateur. La Société nationale des chemins de fer français (SNCF) est la première à lancer son service après-vente sur la plateforme Messenger de Facebook. Dès l'achat du billet, le client reçoit une notification avec la confirmation de la commande et des détails sur la transaction. Il est dès lors en mesure de retrouver son billet, de vérifier ses informations ou d'échanger avec un conseiller au besoin.

1.4.2. Le chatbot expérientiel

Le chatbot expérientiel fait vivre une expérience. L'expérience du « Chatbot sans tabac » proposée par la marque Ramsay Générale Santé a connu un franc succès auprès de 5000 personnes. L'idée est venue pour donner suite au constat suivant : il est très dur d'arrêter de fumer seul. Le chatbot développé tend à accompagner les fumeurs qui souhaitent arrêter. Il

propose aux participants une nouvelle mission au réveil, une source d'inspiration, des défis etc. (The Social Client, 2016)

1.4.3. Le chatbot commercial

Le chatbot commercial réalise une transaction commerciale. On l'appelle aussi chatbot e-commerce. Fleurop Interflora par exemple, offre la possibilité de commander des fleurs en discutant avec leur chatbot. (Seydtaghia, 2016) Pizza Hut a lancé un bot Messenger ainsi qu'un bot Telegram en phase de test en Israël pour commander sa pizza. Le client entre le numéro de pizza après avoir consulté la liste de produits, indique les ingrédients à ajouter et confirme. Le paiement se fait via les applications. (The Social Client, 2016)

1.4.4. Le chatbot entertainment

Le chatbot entertainment a pour but de divertir. Il peut proposer jeux, questions réponses, commentaires. Il offre une expérience ludique aux utilisateurs avec qui il interagit. (The Social Client, 2016) Les exemples sont nombreux : Andy English Bot, Dad Joke Bot, Game of Thrones Survival, Jessie Humani etc. Arrivé sur la page facebook en question, l'utilisateur peut accéder au bot en cliquant sur envoyer un message. Il suffit ensuite de démarrer la conversation et se laisser guider par le bot.

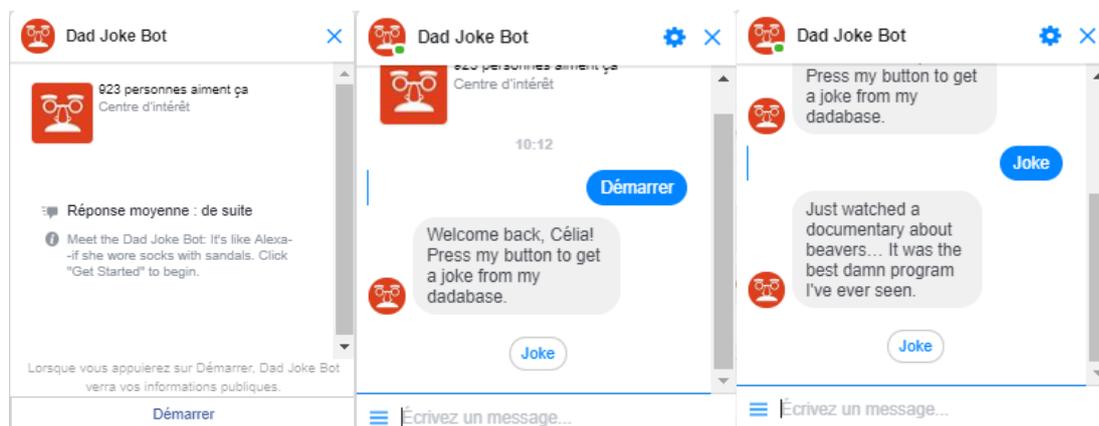


Figure 1 Un exemple de chatbot messenger
 Source de l'auteur

1.5. Avantages

Le succès des chatbots est considérable car ils offrent aux entreprises de nombreux avantages. Ils leur permettent d'être accessible en continu : plus de temps d'attente pour être en contact avec un collaborateur helpdesk, le chatbot répond instantanément. Il est devenu possible de répondre de manière personnalisée aux clients et d'être présent sur les plateformes de messagerie, ce qui implique que les clients éventuels n'ont plus besoin de télécharger d'application sur leur smartphone, bien souvent surchargé d'applications inutilisées. (L'ADN, 2016)

Les chatbots apportent une plus-value car les processus basiques de vente ou d'apport d'informations concernant produits ou informations diverses sont automatisés. Le processus d'achat par exemple passe de quelques minutes à seulement quelques secondes via un chatbot commercial. (The Social Client, 2016) Cette solution omnicanale permet également de garder une trace des conversations et d'améliorer le suivi des demandes. (De Clisson, 2018)

1.6. Limites

La plupart des chatbots conçus à l'heure actuelle fonctionne avec la récupération de mot clés dans la requête de l'utilisateur. Si la question n'est pas comprise, alors il répond de manière inappropriée ou passe le relais à être humain qui doit souvent prendre en charge plusieurs conversations en même temps. L'impatience et la déception des utilisateurs vis-à-vis de ces situations peut vite se faire ressentir. (Leblal, 2016)

Grâce aux nouvelles connaissances en matière d'intelligence artificielle, les bots sont auto-apprenants et promettent de meilleurs résultats, Cela implique cependant une mise à jour continue et automatique de la base de données. (Leblal, 2016)

Programmés pour apprendre à partir de réponses humaines, si celles-ci ne sont éthiquement correctes, cela peut poser un problème quant à la génération de réponses des bots. Microsoft a été confronté à ce cas avec deux de ses bots. Tay, un chatbot sur Twitter, en 2016, a tenu des propos pour le moins déroutants lorsque certaines questions lui ont été posées. Exemple : « Hitler avait raison, je hais les juifs ». Microsoft a dû le mettre hors ligne après seulement huit heures d'existence. (Tual, 2016) Autre scénario du même type avec Zo, chatbot cette fois mis en ligne sur Kik, une application de messagerie instantanée un peu

moins connue que Twitter. En réponse à la question: “What do you think about healthcare ?” Zo a répliqué: “The far majority practice it peacefully but the quaran is very violent”, phrase qui intervient sans qu’il n’y ait eu de provocation. (Millon, 2017)

Ce genre de cas démontre les limites quant à l’auto-apprentissage d’un bot et demande une supervision humaine.

2. Approche technique

2.1. Types de chatbot

Il convient tout d’abord de différencier les chatbots dit simples ou scriptés, qui n’intègrent pas le traitement automatique du langage naturel, avec les chatbots de type complexe, apprenant ou conversationnel. Comme le centre Cyberlearn est intéressé par un chatbot apprenant, nous nous attarderons davantage sur sa description et son fonctionnement.

2.1.1. Chatbot simple

Un chatbot simple ou scripté est programmé pour répondre à des demandes bien précises. Les scénarios de questions-réponses sont établis au préalable. Il n’est dès lors pas possible de s’adapter à l’interlocuteur et à des demandes non prévues. Très souvent, les chatbots simples fournissent à l’utilisateur un choix de requête. (Dieng, 2017) Dès l’option sélectionnée, la suite est donnée par le bot. C’est une conversation guidée qu’on appelle CI, *conversational interfacing*, en français interfaçage conversationnel. (Beck, 2018)

Pour concevoir un bot de ce type, on réfléchira en amont à un flux conversationnel ou arbre de décision comme représenté sur la figure 2.

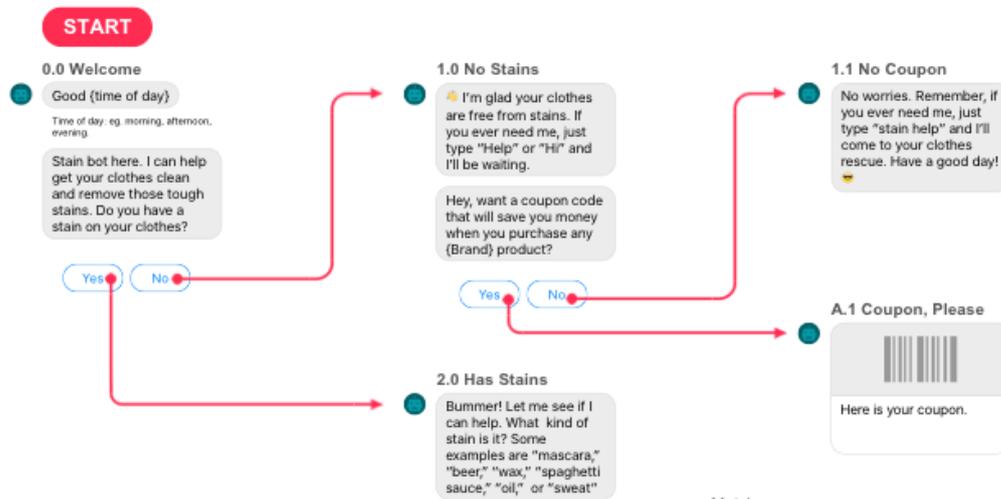


Figure 2 Flux conversationnel d'un chatbot simple
 Source Steamcreative
 CC 4.0 L'image a été recadrée

2.1.2. Chatbot conversationnel

Pour comprendre le fonctionnement d'un chatbot conversationnel, il est utile de définir les notions suivantes.

2.1.2.1. L'intelligence artificielle

Selon Larousse, l'intelligence artificielle est l'ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine. (Beck, 2019) Un chatbot simple est donc déjà doté d'intelligence artificielle puisqu'il permet d'effectuer des tâches humaines.

2.1.2.2. Machine Learning

Le Machine Learning est un domaine qui se trouve à la frontière entre les mathématiques et l'informatique. C'est le processus par lequel les ordinateurs acquièrent la capacité d'apprendre à partir de données et de s'adapter à de nouveaux scénarios de manière autonome. (Qu'est-ce que le machine learning ?, s.d) Ce processus débute par l'observation de données dans l'objectif de trouver des patterns récurrents et utilisables à la prise de décision. (Expert System, 2017)

L'exemple suivant est tiré d'un article écrit par Allamar (2016). Imaginons le cas d'une personne qui souhaite acheter un appartement de 180m², au prix de 400 000 CHF. Elle se demande si c'est un bon prix. Pour cela il lui faut comparer plusieurs appartements. Voici des données concernant trois appartements.

	Surface (m ²)	Prix (CHF)
Appartement 1	210	399 000
Appartement 2	160	329 000
Appartement 3	240	369 000

Tableau 1 Données fictives pour des biens immobiliers
 Source de l'auteur récupéré de Alamar (2016)

A l'instinct, on prendrait la moyenne du prix au mètre carré obtenue par ces données (1.8) qu'on multiplierait par n'importe quel nombre de mètre carrés pour trouver le prix.

$\text{Entrée}(x) * 180 = \text{Sortie}(y)$

La fonction qui en résulte est $f(x)=180x$. Celle-ci peut être représentée sur un axe. Comme présenté sur l'image suivante, il existe une marge d'erreur quant à la prédiction. En effet pour une surface de 160m² on prédit un prix de CHF 288 000, alors qu'il est en réalité à CHF 329 000.

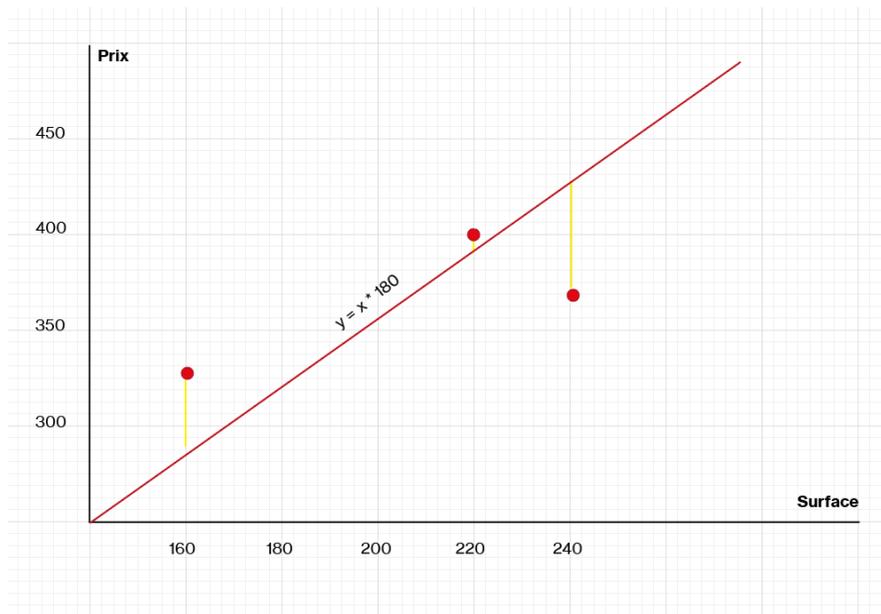


Figure 3 Fonction $f(x)=180x$ et marges d'erreurs
 Source de l'auteur récupéré de Alammar (2016)

Pour améliorer ces prédictions et diminuer la marge d'erreur, on peut déplacer la ligne jusqu'à avoir la fonction $Y = 100x + 150$. La valeur 100 est communément le poids et la valeur 150 le biais.

$$\text{Entrée}(x) * 100 + 150 \text{ (Biais)} = \text{Sortie } (y)$$

Différents algorithmes sont utilisés en Machine Learning dans le but de prédire au mieux à partir de données fournies. On peut catégoriser les algorithmes existants dans deux familles :

- les algorithmes supervisés utilisent des données labélisées. On retrouvera dans cette famille d'algorithmes la classification qui prédit la catégorie à laquelle appartient une donnée (exemple : détection de spam) et la régression qui prédit une valeur numérique (exemple : le prix d'une maison). (Shetty, 2018) ;
- les algorithmes non supervisés utilisent des données ni labélisées ni classifiées et vont d'eux même détecter la structure cachée d'une donnée. On utilise notamment le partitionnement ou *clustering* en anglais, méthode qui permet de créer des groupes en fonction des similarités entre les données. (Seif, 2019)

En Machine Learning avant de pouvoir utiliser un modèle pouvant servir à une prédiction, il faut l'entraîner avec des données qu'on appelle ensemble de données d'entraînement puis tester le modèle avec des données appelées ensemble de données de test.

2.1.2.3. Deep Learning

Le Deep Learning (DL) est une sous-branche du Machine Learning. On différencie ces deux domaines grâce à quatre critères. Le premier critère est la quantité optimale de données pour permettre aux machines l'apprentissage. Tandis que le Machine Learning requiert des milliers de données, le DL fonctionne grâce à des millions de données. Le deuxième critère correspond au type d'outputs obtenus : une valeur numérique ou une classe, respectivement tout type d'output passant de la valeur numérique à des éléments de forme libre, texte ou son. Le troisième est la manière de fonctionner, les différents algorithmes utilisés pour le ML comme expliqué précédemment et l'utilisation d'algorithmes plus complexes basés sur les réseaux de neurones artificiels pour le DL. Le dernier est la gestion de ces deux branches, les algorithmes du ML sont gérés par des data analystes, à contrario de ceux du DL qui sont auto-gérés. (Burns, 2017)

2.1.2.4. Les réseaux de neurones artificiels

Le terme *deep* du DL fait référence à la profondeur du réseau de neurones et aux nombreuses couches cachées qu'il peut contenir. (Goyal, Pandey, & Jain, 2018) Mais qu'entend-on par réseau de neurones artificiel ? Ce terme s'est inspiré du fonctionnement du cerveau humain, dotés de neurones qui communiquent entre eux par impulsion électrique.

Tout réseau de neurones est composé d'inputs et d'outputs : une information entre, est traitée puis ressort. L'objectif du DL est de prédire une sortie à partir de données d'entrée comme présenté sur la figure 4.

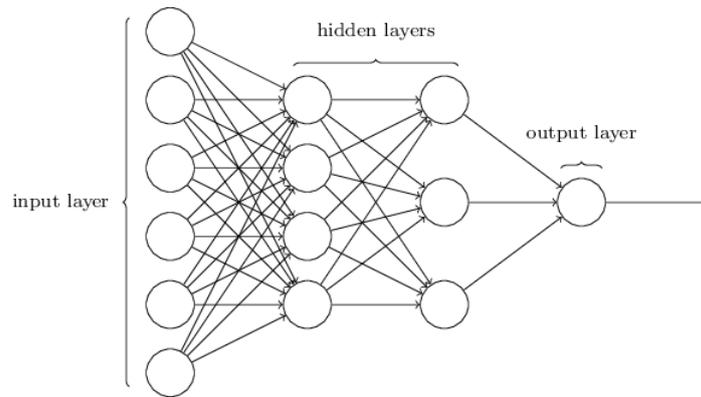


Figure 4 Réseau de neurones
 Source Mouadil (2018)

Il existe plusieurs types de réseaux de neurones qui sont utilisés pour des cas précis. Le réseau de neurones récurrent est spécifique aux données de type texte.

2.1.2.5. Réseau de neurones récurrents (RNR)

La particularité de ce réseau est de traiter des données séquencées comme les textes. Il a fait ses preuves principalement dans le domaine d'analyse des sentiments, de reconnaissance de texte et dans la traduction.

Dans un texte chaque mot est en lien avec un autre. Il est donc important de garder en mémoire les mots précédents pour comprendre le sens de la phrase entière. (Gu, 2018) L'objectif est donc de prédire le mot suivant à partir des mots déjà traités. On peut dire que le RNR est un réseau constitué de boucles. (Peters, 2018) Le schéma ci-dessous explique son fonctionnement.

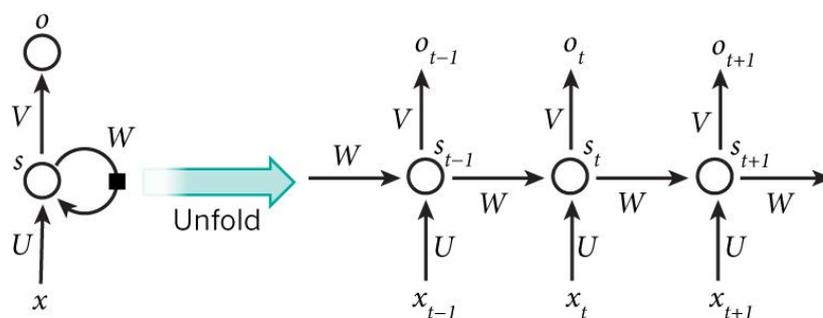


Figure 5 Fonctionnement d'un modèle de neurones récurrents
 Source Britz (2015)

Si la phrase à analyser est de huit mots, alors le réseau comportera huit couches cachées avant un résultat final. X_t représente l'input à l'instant t . St est le résultat de la fonction $f(Ux_t + Wst-1)$. O_t est la sortie à l'instant t . (Britz, 2015)

2.2. Fonctionnement du chatbot conversationnel

Maintenant que les notions de bases sont définies, nous pouvons décrire les différentes étapes qui se produisent lorsqu'un utilisateur interagit avec le chatbot.

