

Mémoire de fin d'études

DIALOGUE HOMME-MACHINE

par

ADRIEN DOS SANTOS

Élève master 2 MIAGE

M.	Henry Jammes	Tuteur de stage
Pr.	Emmanuel Hyon	Professeur référent

Résumé

Depuis des années, on espère voir arriver l'ère de l'intelligence artificielle où la machine ressemblerait à l'homme jusqu'au point où on ne saurait pas faire la différence.

Aujourd'hui encore avec la science et la technologie qui évoluent de jour en jour, de nouvelles pratiques naissent pour s'y rapprocher.

L'intelligence artificielle est devenue omniprésente sous différents aspects et niveaux d'intelligence dans la société. Elle s'impose dans les agents conversationnels ou chatbots pour offrir divers services aux utilisateurs. L'objectif de ce mémoire est de focaliser sur la compréhension du dialogue homme-machine amenant à la création d'un chatbot et comprendre les limites actuelles.

Remerciements

Je voudrais remercier toute l'équipe CRM d'Avanade pour leur accueil. J'ai passé grâce à eux un stage agréable dans un bonne ambiance de travail, et je les remercie pour l'aide et les conseils qu'ils m'ont apportés tout au long de ce stage.

Je remercie tout particulièrement mon tuteur de stage Henry JAMMES ainsi que Matthieu BOLLECKAR, Redwane ID HAMMOU et Mathilde LAFONT pour l'aide et les conseils prodigués tout au long de ce stage. Je tiens également à remercier Emmanuel HYON, tuteur enseignant, pour ses conseils concernant l'approche de mon mémoire.

Pour finir, je souhaite remercier l'ensemble de l'équipe pédagogique qui m'a permis de renforcer mes connaissances pour mener à bien mon stage de fin d'études.

Table des matières

Introduction	11
1 Contexte et problématique	13
1.1 Contexte : L'essor des agents conversationnels	13
1.1.1 L'intelligence artificielle	13
1.1.2 L'agent conversationnel	15
1.1.3 Cas d'utilisation	15
1.2 Problématique	16
2 Concept	19
2.1 Dialogue homme-machine	19
2.2 Natural Language processing	20
2.2.1 But	20
2.2.2 Principes	21
2.3 Natural language understanding	21
2.3.1 Phonologie	22
2.3.2 Morphologie	23
2.3.3 Lexical	23
2.3.4 Syntaxique	23
2.3.5 Sémantique	24
2.3.6 Discours	24
2.3.7 Pragmatique	25
2.3.8 Cognitive	25
2.3.9 Base de connaissance	25
2.4 Natural language Generation	25
2.5 Dialogue chatbot	26
3 Etat de l'art	29
3.1 Etudes et expérimentation sur le TAL via NLTK	29
3.1.1 TAL	29
3.1.2 NLTK	30
3.2 Etudes et expérimentation sur l'AIML	32
3.2.1 Catégories	32
3.2.2 Récursivité	33
3.3 Etudes et expérimentation sur les intents	36
3.3.1 Exemple d'intent	36
3.3.2 Modèle	37

3.3.3	Utilisation du modèle	37
3.4	Evaluations	40
3.5	Etat de l'art des solutions	41
4	Implémentation d'un agent conversationnel	43
4.1	Méthodologie	43
4.1.1	Microsoft bot framework	43
4.1.2	LUIS	43
4.2	Cas concret : incident dans CRM	47
4.3	Développement	48
4.3.1	Architecture	48
4.3.2	Paramétrage	50
4.4	Résultat	51
4.5	Discussion	53
4.6	Limite de la solution	54
5	Bilan	55
5.1	Conclusion	55
5.2	Evolution possibles	55
	Bibliographie	57
	Table des figures	59
A	Code traitement des intents	61
B	Dialogue avec le chatbot	63

Acronymes

IA	Intelligence Artificielle
IHM	Interface Homme Machine
NLP	Natural Language Processing
NLU	Natural Language Understanding
NLG	Natural Language Generation
AIML	Artificial Intelligence Markup Language
TAL	Traitement Automatique des Langues
NLTK	Natural Language ToolKit
LUIS	Language Understanding Intelligent Service
CRM	Customer Relationship Management

Introduction

Depuis les années 50, avec le test d'Alan Turing pour tester le niveau d'intelligence artificielle d'un robot, on ne cesse d'inventer et de créer des robots qui sont de plus en plus performants dans la manière d'analyser et de répondre dans un dialogue homme-machine. Le but recherché dans l'évolution de ce dialogue est d'arriver à un point où on aurait du mal à distinguer si on dialogue avec un robot ou une personne [1].

En parallèle de l'intelligence artificielle, la technologie continue à évoluer. On a maintenant une expansion des smartphones et d'accès à internet, ce qui ouvre à un très grand public. Sur ces supports, on retrouve de l'IA que ce soit des agents virtuels sur internet comme Amélia de chez IPsoft, des assistants vocaux comme Siri de Apple, des moteurs de recherche comme Google et divers chatbots sur différents canaux notamment Facebook messenger. A travers tous ces agents de conversations, le but recherché est de répondre le plus précisément possible aux questions des utilisateurs.

Ayant été amené à travailler sur les chatbots lors de mon stage de fin d'études, j'ai souhaité approfondir le sujet dans le cadre du mémoire.

On se dirige vers un avenir où l'IA sera omniprésente car elle sera plus ou moins aussi intelligente qu'un humain dans sa manière de raisonner et pourra exécuter des tâches humaines compliquées ou physiquement dures.

On peut observer la victoire de l'IA AlphaGO de la société DeepMind, filiale de Google, battre le champion du monde du jeu de go en Mars 2017 où « le nombre de combinaisons dans ce jeu est supérieur au nombre d'atomes que compte l'univers » d'après Demis Hassabis, l'un des fondateurs de la société.

Ce sujet m'intéresse car il consiste à comprendre comment arriver à créer un bot qui répond à un besoin en fonction du dialogue que l'on va utiliser.

Avec les avancées scientifiques, on a fait évoluer ce dialogue en le rendant plus complexe avec plus de choix en entrées et plus de réflexion pour répondre aux besoins des utilisateurs. Le problème réside dans la compréhension de ce que dit ou fait l'utilisateur. Tout au long de ce mémoire, nous étudierons les moyens possibles de compréhension du dialogue et les messages générés ; Le but étant de partir de l'existant pour s'en servir comme support et de comprendre jusqu'où la compréhension du dialogue a permis de créer un bot. L'analyse des différentes études nous permettra de mettre en évidence les améliorations et les limites du paramétrage d'un bot.

Chapitre 1

Contexte et problématique

1.1 Contexte : L'essor des agents conversationnels

1.1.1 L'intelligence artificielle

Ce qui est artificiel a généralement une connotation négative car on le perçoit comme moins bien que l'objet réel. Mais dans la réalité, l'objet artificiel est souvent au-dessus car plus performant. Si on prend le cas d'une fleur artificielle, la fleur ne requiert pas d'eau ni soleil et aura la même utilité qui est de décorer.

L'intelligence artificielle est comme la fleur, elle n'est pas naturelle car elle est la création de l'homme. Il faut comprendre le sens d'intelligence et le définir [2].

Une définition de R. Sternberg, psychologue et professeur de psychologie cognitive américain : “ L'intelligence est la capacité cognitive d'un individu pour apprendre de son expérience, bien raisonner, se rappeler des informations importantes et faire face aux demandes de la vie quotidienne.”

Le but de l'intelligence artificielle est de créer des programmes ou systèmes qui ont une pensée similaire à celle d'un humain. Il faut faire une distinction entre la pensée et l'intelligence car la pensée est la faculté de raisonner, analyser, évaluer et proposer des idées alors que les êtres capables de penser ne sont pas forcément intelligents.

Il y a des degrés d'intelligence parmi les humains, les humains et aussi les IA. C'est en 1950 qu'Alan Turing a établi une méthode pour évaluer celle d'une IA.

Aujourd'hui l'intelligence artificielle est basée sur des algorithmes de règles ainsi que sur de l'apprentissage sur le temps avec des données pour l'entraîner. Les bots ne répondent pas simplement aux questions des utilisateurs mais les guident pour avoir la question la plus précise et donc répondre le mieux aux besoins [3].