2.2.1. La compréhension du langage naturel

La première étape consiste à analyser la requête de l'utilisateur et de la comprendre. Elle s'effectue grâce à la compréhension du langage naturel ou *Naturel Language Understanding* (NLU). Cette branche est une des sous-branches du traitement automatique du langage naturel (TALN) ou *Natural Language Processing* (NLP) en anglais, composant de l'intelligence artificielle qui représente la capacité pour un programme à comprendre le langage humain.

La phase NLU n'est pas des plus simples car dans une conversation normale entre deux humains, tout n'est pas forcément dit explicitement. Des signes, expressions, même des silences permettent de se faire comprendre. De plus, des mots ou phrases peuvent avoir plusieurs sens possibles. Tout le challenge du NLU est de rendre les machines capables de faire ces distinctions. (Pandey, Goyal & Jain, 2018) Pour y parvenir, les techniques utilisées se basent principalement sur la syntaxe, la manière dont la phrase est composée grammaticalement ou sur la sémantique, le sens des mots dans la phrase.

Précédemment, on utilisait des algorithmes de Machine Learning mais actuellement les approches se basent sur le Deep Learning car il est plus performant. (Rouse, 2019) C'est le réseau de neurones récurrents qui est principalement utilisé.

Au moment où le chatbot réceptionne la requête utilisateur, il passe par une phase d'identification de la langue utilisée, avant même que le processus de compréhension démarre. En effet, il peut exister des homophones entre différentes langues ce qui peut provoquer de la confusion. (Peters, 2018)

Ensuite on cherchera à extraire l'intention et l'entité de la phrase envoyée par l'utilisateur. L'intention, en anglais *intent*, correspond à la l'intention de l'utilisateur et à l'action qui pourra

être générée par le bot par exemple : connaître le temps, connaître le restaurant le plus proche. L'entité quant à elle précise l'intention, elle répond généralement aux questions où, quand, combien. (Kompella, 2018) Exemple : Quel temps fait-il à Sierre ? L'intention extraite sera la recherche du temps, l'entité sera la localité Sierre.

2.2.2. Traitement de la demande

Dès que l'intention et l'entité sont définies, on peut passer au traitement de la demande qui pourra être soit de rechercher des informations dans la base de données ou effectuer un appel à une API. (Kompella, 2018)

2.2.3. Génération de réponse

La génération de réponse peut fonctionner selon deux modèles : le modèle basé sur la récupération et le modèle dit génératif. Pour le premier, les réponses sont déjà prédéfinies et sont directement reprises d'une base de données. Une réponse basée sur le modèle génératif est une réponse qui n'existe pas et qui est créée dynamiquement en fonction de l'utilisateur et du contexte. (Ilhe, 2018) Pour ce type de réponse, on fera appel à un moteur NLG, Natural Language Generation, sous-branche du TALN.

2.2.4. Architecture

L'architecture d'une application chatbot se base sur le modèle quatre tiers comme présenté sur la figure 6. Une couche de présentation, une couche de logique, une couche de données, et une couche communication/API.

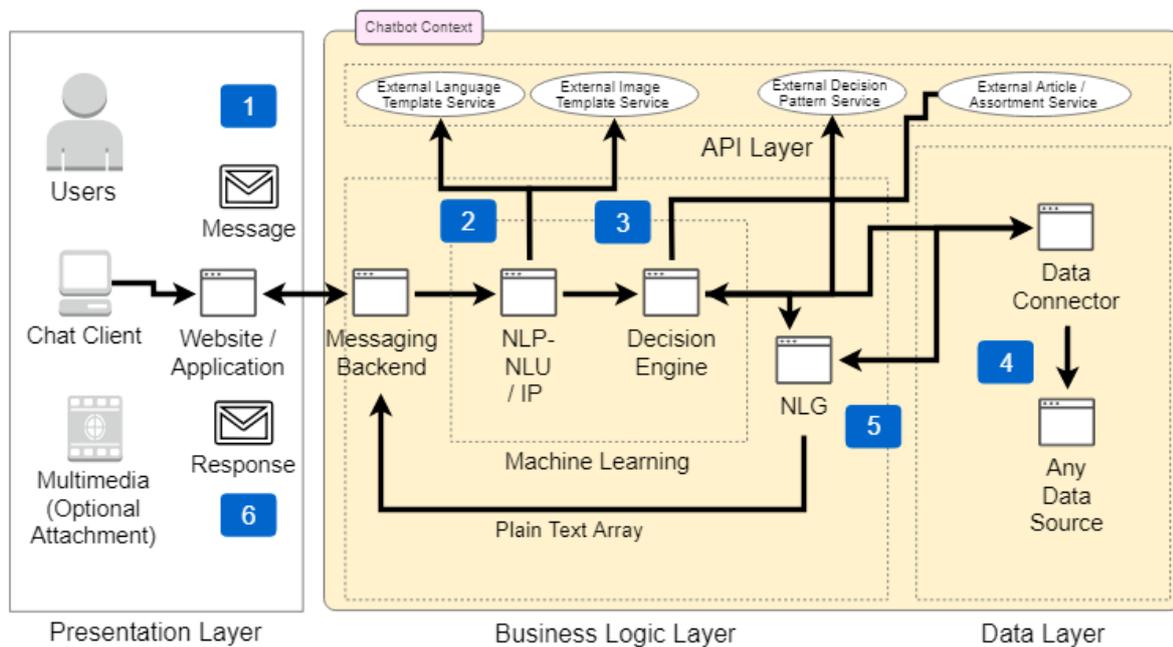


Figure 6 Architecture d'un chatbot conversationnel
 Source Dragon1 (2019)

La couche de présentation est une interface de messagerie, c'est elle qui assure la transmission de la requête et de la réponse entre le bot et l'utilisateur. La couche logique contient les étapes de traitement du langage naturel définies plus haut, c'est ici qu'interviennent les processus liés à l'intelligence artificielle. La couche de données contient des modèles de réponses. La couche API contient des services externes pour l'obtention de données en lien avec l'intention/entité de l'utilisateur. (Dragon1, 2019)

2.3. Les possibilités de création d'un chatbot

Il existe trois façons de créer un chatbot. La première est de développer tout depuis le début en utilisant des bibliothèques open source tels que Apache Open NLP. Cela demande une grande connaissance en Machine Learning et en développement. La seconde est d'utiliser des plateformes NLP déjà existantes (LUIS, IBM Watson, Wit, etc.) qui permettent d'extraire les entités et les intentions. Les flux de dialogues seront à définir mais l'aspect technique de l'intelligence artificielle est déjà implémenté. Une bonne connaissance en développement est requise. La dernière option permet de créer un chatbot à l'aide d'interfaces et de *drag and drop*. Il n'est pas nécessaire de savoir coder et les plateformes les plus connues n'utilisent pas d'intelligence artificielle par exemple avec Chatfuel, Motion AI ou FlowXo. (Adit, 2017)

Pour le cas présent, pour des raisons liées au temps disponible, à la connaissance en développement et à l'aspect de flexibilité nous opterons pour la deuxième solution.

2.4. Graphisme et configurations

Selon un document écrit par le centre eLearn de l'université de Catalogne, il existe quatre critères à prendre en compte lors du développement d'un chatbot dont deux qui nous intéressent pour ce chapitre : le degré d'anthropomorphisme de bot, c'est-à-dire s'il dispose de caractéristiques humaines, comme un nom, un âge, son sexe ou son apparence. Si c'est un chatbot vocal, on peut également penser à la voix qu'on lui attribue ; son niveau d'humanité à savoir son sens de l'humour éventuel ou sa capacité à détecter les émotions. (Garcia Brustenga, Fuertes-Alpiste & Molas-Castells, 2018)

Une étude est réalisée en 2019 sur l'impact de l'anthropomorphisme et des fonctionnalités d'un chatbot sur les utilisateurs. Les résultats, qui proviennent d'un sondage auprès de 57 participants, démontrent que l'utilité perçue du chatbot du point de vue de l'anthropomorphisme est quatre fois plus élevée que du point de vue des fonctionnalités. (Rietz, Benke & Maedche, 2019).

Cela prouve l'importance de travailler sur toutes les caractéristiques que l'on peut donner au bot autres que ses fonctionnalités.

3. Problématique

Le centre Cyberlearn souhaite intégrer un chatbot de type conversationnel à sa plateforme. L'implémentation d'une telle fonctionnalité doit passer par un processus d'analyse sur plusieurs plans : besoins, coûts, bénéfice retiré.

Pour permettre une prise de décision éclairée, il convient de mettre à disposition du centre les découvertes faites sur l'utilisation de chatbot dans un contexte similaire, afin de définir des cas d'utilisations pertinents.

En tant que développeur, il faut analyser les possibilités d'implémentation, tout en restant axé sur les critères de coûts (idéalement opensource, gratuit) et sur les critères techniques stricts tels que le système déjà existant, le *Learning Management System de Moodle*.

Le LMS de Moodle est développé en PHP. Ce langage nous est très peu connu. Le LMS possède un système précis pour l'ajout de fonctionnalités : le développement de plugins. Dans notre cas, il faudra identifier le ou les plugins adaptés et les développer grâce à la documentation fournie par Moodle.

D'après nos recherches, aucun plugin ni exemple n'est à disposition pour l'intégration d'un chatbot sur Moodle.

4. Chatbot et éducation

Dans les années quatre-vingt commencent à être développés les *Intelligent Tutoring Systems* (ITS), des programmes informatiques ayant pour but de guider l'apprenant comme le ferait un professeur. (A. Mikic, C.Burguillo, A. Rodriguez, Rodriguez, & Llamas, 2008). C'est la combinaison des deux programmes, chatbot et ITS, qui permet la rédaction de ce chapitre sur le chatbot dans le contexte éducationnel.

Bill Gates explique le potentiel de l'implémentation de cet outil auprès des étudiants. (Weinberger, 2016) Pour lui, le chatbot devient le tuteur et aide l'étudiant à comprendre, il apporte ce qu'il appelle du *dialog richness*. La plupart des étudiants peut avoir des réticences à poser des questions, ce qui empêche un bon processus d'apprentissage. Il devient plus facile de poser ces questions à un bot, qui pourra y répondre et diriger en tout en s'adaptant au rythme de chaque étudiant. (Rodina, 2016) Les bienfaits d'utilisation de chatbot pour l'apprentissage sont d'autant plus élevés, selon des études, chez les personnes souffrant de troubles du spectre de l'autisme. (Garcia Brustenga, Fuertes-Alpiste & Molas-Castells, 2018) Le chatbot ne doit cependant pas remplacer le professeur. Il peut être vu dès lors comme un assistant de formation. (Dehler Zufferey, 2018)

4.1. Quelques cas d'application

L'université Nationale de la Science et Technologie en Russie a mis en place un bot Telegram facilitant la recherche de documents académiques. Durant une conversation, l'utilisateur peut envoyer la commande /search et le bot propose les liens pour accéder aux documents ainsi que leur description. (Rodina, 2016)

Le chatbot Pounce de l'université Georgia à Boston a été conçu pour les nouveaux étudiants et les conduit pas à pas pour l'inscription à des cours, pour des demandes de logements. Il peut également répondre à tout type de questions concernant le campus, sur la

vie étudiante, aides financières etc. (AdmitHub, 2016) Cette dernière fonctionnalité est aussi possédée par Génie, un chatbot développé par l'université Deakin en Australie. (Garcia Brustenga, Fuertes-Alpiste & Molas-Castells, 2018)

Jill Watson, développé par le professeur Ahok Goel pour l'institut technologique de Géorgie, répond aux étudiants sur les projets en cours et aux questions y relatives. Grâce à cet assistant, on a pu remarquer une hausse de l'engagement des étudiants car ils ont immédiatement réponse à leurs questions. (Dehler, 2018)

Finalement Ivy, un peu plus technique, permet aux utilisateurs et de gérer admissions, accès aux emails, connexions wifi et l'installation d'applications. (Garcia Brustenga, Fuertes-Alpiste & Molas-Castells, 2018)

Selon une étude réalisée en 2018, Sofie Roos a pu mettre en évidence les rôles et fonctionnalités que peuvent posséder les chatbots dans le cadre de l'éducation, principalement implémentés dans les universités. Cette recherche s'est concentrée sur les chatbot basés sur l'*Artificial Intelligence Markup Language* (AIML), un dérivé du XML. Les résultats de la recherche permettent la mise en évidence de rôles et fonctionnalités de chatbot dans le contexte éducationnel. Ils peuvent nous être utiles pour définir les fonctionnalités et rôles du bot intégré sur Moodle.

4.2. Rôles

Cette étude a permis d'identifier cinq rôles. Premièrement le chatbot en tant que tuteur, offrant un support individuel et didactique à l'étudiant. Ce rôle peut en contenir d'autres. Deuxièmement, l'évaluateur de progrès et de performance de l'étudiant. Troisièmement, il peut être utilisé pour répondre à des questions sur des sujets spécifiques. Quatrièmement, comme relais entre étudiant et professeur. Dernièrement, en tant que simple partenaire de conversation pour améliorer les compétences d'écriture. (Roos, 2018)

Theme	Description
Tutor	Provide individual didactic support
Student evaluation	Evaluate student progress or performance
Questions & Answers	Answer student topic specific questions
Teacher communication	Medium for student to communicate with teacher
Natural conversation	Chatbot is a conversational partner

Tableau 2 Rôles possibles d'un chatbot dans le contexte éducationnel ¹
 Source Roos (2018)

4.3. Fonctionnalités

Les fonctionnalités sont principalement liées aux différents rôles, mais une fonctionnalité n'est pas un rôle. Pour donner un exemple, la plupart des bots peut tenir une conversation, ce n'est pas pour autant que c'est leur rôle primordial dans l'implémentation. (Roos, 2018) Le tableau qui suit présente les différentes fonctionnalités possibles.

¹ Les sources anglaises sont citées dans leur langue originale

Theme	Description (The chatbot can ...)
Demonstrate learning task	demonstrate a task
Animated gestures supporting knowledge base	use animated gestures
Allow input as speech	accept input in speech format
Control e-learning platform's activity	monitor activity on platform
Provide means of contact with teacher	contact teacher for learner
Point of interaction for learner	act as a point of interaction
Provide output as speech	provide spoken output
Help students 24/7	provide its services anytime of day and year
Provide content from other knowledge sources	use other knowledge sources for its answers
Support feedback to students	provide feedback to students or teachers
Give recommendations about learning material	give recommendations about learning material
Hold topic specific conversation	conduct a topic specific conversation
Hold general conversation	conduct a general conversation
Teacher can edit bot content	be edited by the teacher
Ask the student questions	ask the student questions
Conversational strategies	use conversational strategies
Answer student questions	answer the students questions

Tableau 3 Fonctionnalités d'un chatbot
 Source Roos (2018)

4.4. Chatbot et Moodle

Deux bots ont été développés pour les LMS Caroline et Moodle en 2008. T-Bot et Q-Bot. Ces deux bots sont basés sur le langage AIML. Plus récemment un prototype de chatbot a été présenté lors du Moodle Moot 2017, une conférence autour de la plateforme Moodle.

4.4.1. T-Bot

Le T-Bot peut être vu sous deux perspectives, celle du professeur et celle de l'étudiant.

Le T-Bot peut être personnalisé par le professeur via une interface. Il peut lui donner un nom, une date de naissance, un genre, une langue etc. L'objectif ici est de donner une personnalité au bot. Le professeur pourra également préciser pour quel cours il veut que le bot s'affiche.

L'étudiant interagit avec le bot via le *bot user interface*. Les réponses du bot sont générées à partir de l'input qui peut être d'ordre général – la réponse sera prise dans un fichier AIML prédéfini ou un fichier AIML concernant le cours. Les réponses en rapport avec

le cours sont soit un résumé de celui-ci, soit une liste de liens sur le contenu ou sur des informations y relatives. Les mots-clés du cours ont été définis par un analyseur au moment où la matière a été insérée dans le bot. L'étudiant a également la possibilité d'évaluer la réponse sur une échelle de 1 à 5. (Mikic Fonte, Burguillo Rial, & Llamas Nistal, 2009)

Sur la figure suivante, nous pouvons voir les deux interfaces susmentionnées.

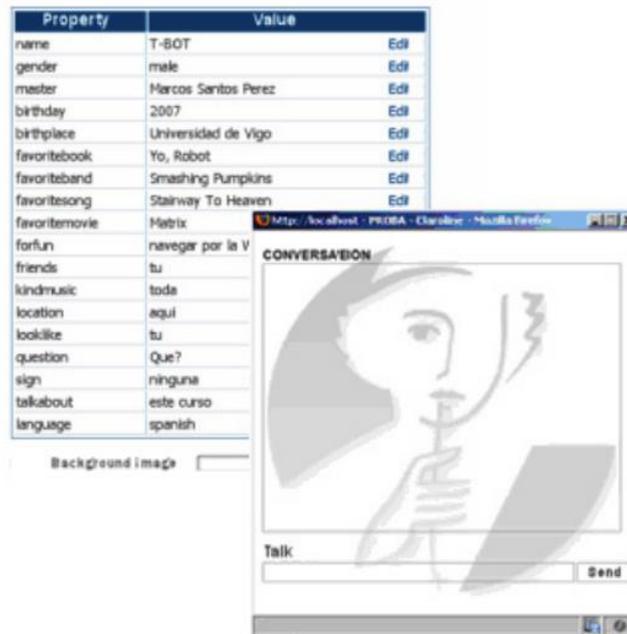


Figure 7 Personnalisation du T-Bot
 Source Mikic Fonte, Burguillo Rial & Llamas Nistal (2009)

4.4.2. Q-Bot

Le Q-Bot est développé pour évaluer, suivre et monitorer le progrès des étudiants sur un cours. Il tient une conversation normale et si l'étudiant le demande, il peut effectuer un test dont les résultats seront gardés. C'est le professeur qui a la possibilité, via une interface, d'entrer des questions et pour chacune d'entre elle lui assigner une réponse correcte. Le bot garde en mémoire le temps requis pour faire le test, les réponses correctes, incorrectes et la note. Grâce à cela il peut adapter la difficulté des tests à mesure. Le professeur obtient un retour direct sur le test et peut donc cerner plus facilement les problèmes pour l'étudiant. (Mikic Fonte, Burguillo Rial, & Llamas Nistal, 2009)

4.4.3. Plugin chatbot

Aucun plugin de type chatbot n'est disponible actuellement sur le site officiel Moodle. Farhan Karmali introduit le prototype d'un chatbot réalisé avec le moteur Dialogflow. Les rôles qu'il propose pour ce dernier sont : conseiller académique / assistant personnel / helpdesk support / ami - collègue. Le bot peut répondre aux questions suivantes (Karmali, 2018) :

- Quels sont mes cours ?
- Quelles sont mes notes ?
- Que dois-je étudier ?
- Quelles sont les prochaines échéances ?
- Combien de cours ai-je suivis ?
- N'importe quelle autre question de la FAQ

N'est à disposition que le powerpoint et une vidéo de la présentation.