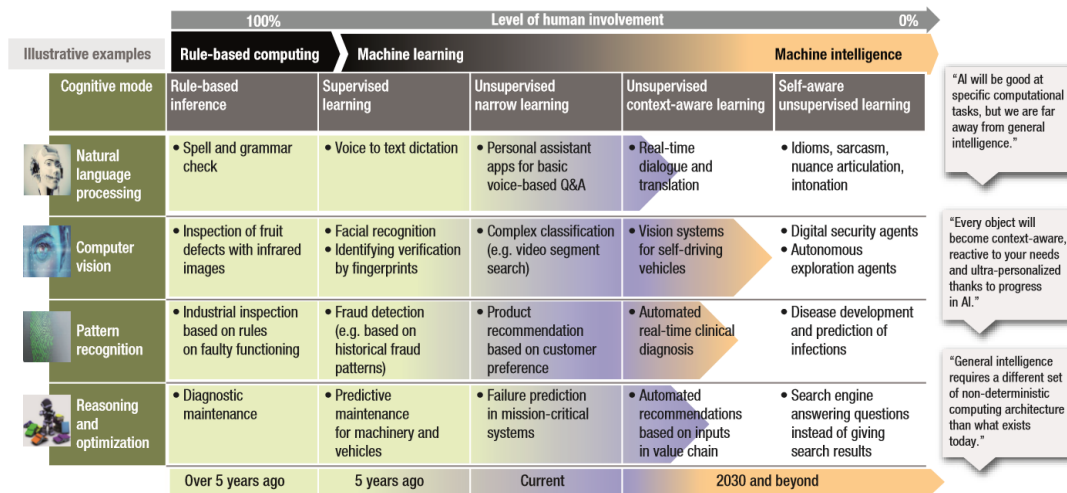


FIGURE 1.1 – Développement de l'IA et ses états

Dans le domaine du dialogue, l'IA est liée au natural language processing que l'on abordera dans la partie concept. On remarque, en se penchant sur l'évolution (figure 1.1), qu'on commence à avoir du dialogue temps-réel et il y a davantage de langues compréhensibles avec un passage de l'une à l'autre par la traduction. On cherche à atteindre un dialogue où l'on peut comprendre un sens caché d'un dialogue comme le sarcasme, l'humour ou encore l'intonation pour les dialogues oraux.

Ce progrès est dû en grande partie aux avancées de la puissance de calcul et les algorithmes d'apprentissage mais aussi au coût des ordinateurs et des espaces de stockages qui a fortement baissé.

Pour le moment, l'intelligence artificielle est vue comme un service pour beaucoup d'entreprises. En effet, elle aurait pour but de réduire les coûts de production, dynamiser des processus, offrir des produits et des services hyper-personnalisés basés sur les préférences des clients. "Messaging apps will introduce a paradigm shift for marketers... new conversational interfaces will drive deeper relationships between consumers and brands." Forrester.

Pour les dialogues, on est à l'ère des assistants personnels que l'on nomme agents conversationnels qui fournissent un service en dialoguant avec un utilisateur. Microsoft, Google, Facebook, IBM et Amazon commencent à investir dans ces agents pour utiliser leurs propres services. "'Conversational AI-first' will supersede 'cloud-first, mobile-first' as the most important, high-level imperative for the next 10 years, AI will be a 5B business by 2020." Gartner.

1.1.2 L'agent conversationnel

Un agent conversationnel/chatbot est un programme perçu par les utilisateurs comme une personne artificielle, un animal ou autre qui tient des conversations avec les humains. Cela peut être une conversation écrite, orale ou non verbale. [4]

Les chatbots sont un phénomène :

« Au cours des derniers mois, les chatbots ont investi le devant de la scène tech. Tous les grands noms du secteur ont apporté leur pierre à l'édifice, que ce soit au travers d'une nouvelle interface conversationnelle ou bien en développant une intelligence artificielle. Google a la volonté de prendre le leadership dans le domaine de l'intelligence artificielle, les annonces dans le domaine sont quasi-quotidiennes » [5].

On reparle des bots car facebook ouvre messenger aux bots en avril 2016 [6]. Mark Zuckerberf explique qu'avec l'IA et le natural language processing combiné à de personnes réelles, les gens parleront aux bots sur messenger comme s'ils parlaient avec leurs amis.

Les chatbots sont une nouvelle manière d'interagir avec des services existants, une expérience utilisateur novatrice qui laisse entrevoir de nombreuses opportunités.