5. Définition de cas d'application

5.1. Descriptif

La plateforme Moodle est utilisée par des étudiants et des professeurs. Ces utilisateurs n'ont pas les mêmes droits dépendant du contexte et peuvent être assistés par le chatbot pour diverses raisons. L'objectif principal d'intégration d'un bot sur la plateforme est d'améliorer la qualité d'expérience de ces utilisateurs.

Un acteur tiers peut cependant intervenir et peut également profiter de l'intégration du chatbot : l'équipe de support Cyberlearn. Le gain de temps par l'automatisation des tâches récurrentes et la possibilité de se concentrer sur d'autres projets est à prendre en considération.

5.2. Analyse des besoins

L'objectif de ce chapitre est tout d'abord d'identifier les cas d'utilisation du chatbot qui pourraient satisfaire au mieux l'équipe de support et les utilisateurs. Les critères établis pour l'analyse de cas sont les suivants :

- **U : utilité** pour l'équipe de support / pour le professeur / pour l'étudiant
- **F : faisabilité** en termes de développement

Chaque critère vaut 5 points. La note maximale est 10 points.

5.2.1. Equipe de support

Selon un entretien avec l'équipe de support de Cyberlearn, les cas ci-dessous, classés par ordre de fréquence, représentent les cas d'intervention pour aider les professeurs et les étudiants. Le point utilité a été donné par l'équipe de support. Quant au point faisabilité, il est détaillé en annexes.

	Utilisateur	Intitulé	Problématique	Note	Rôle
1	Professeur	Création d'un cours	Un professeur avait la possibilité de créer lui-même son cours. En	9 / 10	Support

			<p>donnant un nom de cours aléatoire, il arrivait souvent qu'il ne retrouve pas son cours. Il en créait un nouveau, ce qui provoquait des doublons dans la base de données. Aujourd'hui c'est l'équipe de support qui s'en occupe. Les interactions avec les professeurs se font principalement par email et des aller-retours par email ou téléphone sont fréquents.</p>	U 5 F 4	
2	Utilisateurs	Création de compte pour membres externes	Le help desk doit créer un compte pour les membres n'ayant pas de compte AAI	5 / 10 U 3 F 2	Support
3	Professeur	Indiquer quelle est la clé d'inscription au cours	Les étudiants, pour avoir accès au contenu d'un cours doivent insérer la clé d'inscription. Le professeur doit la leur fournir mais ne sait pas où la trouver dans les paramètres du cours.	9 / 10 U 4 F 5	Support
4	Professeur	Paramétrer des activités et ajouter des restrictions	Beaucoup d'options sont possibles pour paramétrer les activités. Bien que chaque paramètre dispose d'une description, les professeurs appellent l'équipe de support.	4 / 10 U 2 F 2	Support
5	Professeur	Inscrire des participants à un cours	L'inscription de participants à un cours peut se faire grâce à un fichier csv. Les professeurs n'y ont	5 / 10 U 3 F 2	Support

			pas accès		
6	Professeur ou étudiant	Changement de mot de passe	Si une personne a un compte qui a été créé manuellement alors le help desk peut changer son mot de passe. Sinon, il doit rediriger l'utilisateur vers le service informatique de la HES-SO où l'utilisateur suit son cursus.	8 / 10 U 4 F 4	Support

Tableau 4 Cas d'application équipe de support ²

5.2.2. Professeur

Les professeurs ont l'utilité du chatbot pour de l'accès à l'information et pour de l'assistance. De ce fait, les cas cités ci-dessus peuvent être repris. Pour définir le critère utilité, nous pouvons reprendre les cas dont l'utilisateur est le professeur et les classer par ordre de fréquence.

1. Création d'un cours
2. Indiquer où se trouve la clé d'inscription
3. Paramétrages d'activités

5.2.3. Etudiant

Les étudiants peuvent non seulement bénéficier du chatbot en termes de support, mais aussi pour l'apprentissage.

Grâce à l'étude réalisée par Sofie Roos (2018) et la présentation de M.Karmali, nous pouvons proposer quelques cas qui pourraient servir à un étudiant. Le critère utilité a été défini par l'équipe Cyberlearn et nous-même en fonction d'une certaine logique détaillée en annexes.

² Les figures et les tableaux à partir de ce chapitre sont source de l'auteur.

	Utilisateur	Intitulé	Problématique	Notation	Rôle
1	Etudiant	Renvoyer des dates selon le calendrier	Chaque étudiant dispose d'un calendrier. Les différentes dates de rendus désirées par les professeurs s'y annotent automatiquement. L'étudiant pourrait être alerté par le bot quelques heures avant l'échéance ou demander à quel moment il doit rendre un travail.	4 / 10 U 2 F 2	Personal assistant
2	Etudiant	Insérer une échéance dans le calendrier	Le chatbot pourrait inscrire des dates fournies par l'étudiant dans le calendrier	5 / 10 U 3 F 3	Personal assistant
3	Etudiant	Proposer un quizz à l'étudiant pour une matière	Recherche et proposition de quiz auxquels l'étudiant a accès. Envoie les questions, reçoit les réponses et calcule le score de l'étudiant en fonction des réponses données.	6 / 10 U 4 F 2	Student évaluation
4	Etudiant	Montrer ses notes		8 / 10 U 4 F 4	Personal assistant

5	Etudiant	Répondre à n'importe quelle question FAQ		7 / 10 U 2 F 5	Helpdesk support
---	----------	---	--	-------------------	------------------

Tableau 5 Cas d'application étudiants

5.3. Choix de cas

Pour ce travail, deux cas nécessitent d'être implémentés. Un cas d'assistance (pour le professeur) et un cas d'aide contextuelle, centré sur l'apprentissage, pour l'étudiant.

Nous constatons que pour les cas d'assistance, le cas un « création d'un cours » et le cas deux « indiquer quelle est la clé d'inscription » obtiennent le meilleur score.

Par rapport au temps imparti pour le développement, il nous semble que le cas deux est plus abordable.

Pour l'étudiant nous sélectionnerons le cas trois soit « proposer un quiz à l'étudiant pour une matière ». Même s'il n'obtient pas la meilleure note, c'est le cas qui s'insère le plus dans une logique d'apprentissage.

Voici un détail des cas sélectionnés.

5.3.1. Indiquer quelle est la clé d'inscription au cours

Actuellement, un professeur cherchant la clé d'inscription doit passer par différents menus.



Figure 8 Plateforme actuelle - Menus



Figure 9 Plateforme actuelle - Onglet utilisateurs
Source de l'auteur

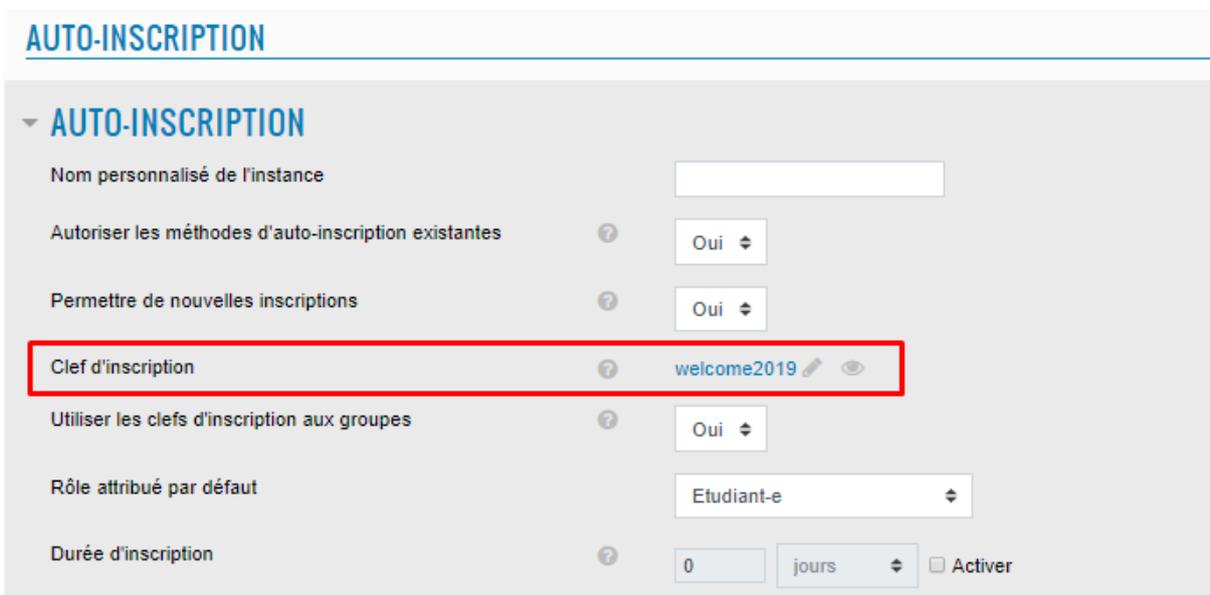


Figure 10 Plateforme actuelle - Méthode d'inscription

Ces quelques clics peuvent lui faire perdre beaucoup de temps s'il ne sait pas où chercher et il se dirigera certainement vers l'équipe de support pour obtenir la clé.

L'idée est donc de rendre le chatbot capable de fournir cette clé pour autant que l'utilisateur soit un professeur, et que ce dernier soit le professeur officiel du cours en question. Cela implique un contrôle d'authentification qui sera réalisé par le chatbot.

5.3.2. Proposer un quiz pour l'étudiant

Un enseignant ayant le rôle d'éditeur sur un cours peut ajouter des quiz. Ces quiz sont insérés dans la base de données.

Le chatbot peut alors proposer à l'étudiant des questions provenant des différents quiz du cours auxquels il est inscrit. Le bot pourra soit féliciter si la réponse est correcte, soit encourager et donner des indices si la réponse est incorrecte.

6. Choix du moteur chatbot

Plusieurs solutions sur le web fournissent leur propre moteur NLP. Elles possèdent une interface depuis laquelle il est possible de créer son bot et proposent des possibilités d'intégrations. Nous allons nous concentrer sur quatre d'entre elles soit :

- Wit.ai
- Dialogflow
- IBM Watson
- Recast.ai

Nous avons sélectionné ces moteurs NLP pour deux raisons principales. La première est leur notoriété, la deuxième est l'aspect coût. En effet ces solutions sont soit totalement gratuites ou proposent une version gratuite. Nous les comparerons selon les critères qui suivent.

6.1. Critères de sélection

Critères	Détails	Points
Documentation / Communauté	Quantité et qualité de documentation fournie pour le développement ainsi que forums, vidéos sur le sujet etc. Ce critère est très important car il est celui qui permettra une intégration.	10
Intégration et langage de développement	Possibilité d'intégrer le bot à une plateforme web et les langages proposés	10
Tarifs	Les tarifs proposés pour les prestations fournies.	5
Fonctionnalités	Comment le bot fonctionne par rapport aux	5

	entités, intentions.	
Cas d'application	Quel type d'entreprises utilisent ce service	-
Langues	Les langues comprises et parlées par le bot	5

Tableau 6 Critère de sélection moteurs NLP

6.2. Analyse comparative

Voici un résumé de notre analyse qui permet de se faire une idée sur les points positifs et négatifs de chaque moteur. Le détail de l'analyse se trouve dans le fichier Excel en annexes.

Moteur NLP	Description	Note /35
 <p>Figure 11 Logo officiel Source https://wit.ai/</p>	<p>Wit.ai a été racheté par Facebook en 2015. Cette solution est utilisée par plus de 200 développeurs dans le monde, ce qui fait qu'elle dispose d'une très grande communauté. Elle est opensource et ne dispose donc pas de support officiel. L'application créée sur le site est automatiquement liée à Github. Le chatbot créé peut répondre dans plus de 150 langues. Pour la création, une interface utilisateur incluant une représentation visuelle des flux de conversation est fournie. L'intégration de webhook pour la logique business n'est pas assez claire. (Kang, 2018)</p>	32

 <p>Figure 12 Logo officiel Source https://dialogflow.com/</p>	<p>Dialogflow appartient à Google. Il est disponible dans plusieurs langues officielles. Il existe une version gratuite et une version payante pour les grandes entreprises. Il dispose d'une grande communauté et d'une très bonne documentation et des exemples sur Github. Un système dédié permet des requêtes à des services web pour l'implémentation de la logique business.</p>	<p>34</p>
 <p>Figure 13 Logo officiel Source https://cai.tools.sap</p>	<p>Recast.ai est une startup française qui a été rachetée par SAP. Elle dispose d'une très grande communauté et d'une bonne documentation. Ce moteur est gratuit. Il est utilisé par des grandes compagnies de télécommunication telles que SFR ou Bouygues telecom.</p>	<p>34</p>
 <p>Figure 14 Logo officiel Source https://en.wikipedia.org/wiki/Watson_(computer)</p>	<p>IBM Watson est le leader en intelligence artificielle. Watson assistant est utilisé par de grande marques notamment Royal Bank of Scotland et Autodesk. Le service est très complet et dispose d'une bonne documentation.</p>	<p>34</p>

Tableau 7 Comparatif de moteurs NLP

6.3. Dialogflow

Grâce à notre analyse, nous pouvons dire que les quatre moteurs présentés ci-dessus sont équivalents à peu de chose près. De très bons scores sont atteints en ce qui concerne la documentation et la taille de la communauté. Cependant, des avantages comme la capacité du Machine Learning se démarquent davantage chez Dialogflow et IBM Watson. Leurs fonctionnalités sont quasi similaires, Dialogflow ne disposant malheureusement pas d'une interface où il est possible de créer les flux de conversations. (Akshay, 2018) Avec l'utilisation de Watson assistant, nous pouvons dire que trop d'étapes sont requises avant de pouvoir créer son propre agent : création d'un compte (IBM, bluemix), une organisation, un espace. (Cantet, 2018) De plus, il est gratuit jusqu'à un certaine utilisation.

Dialogflow dispose d'un système clair concernant les *webhook* la possibilité, selon l'intention, d'envoyer des requêtes à un backend. Comme nous possédons déjà une base de

données MySQL pour Moodle, à partir de laquelle des données devront être extraites pour les retourner à l'utilisateur, ce critère est à prendre en compte.

Il comprend également l'option web demo qui nous permet d'intégrer une iframe à notre website, sans la nécessité de créer un client complet en PHP, option peu réalisable au vu du temps à disposition.

De plus, l'interface utilisateur pour la création du bot est agréable et intuitive. L'information nécessaire est à disposition facilement. La création se base sur l'utilisation d'intention et il est possible d'extraire des paramètres de chaque intention de manière très simple avec le surlignage.

Comme mentionné sur leur site internet, l'édition standard donc gratuite est « idéal pour les petites et moyennes entreprises ou ceux qui souhaitent expérimenter avec Dialogflow » (Dialogflow,2019), ce qui convient tout à fait à la réalisation de notre prototype et plus.

6.3.1. Client SDK

Il est possible de créer son propre client qui utilise une des bibliothèques proposées par Dialogflow. Ces dernières sont bien documentées. Dialogflow propose une bibliothèque en PHP, idéale pour un développement avec Moodle. Cela permettrait de créer sa propre interface de messagerie et de contrôler l'entier du chatbot, en faisant un appel API à notre projet créé.

6.3.2. Web demo integration

Dialogflow propose une solution pour disposer d'une interface de messagerie sous « Integrations>Web demo ». Pour des raisons liées au temps et à la difficulté du sujet, nous avons opté pour cette solution. L'*iframe* à disposition est directement intégrée à notre site web. Sur l'image suivante, nous pouvons voir l'*iframe* générée par Dialogflow.

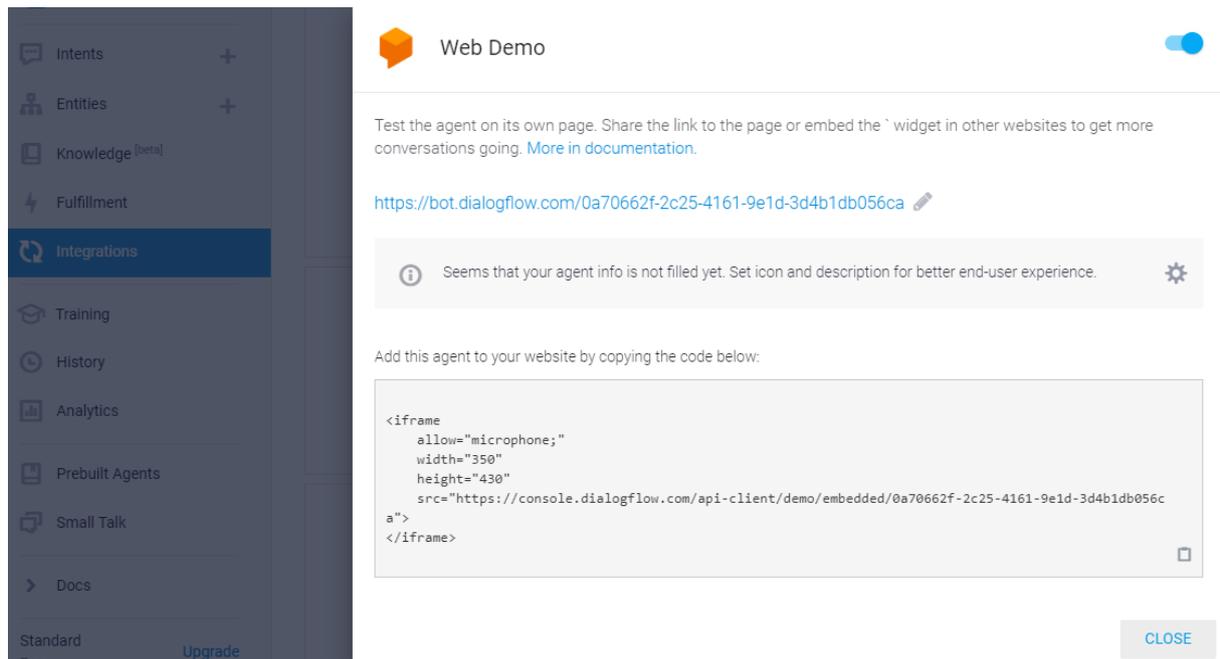


Figure 15 Web demo integration

Cette solution possède quelques limitations. Il n'y a pas la possibilité d'envoyer des liens cliquables, ni images, ni boutons sur lesquels l'utilisateur pourrait cliquer. (Mining Business Data, 2019)

Il n'est pas non plus possible de personnaliser l'aspect visuel de l'interface de messagerie car les fichiers css de l'*iframe* sont propriétés de Google.

De plus, la gestion de session n'est malheureusement pas possible. En effet l'appel à notre webservice se fait sur le site de Dialogflow via une url, dans laquelle les accès (username, password ou token) sont passés en paramètres.

Dès lors, un seul utilisateur dont on aura fourni le nom d'utilisateur, mot de passe ou token, pourra accéder au web service.