De plus, il y a une demande très forte. Cela s'explique notamment par le fait que, depuis fin 2015, l'utilisation des applications de messagerie a surpassé celle des réseaux sociaux et l'utilisation des sms [4]. Cette présence massive et croissante des utilisateurs sur ces interfaces conversationnelles est donc un nouvel « Eldorado » pour les marques.

Cependant, les applications de messagerie ne sont pas la seule explication de cette démocratisation. L'intelligence artificielle est au cœur de la plupart des services conversationnels et au cours des précédentes années, ce domaine de l'informatique a considérablement progressé, entraînant avec lui les chatbots.

1.1.3 Cas d'utilisation

Dans les années 2010, on a les premiers systèmes de dialogue homme-machine grand public sur divers sites Internet, robots de compagnie, téléphones portables, agendas électroniques, systèmes de géolocalisation et autres assistants personnels.

Au niveau des systèmes grand public, on est loin d'un dialogue en langage naturel. Par exemple, les systèmes de géolocalisation et les téléphones portables en sont encore à la détection de mots-clés : noms de ville, noms de destinataires. Loin de la compréhension de « je veux aller à Grenoble en contournant Lyon et en évitant l'autoroute entre Saint-Etienne et Lyon »



FIGURE 1.2 – Différents agents conversationnels actuels

Il y a des chatbots internes aux entreprises, par exemple pour organiser un meeting, regarder qui est disponible, proposer un horaire disponible pour tous les participants, puis la réservation d'une salle. Le chatbot ne fait que consulter les calendriers et proposer des choix, le seul dialogue est celui pour proposer des participants, choisir une date et un lieu. Aucune autre information ne peut être demandée.

Et des chatbots offrent un service payant comme Pizza Hut sur facebook messenger. Il faut prendre contact avec le compte officiel Pizza Hut (être son ami). Quand le bot est sollicité, il présente trois options : passer commande, découvrir les promotions et « autre » (service client avec de vraies personnes). En passant commande, il est possible d'afficher les dernières commandes passées ou d'afficher le menu. Le paiement peut s'effectuer en ligne ou sur place. Quand on choisit de passer une commande, on entre dans un formulaire où le bot pose les questions et l'utilisateur répond. L'utilisateur ne pourra pas poser des questions et il sera bloqué.

D'autres chatbots :

- Bot météo : donne la météo quand on lui demande
- Bot actualités : on lui demande si des choses intéressantes se passent dans l'actualité
- Bot de vie : on lui dit nos problèmes et il essaye de trouver une solution
- Bot ami : En Chine, il y a un bot appelé Xiaoice créé par Microsoft, avec 20 millions qui lui ont parlé.

1.2 Problématique

Il y a deux types de chatbots : un simple avec des règles et l'autre plus intelligent avec du machine learning.

Le chatbot basé sur des règles est très limité. Il va répondre à des commandes spécifiques. Si ce qu'on lui dit ne fait pas partie de ces commandes alors il ne comprend pas.

Alors que le chatbot basé sur le machine learning a une IA. Il comprend en partie le langage naturel et n'a pas juste des commandes. De plus, le machine learning lui permet

d'apprendre en discutant avec les utilisateurs.

Comme nous l'avons vu plus haut, les cas d'utilisation sont pour l'instant basiques. En effet, le problème majeur du bot est de comprendre l'utilisateur. Le bot ne peut pas comprendre le langage naturel sauf si on lui a fait apprendre certaines phrases types. C'est pour ça que des formulaires ou encore une sélection d'actions sont mis en place dans un dialogue avec un bot pour avoir un dialogue à domaine fermé. (figure 1.3)