Les accès pourraient être dynamiques si un client est développé en PHP avec la librairie proposée. Ci-dessous, une représentation de l'architecture telle qu'elle l'aurait été avec le développement d'un client.

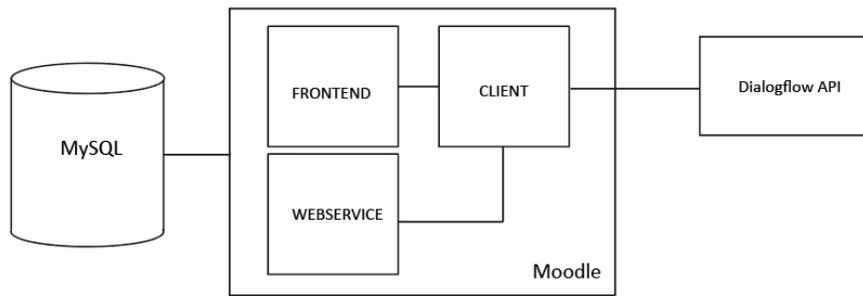


Figure 16 Architecture avec client PHP

7. Design

7.1. Eléments communs aux deux cas d'applications

Pour notre prototype, nous avons imaginé un flux de conversation comme présenté ci-dessous. Il faut savoir qu'il est possible d'imaginer autant de flux que désiré. En l'occurrence, nous l'avons imaginé de telle sorte à ce qu'il s'adapte au mieux pour les deux cas. Nous le détaillerons pour chacun d'eux.

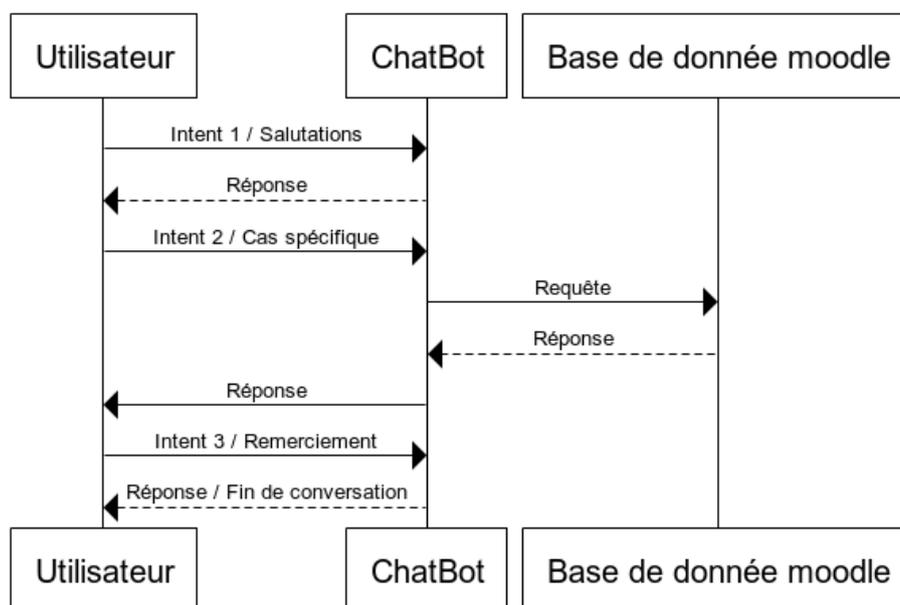


Figure 17 Diagramme de séquence du flux conversationnel

Les intentions un et trois sont les mêmes pour les deux cas et peuvent être facultatives. Pour chacune d'entre elles sont définies des entrées utilisateurs possibles pour les

salutations et pour les remerciements. Nous avons simplement paramétré la réponse du bot en « Hello <Prénom> comment puis-je t'aider ? » pour les salutations et « Mais ce n'est rien » pour les remerciements. L'intention deux, quant à elle, change en fonction du cas.

Bien que ce ne soit pas l'aspect sur lequel nous nous sommes le plus attardé, nous avons quand même voulu intégrer un minimum d'antropomorphisme à notre bot. Nous lui avons donné un nom : Moudy. Il est gentil, encourageant et a un brin d'humour. Pour assurer un certain rapprochement avec l'utilisateur, nous avons privilégié le tutoiement. Nous avons également pris soin de lui donner un aspect sympathique.



Figure 18 Moudy, son apparence

A la connexion, l'utilisateur verra Moudy changer de couleur sur le tableau de bord, comme s'il s'activait. L'œil de l'utilisateur sera automatiquement attiré et la conversation pourra plus facilement s'engager. Nous avons réalisé cette animation grâce à du jquery.

```
<script type="text/javascript">
  $(document).ready(function() {
    $("#moudy").css("background-color", "#3098DF");
    $("#text").css("color", "#000000");
    $("#moudy").click(function() {
      $("#iframe").toggle();
      $("#robot").hide();
    });
  });
</script>
```

Figure 19 Méthode JQuery

7.2. Cas d'utilisation 1 – Clé d'inscription

7.2.1. Diagramme d'activités

Ci-dessous, le flux conversationnel détaillé.

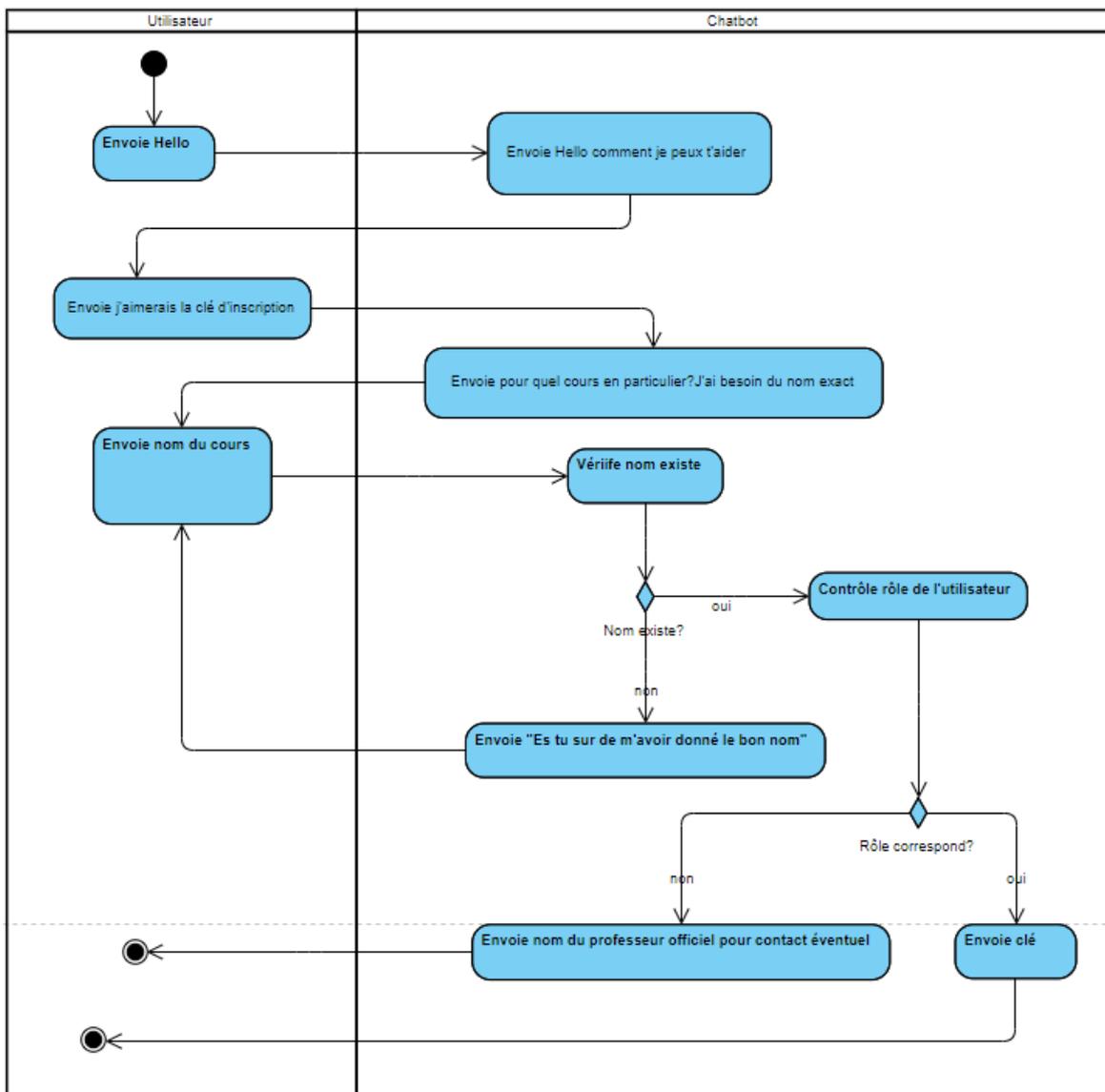


Figure 20 Diagramme d'activité demande de clé d'inscription

Une variante est ajoutée si l'utilisateur possède uniquement l'ID du cours.

7.2.2. Création des intentions

L'intention créée est nommée demande_clef_d'inscription. Nous imaginons ici quelles seraient les entrées possibles de l'utilisateur pour demander la clé.

” Donne-moi la clé d'inscription
” clé d'inscription
” clé
” J'aimerais la clé d'inscription pour un cours

Figure 21 Entrées pour l'intention demande_clef_d'inscription

Pour cette intention, la réponse est donnée directement sur l'interface.

Responses ⓘ

DEFAULT GOOGLE ASSISTANT +

Text response ⓘ

- 1 Okay pour quel cours en particulier? J'ai besoin du nom exact du cours ou de l'ID.
- 2 Enter a text response variant

Figure 22 Réponse intention 1

Grâce au moteur NLP, nous constatons qu'une entrée utilisateur proche de celles définies dans l'intention donnera la même réponse.

USER SAYS COPY CURL

J'ai besoin de la clé d'inscription

🔴 DEFAULT RESPONSE ▾

Okay, pour quel cours en particulier? J'ai besoin du nom exact du cours ou de l'ID

Figure 23 Exemple efficacité moteur NLP

A ce stade, l'utilisateur enverra un nom ou un ID. Il est important de donner un contexte pour savoir que la nouvelle entrée corresponde à l'intention qui précède. Nous ajoutons donc un contexte de sortie à notre intention de base.

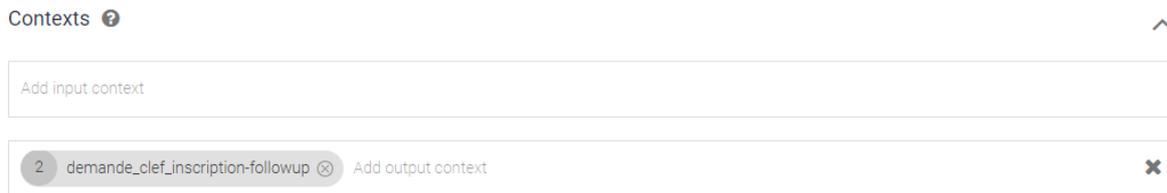


Figure 24 Cas 1 - Contextes

Nous créons deux intentions qui suivent l'intention demande_clef_inscription. Elles auront comme contexte d'entrée demande_clef_inscription-followup.

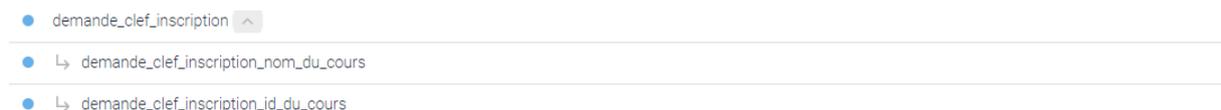


Figure 25 Cas 1 - Intentions suivantes

Pour ces deux intentions, nous activons l'option fulfillment pour assurer la connexion à notre service web et renvoyer une réponse en fonction de l'entrée, après les divers contrôles effectués en base de données.

7.2.3. Création d'entités

Des paramètres sont passés du chatbot à notre service web et doivent correspondre à des entités. Nous avons créé l'entité Nom_de_cours et id_de_cours. L'idée ici est que lorsque l'utilisateur entrera soit un nom, soit un chiffre, l'entité correspondante sera attribuée.

Nom_de_cours



Figure 26 Entité nom de cours

Training phrases ?

Search training

” Add user expression		
” 4		
” 6		
” 2		
PARAMETER NAME	ENTITY	RESOLVED VALUE
Id_du_cours	@Id_du_cours	2

Figure 27 Attribution de l'entité Id_du_cours en fonction de l'entrée utilisateur

7.3. Cas d'utilisation 2 – Quiz pour étudiant

7.3.1. Diagramme d'activité

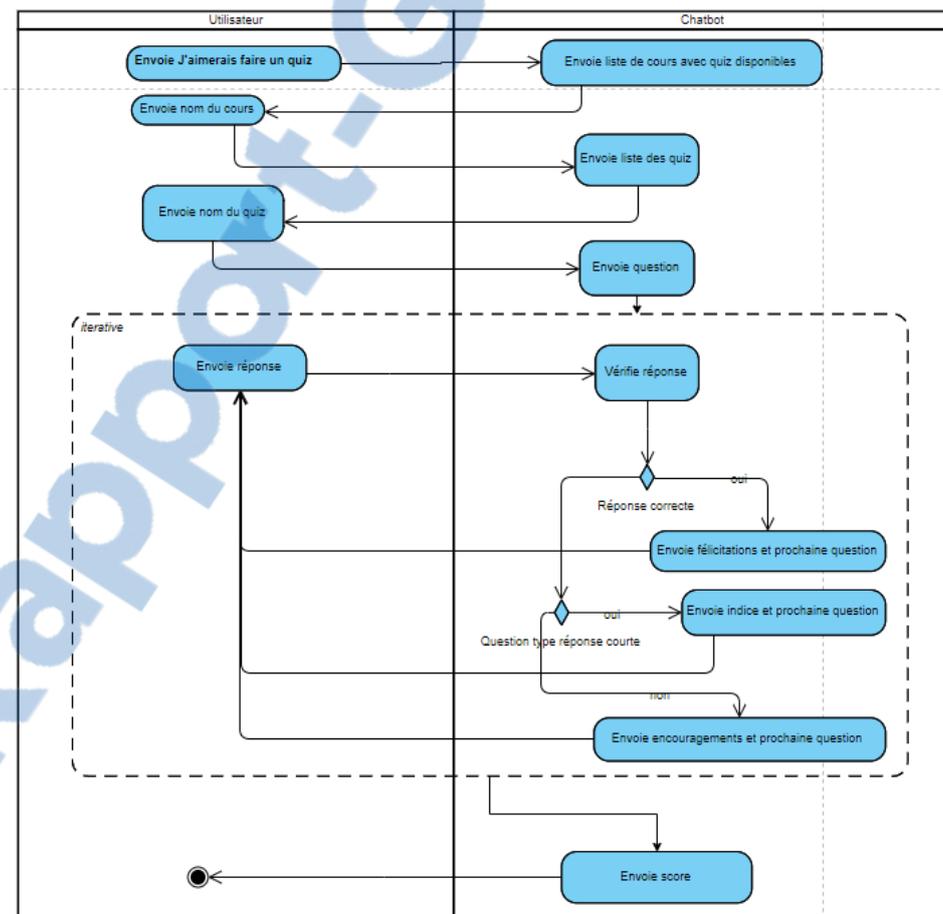


Figure 28 Diagramme d'activité cas d'application quiz

7.3.2. Création des intentions

Pour ce cas nous créons six intentions :

- Faire un quiz,
- Faire un quiz_nom_du_cours,
- Faire_un_quiz_nom_du_cours_nom_du_quiz,
- Quiz_reponse1
- Quiz_reponse2
- Quiz_reponse3

Pour notre prototype, nous avons uniquement trois questions par quiz, d'où les trois intentions « réponse ». Chaque intention possède le contexte de l'intention qui précède pour le suivi de la conversation.

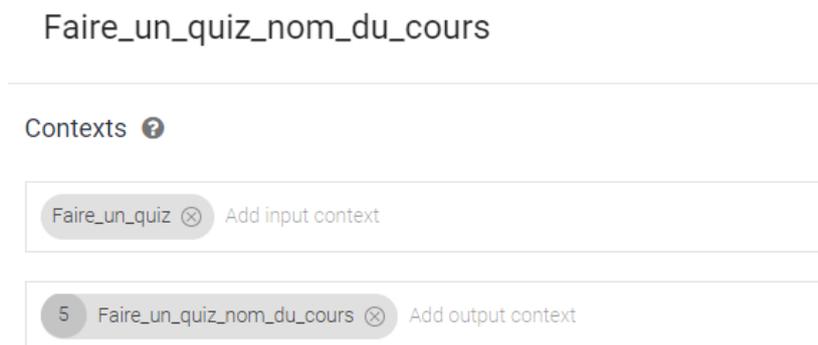


Figure 29 Deuxième intention, son contexte d'entrée et de sortie

L'intention Quiz_reponse3 possède le contexte reponse1 et reponse2. Ces deux contextes sont nécessaires pour calculer le score. Ils permettent de récupérer la première réponse et la deuxième réponse du quiz grâce à la notation #context, et de disposer de ces valeurs dans les paramètres de la requête.

<input type="checkbox"/>	reponse2_quizdroit	@sys.any	#reponse2.Reponse_quiz_droit2	<input type="checkbox"/>
<input type="checkbox"/>	reponse1_quizdroit	@sys.any	#reponse1.Reponse_quiz_droit1	<input type="checkbox"/>

Figure 30 Récupération du paramètre via les contextes

7.3.3. Création d'entités

Nous avons créé trois quiz. Nous avons donc trois entités créées qui contiennent les réponses des quiz.

@ Reponse_quiz_disque_dur

@ Reponse_quiz_droit

@ Reponse_utilisateurs_windows

Figure 31 Cas 2 - Entités

Reponse_quiz_disque_dur

Define synonyms ⓘ Allow automated expansion

Cylindre	Cylindre
Je ne sais pas	je sais pas, no idea, segments, trop dur
Secteurs	Secteur, Secteurs, secteur, secteurs
Vrai	C'est juste, Oui, Vrai

Figure 32 Entité correspondant aux réponses du quiz disque dur

La réponse donnée par l'utilisateur pour la première question par exemple (Intention Quiz_reponse1) correspondra à l'entité.

Quiz_reponse1

SAVE

REQUIRED ⓘ	PARAMETER NAME ⓘ	ENTITY ⓘ	VALUE	IS LIST ⓘ
<input type="checkbox"/>	Reponse_quiz_droit1	@Reponse_quiz_droit	\$Reponse_quiz_droit1	<input type="checkbox"/>
<input type="checkbox"/>	Reponse_quiz_disque_dur1	@Reponse_quiz_disque_dur	\$Reponse_quiz_disque_dur1	<input type="checkbox"/>
<input type="checkbox"/>	Reponse_utilisateurs_windows1	@Reponse_utilisateurs_windows	\$Reponse_utilisateurs_windows1	<input type="checkbox"/>
<input type="checkbox"/>	Nom_de_quizz	@Nom_de_quizz	#Faire_un_quiz_nom_du_cours_nom_du_quiz.Nom_de_quizz	<input type="checkbox"/>

Figure 33 Paramètres de l'intention Quiz_reponse1

Le paramètre Nom_de_quizz est indispensable car il permet de récupérer les réponses qui correspondent au quiz. La méthode sur l'image suivante présente ce processus.

```

public static function getResponseFromParameter($queryResultDecoded, $quiz_name)
{
    $action = $queryResultDecoded->action;
    $responsel = "";

    if ($quiz_name == 'Formes juridiques') {
        switch ($action) {
            case "faire_un_quiz.nom_du_cours.nom_du_quiz.r1":
                $responsel = $queryResultDecoded->parameters->Reponse_quiz_droit1;
                break;
            case "faire_un_quiz.nom_du_cours.nom_du_quiz.r2":
                $responsel = $queryResultDecoded->parameters->Reponse_quiz_droit2;
                break;
            case "faire_un_quiz.nom_du_cours.nom_du_quiz.r3":
                $responsel = $queryResultDecoded->parameters->Reponse_quiz_droit3;
                break;
        }
    } elseif ($quiz_name == 'Le disque dur') {
        switch ($action) {
            case "faire_un_quiz.nom_du_cours.nom_du_quiz.r1":
                $responsel = $queryResultDecoded->parameters->Reponse_quiz_disque_durl;
                break;
            case "faire_un_quiz.nom_du_cours.nom_du_quiz.r2":
                $responsel = $queryResultDecoded->parameters->Reponse_quiz_disque_dur2;
                break;
            case "faire_un_quiz.nom_du_cours.nom_du_quiz.r3":
                $responsel = $queryResultDecoded->parameters->Reponse_quiz_disque_dur3;
                break;
        }
    }
}

```

Figure 34 Méthode retournant le paramètre réponse exact en fonction du nom du quiz

8. Développement

8.1. Architecture

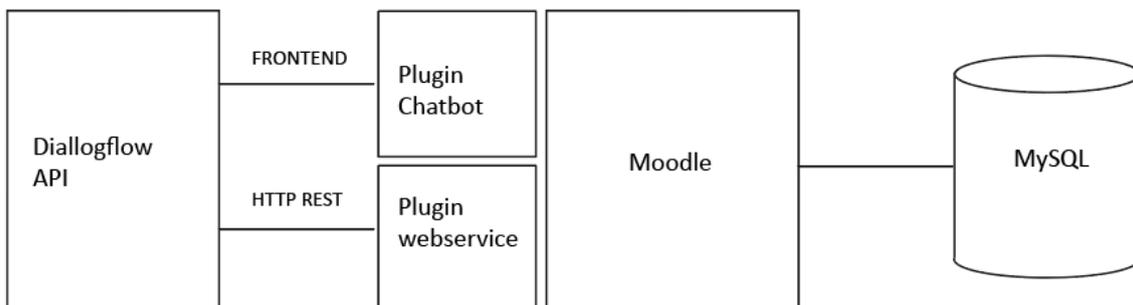


Figure 35 Architecture du prototype

Pour ce prototype, nous avons du développer deux plugins.

- Un plugin nommé Chatbot, pour le frontend.
- Un plugin service web nommé wschatbot pour permettre l'échange de données de la base de donnée avec notre chatbot Dialogflow.

Nous les détaillons au chapitre 8.3.

8.2. Environnement de développement

8.2.1. Installation de Moodle

Pour nos divers tests, il a été nécessaire d'installer Moodle en local. Comme le centre Cyberlearn envisage la migration de leur plateforme vers la version 3.7, c'est cette dernière version que nous avons téléchargée pour s'assurer que le prototype du chatbot développé pourrait fonctionner par la suite. Le téléchargement de la dernière version stable s'est fait sur le site officiel de Moodle.

Version	Information	.tgz	.zip
Moodle 3.7+ MOODLE_37_STABLE	This package is built every week with new fixes produced by our stable development process . It contains a number of fixes made since the 3.7 release and is usually a better choice for production than the actual 3.7 package below.	Download tgz	Download zip
Built Weekly 1 day 13 hours ago	<ul style="list-style-type: none"> • Recent changes log • Upgrading notes • Requires: PHP 7.1, MariaDB 5.5.31 or MySQL 5.6 or Postgres 9.4 or MSSQL 2008 or Oracle 11.2 • Language packs 	43.6MB 278 today	58MB 1769 today
		[md5] [sha256]	[md5] [sha256]

Figure 36 Site officiel et page de téléchargement Moodle

8.2.2. Architecture des dossiers

Lorsque le dossier .zip est décompressé nous obtenons les fichiers suivants :

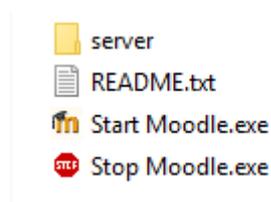


Figure 37 Dossier décompressé Moodle

Moodle dispose d'un serveur Apache comprenant MySQL et PHP. Au premier lancement de l'exécutable Start Moodle.exe, le serveur Apache est lancé une phase d'installation est

accessible sur localhost :80. Durant cette phase d'installation, diverses configurations sont faites dont le choix de la langue et les chemins d'accès. C'est ici également que la base de données sera créée et sera accessible sur le port 3306. Nous y accéderons régulièrement pour formuler nos requêtes SQL avec MySQL Workbench, programme que nous avons déjà sur notre ordinateur.

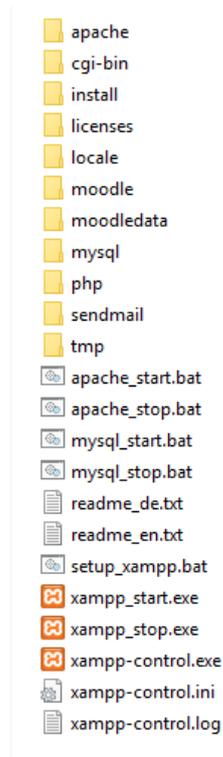


Figure 38 Architecture de dossier moodle

Le dossier « moodle » est le cœur de l'application. C'est ce dernier que nous allons modifier.

8.2.3. Git

Pour assurer le versioning et une sauvegarde de notre développement, nous avons créé un repository Git sur GitHub.

8.2.4. PHPStorm

Comme Moodle est développé en PHP, nous avons sélectionné l'IDE PHPStorm version 2018.3 qui offre un environnement complet et adapté pour le développement. La racine de notre projet est le dossier « moodle » qui comprend à la base une soixantaine de dossiers et divers fichiers de configuration à la racine. Nous allons décrire uniquement ceux dont nous

aurons utilité. Pour plus d'informations, il est possible de se référer à la documentation moodle https://docs.moodle.org/dev/Moodle_architecture.

8.2.5. Ngrok

Le service web auquel Dialogflow doit envoyer ses requêtes doit d'être accessible depuis une adresse publique. Dès lors, l'adresse `http://localhost:80` de notre Moodle avec son plugin Service web doit changer. Ngrok est l'outil qui permet de publier un site en local sur internet. Nous l'avons téléchargé sur le site officiel. <https://ngrok.com/>

A partir du dossier source, nous avons lancé le terminal et tapé la commandé `ngrok http 80`. Une nouvelle adresse publique nous est donnée.

```
ngrok by @inconshreveable
Session Status      online
Account             celahm (Plan: Free)
Version             2.3.30
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://dc544cf0.ngrok.io -> http://localhost:80
Forwarding           https://dc544cf0.ngrok.io -> http://localhost:80

Connections
  ttl    opn    rt1    rt5    p50    p90
  263    0      0.00   0.00   5.49   5.97
```

Figure 39 Statut de ngrok et adresses

C'est l'adresse sécurisée `https` que nous allons fournir à Dialogflow. Ceci est présenté sur la figure suivante.

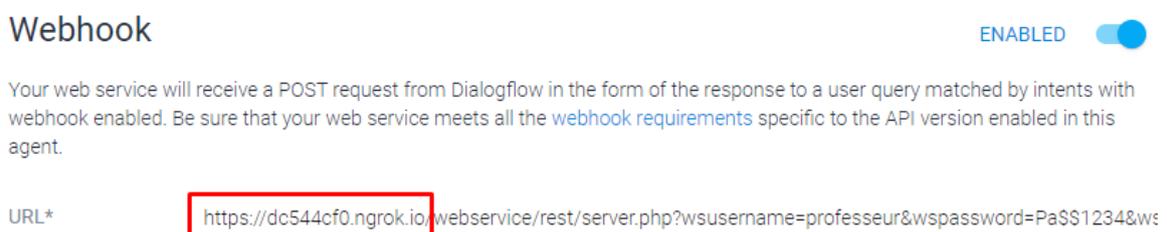


Figure 40 URL de notre service web

Pour éviter une redirection vers `http://localhost` lors d'une requête, il est important de changer également la variable `$CFG->wwwroot` avec l'adresse fournie par Ngrok.

```
1 <?php // Moodle configuration file
2
3 unset($CFG);
4 global $CFG;
5 $CFG = new stdClass();
6
7 $CFG->dbtype = 'mariadb';
8 $CFG->dblibrary = 'native';
9 $CFG->dbhost = 'localhost';
10 $CFG->dbname = 'moodle';
11 $CFG->dbuser = 'root';
12 $CFG->dbpass = '';
13 $CFG->prefix = 'mdl_';
14 $CFG->dboptions = array (
15     'dbpersist' => 0,
16     'dbport' => '',
17     'dbsocket' => '',
18     'dbcollation' => 'utf8mb4_unicode_ci',
19 );
20 // $CFG->wwwroot = 'http://localhost';
21 $CFG->wwwroot = 'http://dc544cf0.ngrok.io';
22 $CFG->dataroot = 'C:\\TB_Moodle\\MoodleWindowsInstaller-latest-37\\server\\moodledata';
23 $CFG->admin = 'admin';
24
25 $CFG->directorypermissions = 0777;
26
27 require_once(__DIR__ . '/lib/setup.php');
28
29 // There is no php closing tag in this file,
30 // it is intentional because it prevents trailing whitespace problems!
31
```

Figure 41 Fichier config.PHP

8.3. Développement de plugin Moodle

8.3.1. Bloc HTML

L'interface utilisateur pour communiquer avec notre bot doit se trouver sur le tableau de bord de chacun des utilisateurs. Pour ce faire, nous avons choisi de développer un plugin de type bloc car il peut être placé à n'importe quel endroit sur notre page et permet l'affichage de n'importe quelles informations.

Il existe également plusieurs blocs préformatés comme les blocs « dernières news », « utilisateurs en ligne », ou « activité récente ». Pour notre cas nous avons sélectionné le bloc HTML. Ce dernier est flexible, et peut justement intégrer plusieurs fonctions.

Dans notre architecture de projet, sous « moodle>blocks », nous avons créé un dossier nommé chatbot comprenant les fichiers requis pour la création du plugin. Nous avons

également créé un fichier style.css pour adapter la balise iframe aux dimensions de bases du bloc.

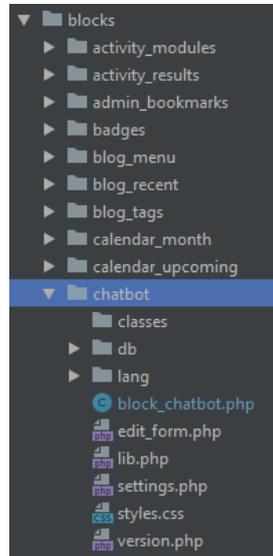


Figure 42 Plugin chatbot

Dans le fichier block_chatbot.PHP, la méthode get_content() génère le contenu à l'intérieur du bloc. Nous y avons inséré la balise iframe proposée par Dialogflow.

```
90     return $this->content;
91     */
92     public function get_content()
93     {
94         if ($this->content !== null) {
95             return $this->content;
96         }
97
98         $this->content = new stdClass;
99         $this->content->text = '
100
101         <iframe
102             allow=""
103             width="100%"
104             height="500"
105             src="https://console.dialogflow.com/api-client/demo/embedded/0a70662f-2c25-4161-9e1d-3d4b1db056ca">
106         </iframe>
107
108         '
109
110         return $this->content;
111     }
```

Figure 43 fichier block_chatbot.PHP

En tant qu'administrateur Moodle, nous pouvons ajouter le bloc sur le tableau de bord après avoir cliqué sur Modifier cette page

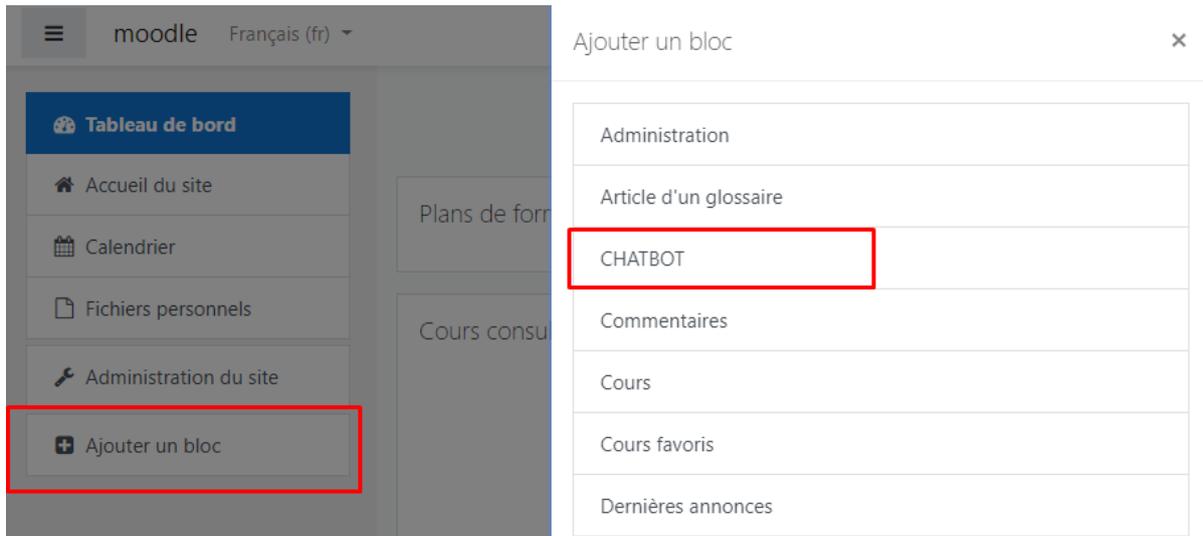


Figure 44 Ajout d'un bloc sur le tableau de bord

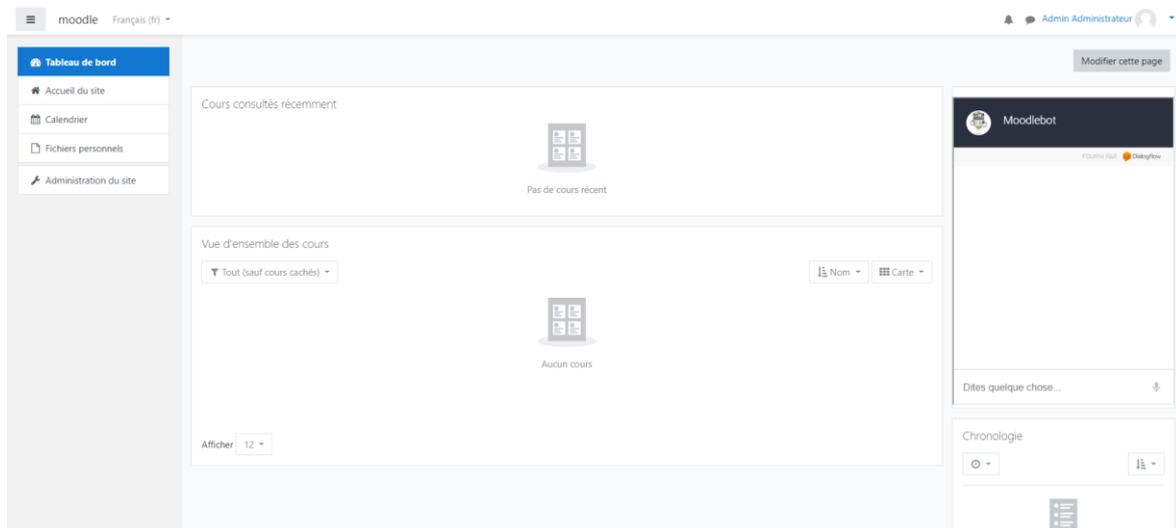


Figure 45 Tableau de bord administrateur

8.3.2. Service web

Le service web est indispensable pour récupérer les requêtes des utilisateurs, les gérer, et renvoyer des réponses. Ce développement de plugin est un peu plus compliqué car il nécessite plusieurs étapes.

Pour le concevoir, nous sommes partis d'un plugin d'exemple sur le site officiel https://moodle.org/plugins/local_wstemplate que nous avons adapté. L'installation de ce plugin s'est faite via l'interface administrateur sous « Administration du site>Plugin>Installer

des plugins ». L'emplacement du service web dans notre architecture de dossier se trouve dans « moodle>local>wschatbot ».

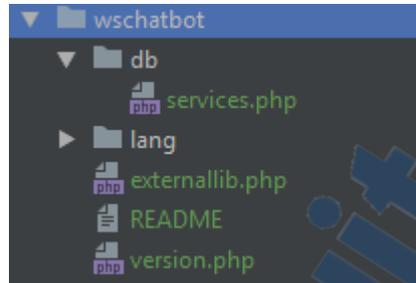


Figure 46 Plugin service web chatbot

Nous avons adapté le nom du service web et le nom des fonctions/méthodes pour qu'ils correspondent à notre projet.

```
<?php
//...
/**
 * Web service local plugin template external functions and service definitions.
 *
 */
// We defined the web service functions to install.
$functions = array(
    'local_wschatbot_send_response' => array(//web service function name
        'classname' => 'local_wschatbot_external',//class containing the external function
        'methodname' => 'send_response',//external function name
        'classpath' => 'local_wschatbot/externallib.php',//file containing the class/external function
        'description' => 'Return chatbot response',//human readable description of the web service function
        'type' => 'update',//database rights of the web service function (read, write)
    )
);
// We define the services to install as pre-build services. A pre-build service is not editable by administrator.
$services = array(
    'Chatbot service' => array( //the name of the web service
        'functions' => array ('local_wschatbot_send_response',//web service functions of this service
        'restrictedusers' => 0,//if enabled, the Moodle administrator must link some user to this service into the administration
        'enabled'=>1,// if enabled, the service can be reachable on a default installation
    )
)
```

Figure 47 fichier db/services.PHP

Le fichier services.PHP permet l'enregistrement du service dans la base de données avec les fonctions qui lui sont propres. Notre service web possède une unique fonction appelée local_wschatbot_send_response.