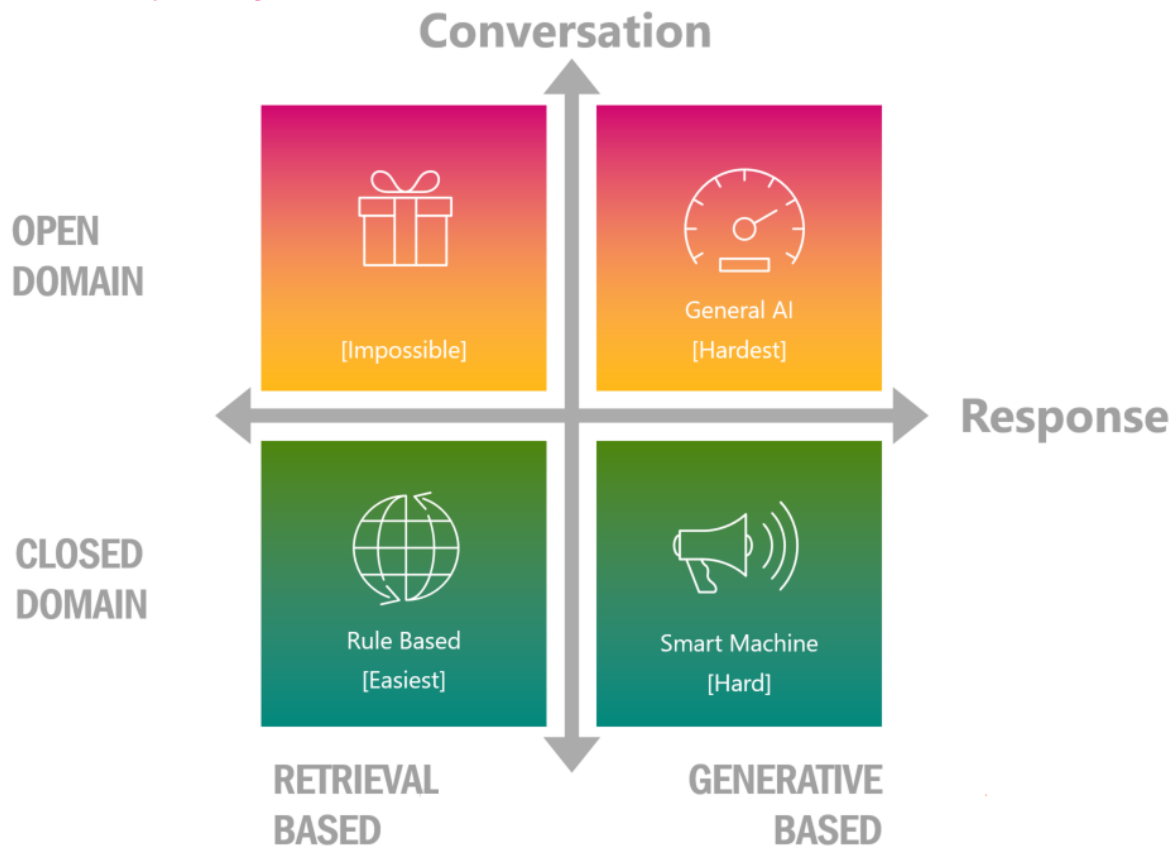


FIGURE 1.3 – Complexité d'un bot

Pour créer un bot, on doit réfléchir à la façon dont on veut le faire interagir avec l'utilisateur. Pour cela, on va se concentrer sur le dialogue homme-machine, avec la phase compréhension du langage humain qui correspondra au natural language understanding et on réfléchira aussi à la façon d'extraire les données pour être ensuite analysées et former une réponse qui sera la phase natural language generation.

Dans ce mémoire, on se posera la question suivante : en quoi comprendre le dialogue homme-machine de façon théorique permet-il de créer un bot ?

Les littératures scientifiques sur les IA et plus récemment sur les chatbots ont fait émerger différentes solutions dont nous parlerons dans ce mémoire.

Chapitre 2

Concept

2.1 Dialogue homme-machine

Le dialogue est un discours que l'on construit à deux, s'il y a plusieurs personnes, on parle alors de multilogue.

Il y a un langage oral avec un niveau très variable ; il peut être spontané, avec la présence d'hésitations, de corrections et de répétitions.

Sa construction se fait avec des tours de paroles, avec la possibilité de parler plus fort que l'autre, de couper l'autre en plein milieu d'une phrase ou encore de l'empêcher de parler.

Pour finir, il y a une importance sur les aspects paralinguistiques (accents, intonation et rythme) et extralinguistiques (gestes de la main, de la tête, du regard et de la posture) qui donnent des sens cachés dans une conversation.

Pour le dialogue avec une machine, on veut utiliser les mêmes caractéristiques du dialogue à la différence que celui-ci n'a d'intérêt que lorsqu'il est finalisé en exécutant une tâche, ce pourquoi il a été créé.

Les enjeux du dialogue homme-machine est de reproduire le dialogue conventionnel. Pour ça, il doit réussir à gérer les tours de paroles en conservant des informations données par l'utilisateur pour les réutiliser au bon moment, et plus il y a de tours et plus c'est compliqué.