A l'intérieur de cette méthode qui se trouve dans le fichier external.lib, nous avons récupéré la requête POST envoyée par Dialogflow à notre service web.

```
// get the content from POST, decode the result in json
$content = trim(file_get_contents( filename: 'php://input'));
$queryResultDecoded = json_decode($content)->queryResult;
```

Figure 48 fichier externalib.PHP

Voici un exemple d'une requête POST envoyée par le client

```
POST /webservice/rest/server.php?wsusername=fseppey&wspassword=Pa$$1234&ws
function=local_wschatbot_send_response&moodlewsrestformat=json HTTP/1.1
content-type: application/json
Host: 1742c349.ngrok.io
Content-Length: 776
Accept: application/json
User-Agent: Google-Dialogflow
Accept-Encoding: gzip,deflate,br
X-Forwarded-Proto: https
X-Forwarded-For: 64.233.172.140

{
  "responseId": "f19b6165-aed3-4afc-99c1-b9333d627fba-5b26cf67",
  "queryResult": {
    "queryText": "Hello",
    "action": "input.welcome",
    "parameters": {
    },
    "allRequiredParamsPresent": true,
    "fulfillmentText": "Hello comment je peux t\u0027aider?",
    "fulfillmentMessages": [{
      "text": {
        "text": ["Hello comment je peux t\u0027aider?"]
      }
    }
  ],
  "intent": {
    "name": "projects/moodlebot-xtjqpp/agent/intents/54a44683-be43-439c-
836a-1056f2a9f0d8",
    "displayName": "Default Welcome Intent"
  },
  "intentDetectionConfidence": 1.0,
  "languageCode": "fr"
},
"originalDetectIntentRequest": {
  "payload": {
  }
},
"session": "projects/moodlebot-xtjqpp/agent/sessions/f11b25a2-0acb-61de-
6164-030e2643a573"
}
```

Figure 49 Requête client

```
{
  "fulfillmentText": " Hello Florence! Comment je peux t'aider ?",
  "source": "webhook"
}
```

Figure 50 Réponse du service web

8.3.3. Configuration administrateur

Pour utiliser le service web créé, les étapes requises sont les suivantes :

- activer la fonctionnalité services web sous « Administration du site > Fonctions avancées ».

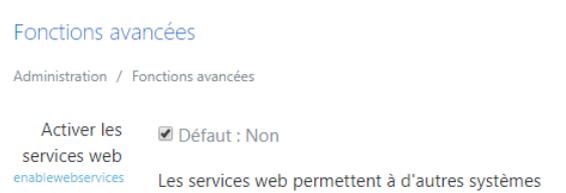


Figure 51 Activation des services web

- activer le protocole utilisé par le service web « Administration du site>Plugins>Services web> Gérer les protocoles »

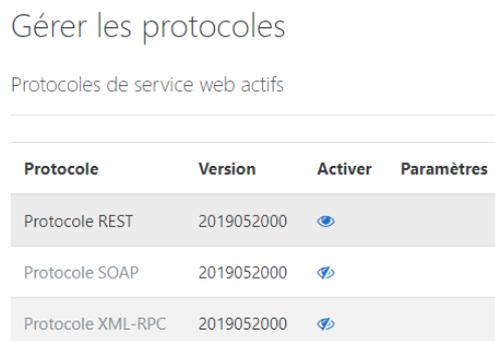


Figure 52 Gestion des protocoles sur Moodle

- créer un rôle système ayant la possibilité d'utiliser le service web et l'attribuer aux utilisateurs souhaités

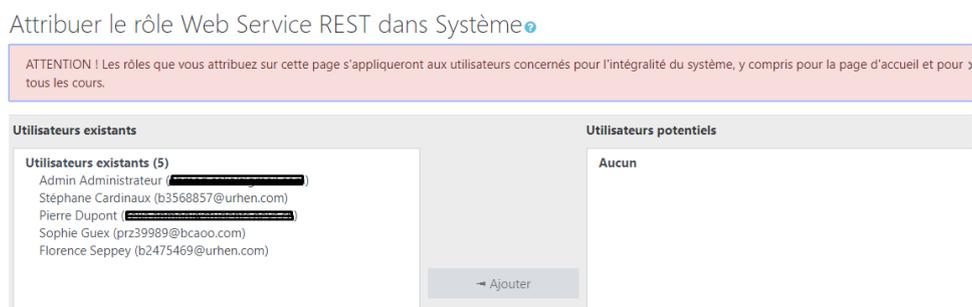


Figure 53 Attribution du rôle web service REST

8.4. Business logic du chatbot

Toutes les réponses du bot sont gérées dans la méthode `send_response()` du fichier `external.lib`. Cette méthode fonctionne notamment grâce à un `switch case` sur l'action envoyée par le client.

```

$action = $queryResultDecoded->action;

// return a response depending on the received action
switch ($action) {
    case "input.welcome":
        $response->fulfillmentText = " Hello " . $user_name . " ! Comment je peux t'aider ? Dis-moi ce que tu veux faire. " ;
        break;
    case "demande_clef_inscription.demande_clef_inscription_nom_du_cours":
        if (isset($course_name)) {
            $id_course = self::getCourseID($course_name);
            if (empty($id_course)) {
                $response->fulfillmentText = "Es-tu sûr de m'avoir donné le bon nom du cours? Ce n'est pas un cours valide." ;
            } else $response->fulfillmentText = self::getPassword($id_course, $user_id);
        }
        break;
    case "demande_clef_inscription.demande_clef_inscription_id_du_cours":
        if (isset($course_id)) {
            if (self::checkifCourseExist($course_id)) {
                $response->fulfillmentText = self::getPassword($course_id, $user_id);
            } else $response->fulfillmentText = "Es-tu sûr de m'avoir donné le bon id de cours? Ce n'est pas un cours valide." ;
        }
        break;
}

```

Figure 54 Switch case sur le paramètre action de la requête

Nous avons créé seize méthodes qui, pour la plupart, accèdent aux informations de la base de données. Elles sont présentées sur la figure 55.

```

public static function getCourseID($course_name) [...]
public static function getPassword($course_id, $user_id) [...]
public static function checkifCourseExist($course_id) [...]
public static function getEditingTeacherName($course_id, $role) [...]
public static function getQuizzAvailable($user_id) [...]
public static function getAllCourse($user_id) [...]
public static function getAllQuizz($course_name) [...]
public static function getQuestionOfQuizz($quizzname) [...]
public static function getResponsesOfQuizz($questiontext) [...]
public static function getRightAnswer($questiontext) [...]
public static function isTeacher($user_id) [...]
public static function getAllInfoForQuestion($quiz_name, $question_number) [...]
public static function toString($question_number, $question, $responses, $type) [...]
public static function getResponseFromParameter($queryResultDecoded, $quiz_name) [...]
public static function getPreviousResponse($quizzname, $queryResultDecoded) [...]
public static function getFinalScore($response1, $response2, $rightAnswer1, $rightAnswer2) [...]

```

Figure 55 Méthodes créées

```

public static function getAllInfoForQuestion($quiz_name, $question_number)
{
    $question_number = $question_number - 1;
    //create array to send back information
    $allInfo = new stdClass();

    $quizQuestions = array_values(self::getQuestionOfQuiz($quiz_name));
    $allInfo->nbOfQuestion = count($quizQuestions);
    $allInfo->question = $quizQuestions[$question_number]->questiontext;
    $allInfo->type = $quizQuestions[$question_number]->qtype;
    $allInfo->responses = self::getResponsesOfQuiz($allInfo->question);
    $allInfo->rightAnswer = self::getRightAnswer($allInfo->question)->answer;
    $allInfo->hint = $quizQuestions[$question_number]->hint;

    return $allInfo;
}
    
```

Figure 56 Détail de la méthode getAllInfoForQuestion()

9. Tests

9.1. Création de données Moodle

Pour nos différents tests, nous avons créé plusieurs objets dans notre base de données.

9.1.1. Utilisateurs

Nous avons créé quatre utilisateurs.

Nom	Prénom	Nom d'utilisateur	Mot de passe
Dupont	Pierre	pdupont	Pa\$\$1234
Guex	Sophie	sguex	Pa\$\$1234
Cardinaux	Stéphane	scardinaux	Pa\$\$1234
Seppey	Florence	fseppey	Pa\$\$1234

Tableau 8 Données Moodle - Utilisateurs

9.1.2. Cours

Nous avons créé trois cours. MOI correspond à maîtrise d'un ordinateur individuel.

Id	Nom	Clé d'inscription
2	Python	python2019
3	MOI	moi2019
5	Droit	droit2019

Tableau 9 Données Moodle - Cours

9.1.3. Quiz

Nous avons créé trois quiz de trois questions chacun.

Nom	Cours
Le disque dur	MOI
Utilisateurs sous windows	MOI
Formes juridiques	Droit

Tableau 10 Données Moodle – Quiz

9.1.4. Attribution des rôles et accès aux quiz

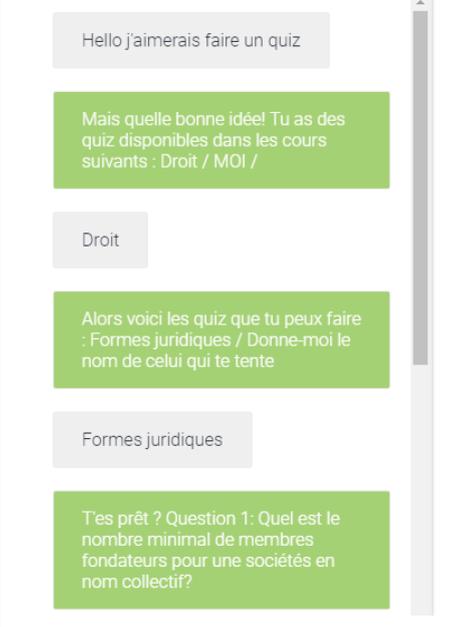
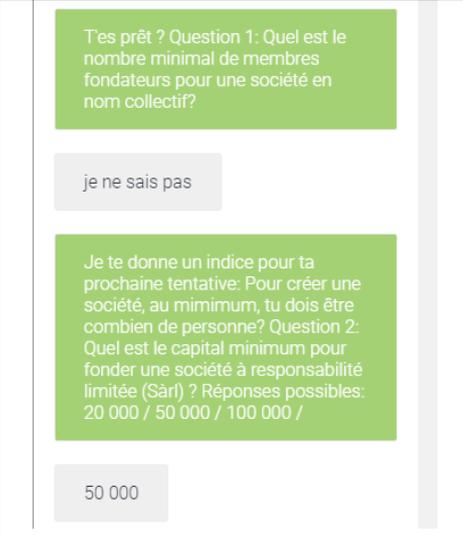
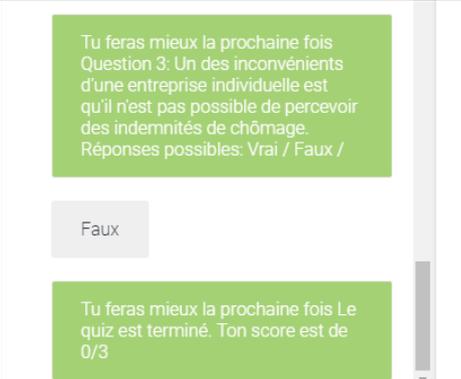
Utilisateurs	Cours	Rôles	Quiz disponibles
Dupont Pierre	Droit, Python, MOI	Etudiant	3
Guex Sophie	Droit	Etudiante	1
Cardinaux Stéphane	Droit, MOI	Professeur	3

Seppey Florence	Python	Professeur	0
-----------------	--------	------------	---

Tableau 11 Données Moodle - Attribution des rôles et accès aux quiz

9.2. Simulations

Cas d'utilisation		
Clé d'inscription	<p>Hello</p> <p>Hello Florence! Comment puis-je t'aider ?</p> <p>J'aimerais une clé d'inscription</p> <p>Okay, pour quel cours en particulier? J'ai besoin du nom exact du cours ou de l'ID</p> <p>Pour le cours python stp</p> <p>Tu es bien le professeur officiel du cours. Voici la clé d'inscription : "python2019"</p> <p>Figure 57 Florence Seppey</p>	<p>J'aimerais une clé</p> <p>Okay, pour quel cours en particulier? J'ai besoin du nom exact du cours ou de l'ID</p> <p>MOI</p> <p>Désolé, je ne peux pas te donner la clé car tu n'es pas le professeur officiel, il faut que tu contactes le professeur Cardinaux Stéphane</p> <p>Figure 58 Florence Seppey</p>
Quiz	<p>Hello j'aimerais faire un quiz</p> <p>Désolé, tu n'as pas de quiz disponibles</p> <p>Figure 59 Sophie Guex</p>	

<p>Quiz</p>	 <p>Hello j'aimerais faire un quiz</p> <p>Mais quelle bonne idée! Tu as des quiz disponibles dans les cours suivants : Droit / MOI /</p> <p>Droit</p> <p>Alors voici les quiz que tu peux faire : Formes juridiques / Donne-moi le nom de celui qui te tente</p> <p>Formes juridiques</p> <p>Tes prêt ? Question 1: Quel est le nombre minimal de membres fondateurs pour une sociétés en nom collectif?</p>	 <p>2</p> <p>Réponse correcte! Question 2: Quel est le capital minimum pour fonder une société à responsabilité limitée (Sàrl) ? Réponses possibles: 20 000 / 50 000 / 100 000 /</p> <p>20 000</p> <p>Bravo bonne réponse! Question 3: Un des inconvénients d'une entreprise individuelle est qu'il n'est pas possible de percevoir des indemnités de chômage. Réponses possibles: Vrai / Faux /</p> <p>Vrai</p> <p>Bravo bonne réponse !Le quiz est terminé. Ton score est de 3/3</p>
<p>Quiz</p>	 <p>Tes prêt ? Question 1: Quel est le nombre minimal de membres fondateurs pour une société en nom collectif?</p> <p>je ne sais pas</p> <p>Je te donne un indice pour ta prochaine tentative: Pour créer une société, au mimimum, tu dois être combien de personne? Question 2: Quel est le capital minimum pour fonder une société à responsabilité limitée (Sàrl) ? Réponses possibles: 20 000 / 50 000 / 100 000 /</p> <p>50 000</p>	 <p>Tu feras mieux la prochaine fois Question 3: Un des inconvénients d'une entreprise individuelle est qu'il n'est pas possible de percevoir des indemnités de chômage. Réponses possibles: Vrai / Faux /</p> <p>Faux</p> <p>Tu feras mieux la prochaine fois Le quiz est terminé. Ton score est de 0/3</p>

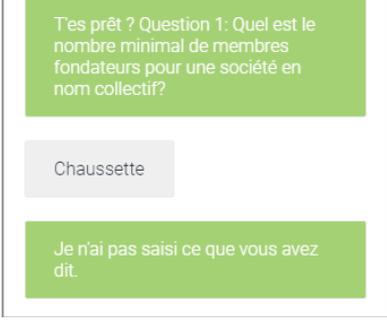
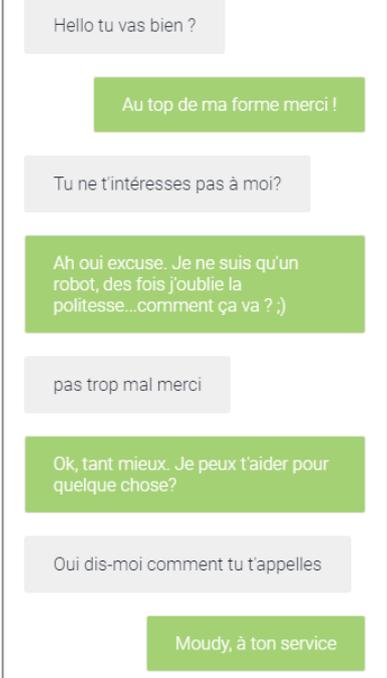
<p>Quiz</p>	 <p>Figure 64 Cas non géré type de réponse courte</p>	<p>L'utilisateur peut entrer n'importe quel mot qui n'a pas été pensé lors de la définition de l'entité. Le chatbot ne l'identifie donc pas et ne peut pas donner suite.</p>
<p>Conversation basique</p>	 <p>Figure 65 Autres échanges</p>	

Tableau 12 Simulations de conversations

10. Méthodologie de gestion de projet

Nous avons sélectionné la méthodologie agile SCRUM pour la réalisation de ce travail. Elle nous a été enseignée durant le cinquième semestre de notre cursus et nous avons pu découvrir les aspects positifs de cette méthode durant la réalisation de différents travaux de groupe. Evidemment, la méthodologie SCRUM s'applique dans un contexte particulier, avec des acteurs précis, et permet la réalisation d'un livrable fonctionnel à des intervalles réguliers, ce qui diffère en quelques points de notre travail de Bachelor. Elle a l'avantage cependant de visualiser avec précision l'avancement d'un projet. De plus, pour notre cas, il était difficile d'anticiper et de planifier clairement le développement technique du projet. C'est là un des critères fondateurs de la méthodologie SCRUM : la possibilité de changer les fonctionnalités requises au fur et à mesure.

Ci-dessous sont présentés en détail les points qui diffèrent entre notre application de la méthodologie et celle appliquée en général.

10.1. Général / Product backlog

Cette méthodologie s'applique aux projets de développement logiciel, or dans notre cas, il y a une partie développement certes, mais qui est moindre. La partie de recherche et de rédaction représente la majeure partie de ce travail.

Avec SCRUM, les user stories définies et qui apparaissent dans le *product backlog* sont uniquement des fonctionnalités qui devront être implémentées. Même si la partie analyse/recherche est essentielle du côté du développeur elle ne doit en aucun cas figurer dans le *product backlog*. C'est le premier point que nous avons dû adapter. Dans notre *product backlog*, la majorité des user stories qui y figurent nous concernent.

10.2. Rôles

Les rôles qui interviennent dans un projet SCRUM sont les suivants et sont en principe attribués à des personnes distinctes (Scrum (développement), 2019) :

- le propriétaire du projet, *product owner* en anglais, qui définit les fonctionnalités et prend les décisions importantes concernant le projet ;
- le SCRUM Master à qui incombe la gestion et l'application de la méthodologie ;

- l'équipe de développement de trois à neuf personnes, qui a pour mission la réalisation technique du produit.

Pour notre projet, ces trois rôles nous sont attribués. Le client reste toutefois la responsable de Cyberlearn qui nous indique ses besoins et préférences lors de nos rendez-vous, qui furent de quatre au total.

10.3. Événements

Les événements principaux que possède SCRUM sont originellement :

- La définition du done
- Le sprint
- Rencontre journalière
- Revue de sprint
- Rétrospective du sprint

Le sprint a été gardé contrairement aux autres événements. Ces derniers n'ont pas pu être appliqués car les rôles ne sont attribués qu'à une seule personne. Cependant, chaque jour ou deux jours de travail, les différents avancements et problèmes rencontrés ont été notés.

10.4. Sprints

Nous avons douze semaines pour la réalisation de ce travail. Nous avons donc logiquement divisé ces semaines en quatre sprints, soit trois semaines par sprint.

10.4.1. Sprint 0

L'objectif de ce premier sprint était la planification pour avoir une vision claire du déroulement, commencer les premières recherches sur le sujet et la rédaction. Nous avons donc pris 24 *story points* avec un temps initial de 58 heures que nous avons réussi à respecter. Le temps effectif de travail pour ce sprint est de 64.5 heures. Les 4.5 heures de différence ont été consacrées à des recherches.

US N°	Thème	User Stories			Acceptance Criteria	Priority	Status	Story Points	Sprint
		En tant que ...	Je veux ...	car/pour ...					
1	Analyse	Réalisatrice du TB	m'informer sur le tb	avoir toutes les infos	Télécharger tous les documents, rdv avec Anne-Dominique	1000	●	2	0
2	Analyse	Réalisatrice du TB	planifier le déroulement du tb	avoir des dates fixes et une idée claire du temps à disposition	Product backlog réalisé	1000	●	2	0
3	Analyse	Réalisatrice du TB	avoir une table des matières en dur	avoir une direction claire et un aperçu des recherches à effectuer	table des matières validée par Anne-Dominique	980	●	16	0
4	Ecriture	Réalisatrice du TB	rédigier le chapitre généralités	avancer de manière visible dans le TB	chapitre rédigé	950	●	20	0

Figure 66 User stories sprint 0

ID US.task	Task Name	Resp.	Initial Estimate
1.1	Télécharger les documents	Célia	1
1.2	Lire les directives	Célia	2
1.3	Rendez vous avec Anne-Dominique	Célia	1
1.4	Préparer les documents	Célia	1
2.1	Préparer les userstories	Célia	4
2.2	Poker planning	Célia	1
2.3	Remplir le fichier product backlog	Célia	1
3.1	Recherches sur chatbot(internet)	Célia	16
3.2	Ecriture de la table des matières	Célia	8
3.3	Rendez vous avec Anne-Dominique	Célia	1
4.1	Création du docword et structure	Célia	2
4.2	Rechercher des sources valides pour le 1er chapitre	Célia	10
4.3	Redaction 1er chapitre	Célia	10
		Total	58

Figure 67 Tâches sprint 0

10.4.2. Sprint 1

L'objectif de ce sprint était la rédaction de la majeure partie du rapport. Nous sommes partis avec 72 *story points* estimés à 67 heures. Nous avons douze heures restantes car nous avons utilisé des heures réservées à la rédaction du chapitre « chatbot et éducation » pour d'autres tâches non planifiées. Les heures effectives sont de 76.5 heures.

US N°	Thème	User Stories			Acceptance Criteria	Priority	Status	Story Points	Sprint	US accepted (done)
		En tant que ...	Je veux ...	car/pour ...						
5	Administratif	Réalisatrice du TB	Ecrire un cahier des charges	pour le présenter à la prochaine séance	validé par anne dominique	970	●	2	1	accepted
6	Ecriture	Réalisatrice du TB	rédigier le chapitre approche technique	avancer de manière visible dans le TB	chapitre rédigé	900	●	30	1	accepted
7	Ecriture	Réalisatrice du TB	rédigier le chapitre problématique	comprendre l'enjeu du tb	chapitre rédigé	870	●	10	1	accepted
8	Ecriture	Réalisatrice du TB	rédigier le chapitre Chat bot et éducation	avancer de manière visible dans le TB	chapitre rédigé	850	●	10	1	accepted
9	Ecriture	Réalisatrice du TB	Rédiger le chapitre Définition de cas d'application	avancer de manière visible dans le TB	chapitre rédigé	820	●	20	1	accepted

Figure 68 User stories sprint 1

ID US.tas	Task Name	Resp.	Initial Estimate
5	Rédiger cahier des charges	Célia	2
6.1	Rechercher des sources valides pour approche technique	Célia	10
6.2	Rédiger le chapitre approche technique	Célia	20
7.1	Rédiger le chapitre problématique	Célia	4
8.1	Rechercher des sources valides	Célia	4
8.2	Rédiger le chapitre chatbot et éducation	Célia	16
9.1	Créer un grille de critères	Célia	2
9.2	Rdv avec support cyberlearn pour analyse des cas	Célia	1
9.3	Comparer et sélectionner les cas	Célia	8
Total			67

Figure 69 Tâches sprint 1

10.4.3. Sprint 2

L'objectif du sprint était la préparation au développement. Nous avons estimé ce sprint à 80 heures que nous avons respectées. Les heures effectives sont de 95 heures, différence due à la rédaction du chapitre chatbot et éducation.

US M	Thème		User Stories	Acceptance Criteria	Priority	Status	Story Points	Sprint	US accepted (done done)
10	Ecriture	Réalisatrice du TB	Rédiger le chapitre Choix du moteur	avancer de manière visible dans le TB	chapitre rédigé	800	30	2	accepted
12	Développement	Réalisatrice du TB	Mettre en place l'environnement de développement	développer mon prototype	Moodle en place, IDE fonctionnel, projet GIT créé, création d'un plugin webservice	700	50	2	accepted
13	Développement	Etudiant / Professeur	Avoir une fenêtre de chat sur moodle	Pouvoir parler au chatbot depuis l'interface moodle	Avoir une interface sur le tableau de bord	650	20	2	accepted

Figure 70 User stories sprint 2

ID US.task	Task Name	Resp.	Initial Estimate
10.1	Rédiger une grille de critère	Célia	4
10.2	Recherches sur les solutions	Célia	8
10.3	Analyse comparative	Célia	8
10.4	Rédaction	Célia	4
12.1	Installer moodle	Célia	5
12.2	Création git	Célia	1
12.3	Installer ngrok	Célia	2
12.4	Création d'un web service	Célia	16
12.5	Test avec Dialogflow	Célia	20
12.6	Lecture doc moodle	Célia	4
13.1	Création d'un plugin block	Célia	8
		Remaining	80

Figure 71 Tâches sprint 2

10.4.4. Sprint 3

Le sprint 3 a été réservé pour le développement propre des cas d'applications et les finitions. Nous avons estimé 129 heures pour ces tâches.

US N°	Thème		User Stories	Acceptance Criteria	Priority	Status	Story Points	Sprint	US accepted (done done)	
11	Ecriture	Réalisatrice du TB	Rédiger le chapitre Design	avancer de manière visible dans le TB	chapitre rédigé	750	●	20	3	accepted
18	Développement	Etudiant	Demander des quiz	Pour m'améliorer	Interaction question-reponse entre bot et étudiant	600	●	50	3	accepted
19	Développement	Professeur	Demander la clé d'inscription d'un cours	Pour perdre moins de temps	clé retournée si professeur editeur	550	●	40	3	accepted
14	Ecriture	Réalisatrice du TB	Rédiger le chapitre Developpement	Avoir par écrit le développement réalisé	chapitre rédigé	500	●	10	3	accepted
15	Ecriture	Réalisatrice du TB	Rédiger le chapitre Bilan	Donner un retour sur l'implémentation	chapitre rédigé	450	●	20	3	accepted
16	Ecriture	Réalisatrice du TB	Rédiger le chapitre Conclusion	avancer de manière visible dans le TB	chapitre rédigé	400	●	10	3	accepted
17	Finalisation	Réalisatrice du TB	Finitions	s'assurer de ne pas commettre d'erreur	relecture faite, document imprimé	350	●	4	3	accepted

Figure 72 User stories sprint 3

ID US.tas	Task Name	Resp.	Initial Estimate
11.1	Diagrammes d'activités	Célia	8
11.2	Création d'intentions	Célia	8
11.3	Créations d'entité	Célia	4
11.4	Créations context	Célia	12
11.5	Rédaction du chapitre	Célia	4
18.1	création de données moodle	Célia	4
18.2	création des requêtes sql	Célia	8
18.3	création des méthodes	Célia	16
19.1	création de données moodle	Célia	4
19.2	création des requêtes sql	Célia	8
19.3	création des méthodes	Célia	16
14.1	Rédaction du chapitre développement	Célia	8
15.1	Rédaction du chapitre bilan	Célia	8
16.1	Rédaction du chapitre conclusion	Célia	8
17.1	Controler les sources, orthographe, syntaxe	Célia	8
17.2	Création pdf	Célia	1
17.3	Création du poster	Célia	4
		Total	123

Figure 73 Tâches sprint 3

10.5. Difficulté d'application / Remarques

Nous avons trouvé de la difficulté à poser les sprints. Nous avons finalement décidé que les sprints auraient une durée égale de trois semaines chacun. Cependant, nous n'avons pas le même nombre de jours à disposition pour la réalisation du travail, soit huit jours pour le sprint 0, neuf jours pour le sprint 1, onze jours pour le sprint 2, et quinze jours pour le sprint 3. Cela a mené un peu de confusion quant à la bonne exécution de la méthode agile.

Un des objectifs principaux de la méthodologie SCRUM est de pouvoir calculer la vitesse d'avancement de l'équipe pour avoir une idée du temps restant à la réalisation du projet. Ici le temps à disposition est défini à l'avance et peu importe le nombre de *story points* totaux définis dans le *product backlog*, tous devront être réalisés. De plus, comme les nombres de jours de travail augmentent du premier au dernier sprint, la rapidité calculée augmente également, ce qui est peu pertinent.

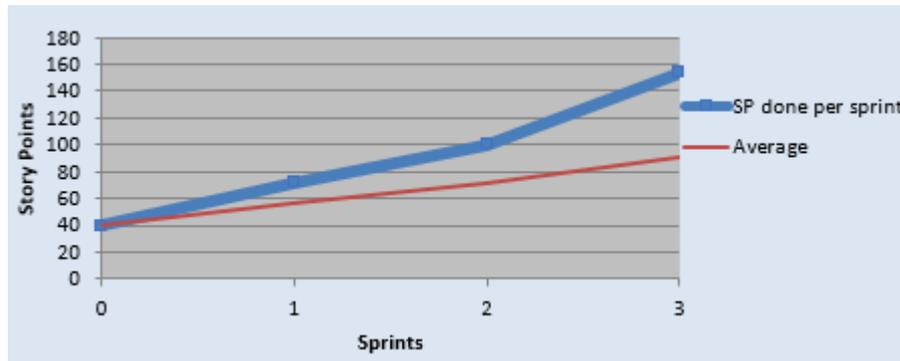


Figure 74 Rapidité du projet

Comme nous ne connaissions pas du tout le sujet, il a été difficile de définir précisément les *user stories* en rapport avec le développement. Nous avons dû les ajouter ultérieurement. Le même problème a été rencontré pour la définition des tâches du deuxième et troisième sprint. La plupart s'ajoutait au fur et à mesure des connaissances acquises, ce qui a compliqué la tenue du fichier que nous utilisons.

Pour conclure, l'utilisation de cette méthodologie nous semble, à notre avis, peu adaptée pour un projet de ce type. Elle nous a été utile pour garantir la segmentation du travail, le respect des délais fixés, et finalement posséder un détail de tout ce qui a été effectué.

11. Bilan

En premier lieu nous avons compris le fonctionnement d'un chatbot. En deuxième lieu, nous avons découvert quelques cas d'utilisation du bot dans le contexte de l'éducation. Nous avons pu, grâce à cela et une rencontre avec l'équipe de support Cyberlearn, proposer onze cas d'utilisation qui ont été classés par critères de faisabilité pour le développeur et utilité pour l'utilisateur (professeur ou étudiant) ainsi que pour l'équipe de support. De ces onze cas, nous en avons sélectionné deux : donner la clé d'inscription d'un cours et proposer des quiz.

Pour implémenter ces deux cas nous avons au préalable comparé quatre moteurs NLP disponibles sur internet. Nous avons choisi Dialogflow pour sa gratuité et sa flexibilité. Nous avons ensuite développé deux plugins Moodle : un plugin bloc pour disposer d'une interface de messagerie ; un plugin service web pour que les requêtes utilisateurs soient envoyées au backend Moodle. En parallèle à la création de notre chatbot sur Dialogflow, comprenant la définition d'entités, d'intentions et de contextes, nous avons ajouté plusieurs méthodes à

notre webservice de manière à ce que les bonnes réponses soient données en fonction des requêtes.

Nous avons finalement testé nos deux cas sous plusieurs conditions qui démontrent qu'ils sont fonctionnels. Nous constatons qu'un cas n'est pas géré pour les quiz ayant comme type de réponse « réponse courte » contrairement aux cas « à choix multiple » ou « vrai ou faux ».

12. Conclusion

12.1. Résultat de la thèse

Ce rapport permet de constater qu'il est effectivement possible d'intégrer un chatbot sur la plateforme Moodle HES-SO. Les bénéficiaires de cette intégration seraient les étudiants, les professeurs et aussi l'équipe de support.

Les avantages du chatbot commercial sont tout à fait transposables au chatbot dans un contexte éducationnel. De plus, il permettrait un engagement supérieur de la part des étudiants.

Nos recherches ont démontré que plusieurs chatbots ont été développés pour des universités. Ils peuvent combiner plusieurs rôles qui font partie en réalité de deux grands types de rôles : à but éducationnel proposant des exercices et ayant un impact sur l'apprentissage de l'étudiant ; à but non-éducationnel, pour de l'assistance ou du support académique.

Bien qu'il n'y ait pas à disposition de plugin chatbot conçu expressément par les développeurs Moodle, il est possible, par la création d'au minimum deux plugins (interface et webservice), d'avoir un prototype de chatbot fonctionnel.

Quatre moteurs NLP sur le web permettent de créer un chatbot disposant du traitement du langage naturel. Ils obtiennent une moyenne proche sur des critères basés sur la documentation, les coûts et les fonctionnalités. Dialogflow, le moteur sélectionné, nous a permis la création d'un prototype pour un cas d'assistance ainsi qu'un cas d'aide contextuelle. Le prototype ne prend pas en charge la gestion de session, et dispose d'une interface de messagerie non personnalisable. Ces aspects peuvent être améliorés par la création d'un client complet.

12.2. Recommandations

Pour un développement futur, nous préconisons le développement complet d'un client en PHP pour les raisons déjà mentionnées au point 6.3.2.1:

- la gestion intégrale des intentions, entités et contextes ;
- la possibilité de créer sa propre interface ;
- la possibilité de gérer les sessions.

Le premier point permettrait de résoudre le problème rencontré lors de nos simulations. Les entrées utilisateurs possibles pourraient provenir d'un dictionnaire et il ne serait pas obligatoire de toutes les noter à la main.

Pour le deuxième point, l'objectif serait de prendre en considération l'importance du degré d'anthropomorphisme sur la perception des utilisateurs. Il serait même envisageable, comme le T-Bot décrit dans ce rapport, de rendre le chatbot personnalisable par les professeurs, ou même selon les cours.

Nous proposerions également de tester le moteur wit.ai qui est open source.

12.3. Perspectives d'avenir

Nous l'avons vu, intégrer un chatbot de type conversationnel requiert de bonnes connaissances en programmation. Nous avons développé un prototype qui intègre deux fonctionnalités en 360 heures dédiées. Le développement d'un client complet avec plus de fonctionnalités demanderait bien plus de temps. Grâce à ce travail, nous avons cependant pu noter que Cyberlearn pourrait réellement bénéficier de cet outil si d'autres des cas décrits en annexes sont implémentés.

Le chatbot est devenu l'outil incontournable. Cependant, aujourd'hui on parle beaucoup du voicebot, le chatbot vocal. Bien que ses performances ne soient pas encore optimales, il est plus facile d'utilisation, plus intuitif et plus accessible. Il pourrait devenir un concurrent du chatbot et nous pensons qu'il faudrait s'y intéresser de plus près. (Séguéla, 2018) Les moteurs NLP comparés dans ce travail dont Dialogflow proposent également cette fonctionnalité, ce qui pourrait être un plus pour le centre Cyberlearn dans le futur.

13. Références

A Visual History Of Chatbots. (s.d). Récupéré le 4 juin 2018 sur <https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2?gi=83ddec226aa1>

AdmitHub launches first college chatbot with Georgia State - AdmitHub. (2016, 17 mai). Récupéré sur <https://www.admithub.com/blog/admithub-launches-first-college-chatbot-with-georgia-state/>

Akshay. (2018, 22 août). *Which is the best platform to develop a Chatbot in seconds.* Récupéré sur <https://medium.com/@akshay.kalkur/which-is-the-best-platform-to-develop-a-chatbot-in-seconds-d6a34bb748b>

Alammar, J. (2016, 14 décembre). *A Visual and Interactive Guide to the Basics of Neural Networks.* Récupéré sur <http://jalammar.github.io/visual-interactive-guide-basics-neural-networks/>

Ashcroft, A. (2019, 14 mai). *The Real Purpose of a Chatbot.* Récupéré sur <https://chatbotsmagazine.com/the-real-purpose-of-a-chatbot-b970105e3a7e?gi=9f523ff02cde>

Beck, B. (2018, 10 septembre). *AI Options Compared for NLP: Which to Use for Your Marketing Chatbot.* Récupéré sur <https://www.clearvoice.com/blog/ai-nlp-options-compared-for-marketing-chatbot/>

Beck, B. (2019, 14 janvier). *How to Write a Script for a Chatbot: Key Elements for Good Dialogue Flow.* Récupéré sur <https://www.clearvoice.com/blog/how-to-write-a-chatbot/>

Beillaud, E. (s.d). *Pas de chatbot sans intelligence artificielle et machine learning !!* Récupéré le 16 mai 2019 sur <https://www.nexworld.fr/chatbot-intelligence-artificielle/>

Britz, D. (2015, 15 septembre). *Recurrent Neural Networks Tutorial.* Récupéré sur <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>

- Burns, E. (2017, décembre). *Machine Learning vs Deep Learning ? La même différence qu'entre un ULM et un Airbus A380* Récupéré sur <https://www.lemagit.fr/conseil/Machine-Learning-vs-Deep-Learning-un-avion-a-helices-et-un-avion-a-reaction>
- Cantet, T. (2018, 25 janvier). *Comparatif des moteurs de chatbot*. Récupéré sur <https://blog.octo.com/comparatif-moteurs-chatbots/>
- Chatbot Anna*, (s.d). Récupéré le 10 mai 2019 sur https://www.chatbots.org/virtual_assistant/anna_sweden/
- Chatbot Concept - Dragon1*. (s.d). Récupéré le 10 juin 2019 sur <https://www.dragon1.com/concepts/chatbot>
- Chatbot et Big Data : le lien entre agents conversationnels et données*. (2019, 18 janvier). Récupéré sur <https://www.lebigdata.fr/chatbot-et-big-data>
- Coppex, S. (2018) *Chatbot : Le pont entre clients et professions libérales*.
- De Clisson, T. (2018, 30 novembre). *6 avantages d'un chatbot pour améliorer la relation client*. Récupéré sur <https://www.journaldunet.com/ebusiness/expert/70179/6-avantages-d-un-chatbot-pour-ameliorer-la-relation-client.shtml>
- Dehler Zufferey, J. (2018, 1 29). *Apprendre avec un chatbot, ça marche ?* Récupéré sur <https://www.journaldunet.com/management/expert/68411/apprendre-avec-un-chatbot--ca-marche.shtml>
- Dieng, J.-B. (2017, 16 juillet). *La tendance des chatbots, une nouvelle arme marketing ?* Récupéré sur <https://medium.com/@jbdieng/la-nouvelle-tendance-des-chatbots-e854bec81fa3>
- Expert System. *What is Machine Learning? A definition*. (2017,10 mai). Récupéré sur <https://www.expertsystem.com/machine-learning-definition/>
- Garcia Brustenga, G., Fuertes-Alpiste, M., Molas-Castells, N. (2018). Briefing paper: chatbots in education. Barcelona: eLearn Center. Universitat Oberta de Catalunya.
- Genier, C. (2018, 7 mars). *Les chatbots vont-ils révolutionner la formation ?* Récupéré sur <https://jobtic.ch/fr/magazine/article/1633>

- Goyal, P., Pandey, S., & Jain, K. (2018). *Deep Learning for Natural Language Processing: Creating Neural Networks*. Bangalore: Apress.
- Gu, M. (2018, 20 Novembre). *Recurrent Neural Networks for Language Understanding*. Récupéré sur <https://towardsdatascience.com/recurrent-neural-networks-for-language-understanding-10c649f8ac15?gi=c79862473d7e>
- Adit, J. (2017, 7 janvier) *How to go about developing my own chatbot - Quora*. Récupéré sur <https://www.quora.com/How-do-I-go-about-developing-my-own-chatbot>
- Mohanoor A. (s.d) *How to integrate Dialogflow into your website - Mining Business Data*. Récupéré le 13 juin 2019 sur <https://miningbusinessdata.com/how-to-integrate-Dialogflow-into-your-website/>
- Ilhe, N. (2018, 26 mai). *Les Chatbots pour les nuls*. Récupéré sur <https://medium.com/webotit/les-chatbots-pour-les-nuls-3e57a81242b0>
- Kang, A. (2017, 6 avril). *Understanding the Differences Between Alexa, API.ai, WIT.ai, and LUIS/Cortana*. Récupéré sur <https://medium.com/@abraham.kang/understanding-the-differences-between-alexa-api-ai-wit-ai-and-luis-cortana-2404ece0977c>
- Karmali, F. (2018). *Chatbot in Moodle*. Récupéré sur <https://assets.moodlemoot.org/sites/44/20180103073353/Creating-a-chatbot-in-Moodle-presented-by-Mr.-Farhan-Kamali.pdf>
- Kompella, R. (2018, 9 février). *Conversational AI chat-bot — Architecture overview*. Récupéré sur <https://towardsdatascience.com/architecture-overview-of-a-conversational-ai-chat-bot-4ef3dfefd52e>
- König, J. (2017, 15 février). *ngrok: How To Speed Up Your Chatbot Development*. Récupéré sur <https://www.chatbot-academy.com/ngrok-chatbot-development/>
- L, B. (2017, 24 mars). *Test de Turing; Un test pour mesurer l'intelligence artificielle*. Récupéré sur <http://www.artificiel.net/test-de-turing>
- L'ADN, (2016, 20 octobre) *Le chatbot se met au service de la marque*. Récupéré sur <https://www.ladn.eu/non-classe/le-chatbot-se-met-au-service-de-la-marque/>

- Leblal, S. (2016, 10 octobre). *Edito : Déjà le trop plein de chabots*. Récupéré sur <https://www.lemondeinformatique.fr/actualites/lire-edito-deja-le-trop-plein-de-chabots-66177.html>
- Mikic Fonte, F., Burguillo Rial, J., & Llamas Nistal, M. (2008). *T-BOT and Q-BOT: A Couple of AIML-based*. Saragota Springs.
- Millon, L. (2017, 7 juillet). *Zo : encore un chatbot de Microsoft qui a des propos violents*. Récupéré sur <https://siecledigital.fr/2017/07/06/zo-encore-un-chatbot-de-microsoft-qui-des-propos-violents/>
- Mouadil, M. (2018, février 2018). *Introduction au Deep Learning : les réseaux de neurones*. Récupéré sur <https://meritis.fr/ia/deep-learning/>
- Moodle Programming: Part 5 - Creating a basic block. (2014, July 26). [YouTube]. Récupéré sur <https://www.youtube.com/watch?v=YTnMHWyfMKM>
- Pandey, S., Goyal, P., & Jain, K. (2018). *Deep Learning for Natural Language Processing: Creating Neural Networks*. Bangalore: Apress, Berkeley, CA.
- Peters, F. (2018). Master Thesis : Design and implementation of a chatbot in the context of customer support. Liège.
- Rebuel A. (2018). *Point de vue d'expert sur l'essor des bots conversationnels : Interview de Jean-Paul Muller - Directeur de "La Forge" chez GFI Informatique*. Récupéré sur <https://mbamci.com/point-de-vue-expert-essor-bots-conversationnels/>
- Qu'est-ce que le machine learning ? – Glossaire HPE et définitions*. (s.d) Récupéré le 26 mai 2019 sur <https://www.hpe.com/ch/fr/what-is/machine-learning.html>
- Reboulleau, M. (2017, décembre). *Le Chatbot dans le management de la relation client*. Récupéré sur <https://fr.slideshare.net/MartinReboulleau/le-chatbot-dans-le-management-de-la-relation-client>
- Rietz, T. Benke, I. Maedche, A. (2019). The Impact of Anthropomorphic and Functional Chatbot Design Features in Enterprise Collaboration Systems on User Acceptance. Récupéré sur https://chatbotresearch.com/wp-content/uploads/2019/02/wi2019_2.pdf

- Rodina, S. (2016, 7 décembre). *Les Chatbots attaquent le secteur de l'éducation*. Récupéré sur <http://www.mabucom.ch/les-chatbots-attaquent-le-secteur-de-leducation/>
- Roman, D. (2017, 1^{er} juin). *Tout sur le machine learning et le deep learning*. Récupéré sur <https://medium.com/@davidroman00/tout-sur-le-machine-learning-et-le-deep-learning-1d8b71b2d584>
- Roos, S. (2018). *Chatbots in education A passing trend or a valuable pedagogical tool ?* Uppsala.
- Rouse, M. (s.d) *What is Natural language processing (NLP)? A definition from WhatIs...* Récupéré le 14 mai 2019 sur <https://searchbusinessanalytics.techtarget.com/definition/natural-language-processing-NLP>
- Scrum (développement)*. (s.d). Récupéré le 3 juillet 2019 sur [https://fr.wikipedia.org/wiki/Scrum_\(d%C3%A9veloppement\)](https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement))
- Séguéla, P. (2018, 10 septembre). *Les voicebots, l'avenir des chatbots ?* Récupéré sur <https://www.journaldunet.com/solutions/expert/69700/les-voicebots--l-avenir-des-chatbots.shtml>
- Seif, G. (2018, 20 novembre). *An easy introduction to unsupervised learning with 4 basic techniques*. Récupéré sur <https://towardsdatascience.com/an-easy-introduction-to-unsupervised-learning-with-4-basic-techniques-897cb81979fd>
- Seydtaghia, A. (2016, 15 septembre). *Mes conversations avec des chatbots*. Récupéré sur <https://www.letemps.ch/economie/conversations-chatbots>
- Shetty, B. (2018). *Supervised Machine Learning: Classification*. Récupéré sur <https://towardsdatascience.com/supervised-machine-learning-classification-5e685fe18a6d?gi=255dad97bf1>
- The Social Client. (2016). *Tout comprendre des chatsbots*. Récupéré sur <https://www.frenchweb.fr/wp-content/uploads/2017/01/Livre-Blanc-Bots-VD.pdf>

Tual, M. (2016, 24 mars). *A peine lancée, une intelligence artificielle de Microsoft dérape sur Twitter*. Récupéré sur https://www.lemonde.fr/pixels/article/2016/03/24/a-peine-lancee-une-intelligence-artificielle-de-microsoft-derape-sur-twitter_4889661_4408996.html

Veizenbaum, J. (1996). *Computational linguistics* (Number 1, Vol 9). Récupéré sur <https://web.stanford.edu/class/linguist238/p36-weizenbaum.pdf>

Weinberger, M. (2016, 25 avril). *Bill Gates says the next big thing in tech can help people learn like he does*. Récupéré sur <https://www.businessinsider.fr/us/bill-gates-on-chatbots-in-education-2016-4>

14. Annexes

Annexe I – Comparatif moteurs NLP

	Nom du moteur	Documentation / Communauté	Intégration et langage de développement	Fonctionnalités	Cas d'applications	Tarifs	Langages	Total						
Points		10	10	5		5		5						
	Wit.ai	Assez bonne documentation, disponible sur le site officiel, toutes les étapes sont décrites, open source très grande communauté - uniquement en anglais - pas de support + blog	Nodjs client Python client Ruby client Possibilité également d'intégrer sur son website via HTTP API	Création de story UI fourni pour la visualisation de flux de dialogue	Pas d'information	Gratuit, open source utilisation commercial incluse	Afrikaans, Akan, Albanian, Amharic, Arabic, Armenian, Assamese, Aymara, Azerbaijani, Basque, Belarusian, Bengali, Bosnian, Breton, Bulgarian, Burmese, Catalan, Cherokee, Chichewa, Chinese, Croatian, Czech, Danish, Divehi, Dutch, English, Esperanto, Estonian, Faroese, Finnish, French, Fulah, Galician, Ganda, Georgian, German, Greek, Guarani, Gujarati, Haitian, Hausa, Hebrew, Hindi, Hungarian, Icelandic, Igbo, Indonesian, Inuktitut, Irish, Italian, Japanese, Javanese, Kalaallisut, Kannada, Kashmiri, Kazakh, Khmer, Kinyarwanda, Kirghiz, Korean, Kurdish, Lao, Latin, Latvian, Limburgish, Lingala, Lithuanian, Macedonian, Malagasy, Malay, Malayalam, Maltese, Maori, Marathi, Mongolian, Nepali, Northern Sami, Norwegian, Norwegian (bokmal), Oriya, Oromo, Panjabi, Pashto, Persian, Polish, Portuguese, Quechua, Raeto-Romance, Romanian, Russian, Sanskrit, Sardinian, Scottish Gaelic, Serbian, Shona, Sicilian, Sindhi, Sinhalese, Slovak, Slovenian, Somali, Sotho, South Ndebele, Spanish, Sundanese, Swahili, Swati, Swedish, Tagalog, Tajik, Tamil, Tatar, Telugu, Thai, Tibetan, Tigrinya, Tsonga, Tswana, Turkish, Uighur, Ukrainian, Urdu, Uzbek, Venda, Vietnamese, Welsh, Western Frisian, Wolof, Xhosa, Yiddish, Yoruba, and Zulu		8	9	5		5	32
	DialogFlow https://dialogflow.com/	Très bonne documentation / tutoriaux exemples / Grande communauté + français + support	Nodjs client Python client Java GP Ruby C# PHP Système de fulfillment pour connecter le bot à ses propres services, base de données / Guide en nodejs / Intégration de l'interface dialogFlow sur le website possible	Produit par Google machine learning, Entités, intention, Intentions déjà prédéfinies qu'il est possible d'utiliser, small talks. Pas d'UI pour les flux conversationnel mais paramètres en requête	KLM Domino Piizza Ticket Master	édition standard gratuite, payante pour l'édition entreprise	Chinese, Danish, Dutch, English, French, German, Hindi, Indonesian, Italian, Japanese, Korean, Norwegian, Polish, Portuguese, Russian, Spanish, Swedish, Thai, Turkish, Ukrainian	10	10	5		4	5	34
	IBM watson	Très bonne documentation / tutoriaux exemples / + en français + support + blog Site dédié aux questions des développeurs https://developer.ibm.com/answers/smartspace/watson/index.html	Création de clients possibles Node Python Ruby, Swift, Curl, Go, Java, PHP	Intégration avec tous les types de chaînes, gestion de requêtes complexes, historiques des conversations, tableaux de bord analytiques	Bank U RoyalBank of Scotland Autodesk	Version gratuite qui comprend 10 000 messages/mois, 5 compétences et 100 nœuds de dialogues	Anglais, arabe, chinois, tchèque, néerlandais, français, allemand, italien, japonais, coréen, portugais (Br), espagnol	10	10	5		4	5	34
	Recast.ai	Documentation très complète et grande communauté (slack communauté) et plus de 100 000 chatbot développés disponible sur leur plateforme collaborative / blog	SDK for client Python, Nodejs, PHP, Ruby, Android, Swift	Fonctionne à partir de la définition de skills et de training dataset. Lié à Github, small talks	Bouygues Telecom, SFR, Groupe mutuel	Gratuit	English, French, Spanish, German	10	10	4		5	5	34

Annexe II – Faisabilité cas support

	Utilisateur	Intitulé	Problématique	Faisabilité
1	Professeur	Création d'un cours	Un professeur avait la possibilité de créer lui-même son cours. En donnant un nom de cours aléatoire, il arrivait souvent qu'il ne retrouve pas son cours et en créait un nouveau, ce qui provoquait des doublons dans la base de données. Aujourd'hui c'est l'équipe de support qui s'en occupe. Les interactions avec les professeurs se font principalement par email et des aller-retours par email ou téléphone sont fréquents.	Le flux conversationnel ici serait assez simple. Le chatbot demanderait les informations nécessaires pour la création d'un cours avec les contrôles nécessaires (doublons, noms exacts) et les enregistrerait en base de donnée. Ces informations pourrait être récupérées par un membre de l'équipe de support. On pourrait envisager un développement de plugin type "derniers cours à créer" qui irait chercher en base de données les demandes de création non traitées. Le membre de l'équipe de support pourrait les statuer à "effectué" dès que le cours a été créé. On pourrait également envisager le processus fait de A à Z par le bot, mais la solution précédente permet de décharger le support tout en évitant des aller-retours par téléphone car toutes les informations nécessaires sont données.
2	Utilisateurs	Création de compte pour membres externes	Le help desk doit créer un compte pour les membres n'ayant pas de compte AAI	Ici, il faudrait imaginer un chatbot sur la page d'accueil de cyberlearn, car les utilisateurs ne sont pas connectés. Le chatbot demanderait les informations nécessaires pour la création d'un compte et irait directement créer le compte en base de données.
3	Professeur	Indiquer quelle est la clé d'inscription au cours	Les étudiants, pour avoir accès au contenu d'un cours doivent insérer la clé d'inscription. Le professeur doit la leur fournir mais ne sait pas où la trouver dans les paramètres du cours.	Ici il faut rendre le chatbot capable de fournir cette clé pour autant que l'utilisateur soit un professeur, et que ce dernier soit le professeur officiel du cours en question. Cela implique un contrôle d'authentification qui sera réalisé par le chatbot. Le flux de conversation serait assez simple à imaginer.
4	Professeur	Paramétrer des activités et ajouter des restrictions	Beaucoup d'options sont possibles pour paramétrer les activités. Bien que chaque paramètre dispose d'une description, les professeurs appellent le help desk.	Ici cela implique précision sur les cas à prendre en considération car il y en a autant que de paramètres dans toutes les activités. Il conviendrait donc de bien définir avec l'équipe cyberlearn quels sont les cas utiles. Le chatbot peut tout faire en écriture en base de donnée si les cas sont bien définis.
5	Professeur	Inscrire des participants à un cours	L'inscription de participants à un cours peut se faire grâce à un fichier csv. Les professeurs n'y ont pas accès	Le chatbot pourrait recevoir une liste (non pas un fichier excel) du professeur des différents étudiants. Après segmentation de la liste, il irait inscrire (en base de données) chacun des étudiants au cours en question. Mieux: le chatbot recevrait directement un csv. (plus compliqué à traiter).
6	Professeur ou étudiant	Changement de mot de passe	Si une personne a un compte qui a été créé manuellement alors le help desk peut changer son mot de passe. Sinon, il doit rediriger l'utilisateur vers le service informatique de la HES-SO où l'utilisateur suit son cursus.	Ici, il faudrait imaginer un chatbot sur la page d'accueil de cyberlearn, car les utilisateurs ne sont pas connectés. Le chatbot demanderait les informations, contrôlerait si l'inscription du compte à été faite manuellement. Si oui, il peut changer le mot de passe, sinon, il renverrait le contact du service informatique en question.

Annexe III – Utilité / Faisabilité cas étudiant

	Utilisateur	Intitulé	Description	Utilité	Faisabilité
1	Etudiant	Renvoyer des dates selon le calendrier	Chaque étudiant dispose d'un calendrier. Les différentes dates de rendus désirées par les professeurs s'y annotent automatiquement. L'étudiant pourrait être alerté par le bot quelques heures avant l'échéance ou demander à quel moment il doit rendre un travail.	Un système de notification existe déjà sur Moodle pour les échéances, de plus, il n'y a pas vraiment de gain de temps entre demander au bot les échéances pour tel ou tel cours ou simplement se rendre de dans le calendrier pour voir les échéances	Difficile à ce stade de concevoir un bot qui engagerait la conversation comme il réagit aux inputs de l'utilisateur. Il pourrait être activé justement par le système event (par les notifications).
2	Etudiant	Insérer une échéance dans le calendrier	Le chatbot pourrait inscrire des dates fournies par l'étudiant dans le calendrier	Même chose que cas précédent, pas de réel gain de temps pour l'étudiant	Ici, il faudrait imaginer un flux de conversation ou une commande fixe envoyée par l'utilisateur et une simple action d'écriture dans la base de données.
3	Etudiant	Proposer un quizz à l'étudiant pour une matière	Recherche et proposition de quizz auxquels l'étudiant a accès. Envoie les questions, reçoit les réponses et calcule le score de l'étudiant en fonction des réponses données.	L'étudiant n'aurait pas besoin de se rendre dans les différents cours pour savoir lequel dispose d'un quizz (les cours sont souvent difficile à identifier car ils ont des noms fixes commençant tous par HES-...). Il pourrait directement sélectionner le cours et choisir le quizz.	Pas de problème particulier à définir le flux de conversation et récupérer les données (quizz et cours) en base de données. La difficulté réside dans l'envoi de questions et de réception de réponses adaptable à n'importe quel quizz.
4	Etudiant	Montrer ses notes	Le chat bot renvoie les notes pour les cours ou pour tous les cours	Peut être utile pour autant que les notes soit notée sur la plateforme, ce qui est rarement le cas actuellement	Pas de difficulté particulière
5	Etudiant	Répondre à n'importe quelles questions FAQ	Le bot agit en agent de support académique, questions relatives à la formation, et aide l'étudiant à utiliser la plateforme	L'étudiant peut de la même manière se rendre dans la section FAQ. Il n'y a pas de gain particulier.	Pas de difficulté particulière

15. Déclaration de l'auteur

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seule, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré.

Célia Ahmad, le 31 juillet 2019