Il doit comprendre le langage naturel dans toute sa complexité, ce qui comprend l'orthographe et la grammaire d'une langue, on reviendra sur ce point dans la partie natural language understanding. Dans un langage oral avec un aspect cognitif, il doit tenter de comprendre au maximum les aspects para et extralinguistiques.

Il y a plusieurs définitions du dialogue ; je reprends celles utilisées dans une thèse sur le dialogue homme-machine :

Définition 1 : Le dialogue est l'ensemble des échanges entre un utilisateur et une machine (ou un ensemble de logiciels). [7]

Le terme échange regroupe ici toutes les interactions possibles avec l'utilisateur, par dialogue écrit sur ordinateur, vocal, visuel et bien d'autres. Cette définition est une définition générale qui intègre les définitions 2 et 3.

Définition 2 : Le dialogue est l'ensemble des interactions produites par un humain sur un ordinateur. [7]

Une interaction est une action qui provoque une réaction perceptible. Elle regroupe des notions telles que les clics souris, la frappe de touches, les pressions sur un écran tactile, etc. Cette définition est la plus courante en matière d'IHM.

Définition 3 : Le dialogue est l'utilisation du langage naturel pour travailler sur un ordinateur. [7]

2.2 Natural Language processing

Nous allons maintenant voir comment comprendre le langage naturel de l'utilisateur par la machine. Pour cela, on a le natural language processing qui est l'approche informatique pour analyser du texte en se basant sur des théories et des connaissances.

Définition : Le natural language processing est la théorie des techniques informatiques pour analyser et représenter des textes naturels sur plusieurs niveaux d'analyse linguistique afin de réaliser le traitement du langage semblable à l'homme pour un ensemble de tâches ou d'applications.

2.2.1 But

Le but du NLP : il doit accomplir le traitement du langage semblable à l'homme. Le choix du mot « traitement » est très délibéré et ne devrait pas être remplacé par de « la compréhension ». Bien que le domaine du NLP ait été à l'origine mentionné comme le Language Natural Understanding (NLU) dans les premiers jours de l'intelligence artificielle, il est bien accepté aujourd'hui que le but n'a pas encore été accompli.

Un Système NLU complet devrait être capable de :

- Paraphraser un texte en entrée
- Traduire le texte dans une autre langue
- Répondre aux questions en fonction du texte
- Faire des déductions à partir du texte

Le but du système NLP est de représenter la vraie signification et l'intention de la question de l'utilisateur qui peut être exprimée naturellement dans le langage courant. Aussi, le contenu d'informations qui sont recherchées sera représenté à tous leurs niveaux de signification pour qu'un vrai matching entre le besoin et la réponse puisse être trouvé.

2.2.2 Principes

Tandis que le domaine entier est mentionné comme le NLP, il y a en fait deux parties distinctes : la compréhension du langage (NLU) et la génération de langage (NLG).

Le premier de ceux-ci se réfère à l'analyse de la langue pour produire une représentation significative, tandis que le dernier se réfère à faire une production de la représentation.

Le NLP est équivalent au rôle de lecteur/auditeur, tandis que la tâche NLG est celle de l'auteur/orateur. Alors qu'une grande partie de la théorie et de la technologie est partagée par ces deux divisions, la Génération de Langage naturel exige aussi une capacité de planification. C'est-à-dire que le système de génération exige un modèle sur le but de l'interaction pour décider ce que le système doit produire à chaque interaction.

2.3 Natural language understanding

La méthode pour présenter ce qui arrive en réalité dans un système de traitement automatique des langues se fait grâce à l'approche du « niveau des langues ». Ceci est aussi mentionné comme le modèle synchrone de langue qui établit l'hypothèse suivante : les niveaux de traitement du langage humain se suivent d'une façon strictement séquentielle. La recherche psycholinguistique suggère que le traitement du langage soit beaucoup plus dynamique de même que les niveaux peuvent interagir dans une variété d'ordres.

Par exemple, la connaissance pragmatique que le document que vous lisez est de la sociologie sera utilisé quand un mot particulier qui a plusieurs sens est rencontré et le mot sera interprété comme ayant du sens pour la sociologie.

La description des niveaux de langage sera présentée séquentiellement (figure 2.1)

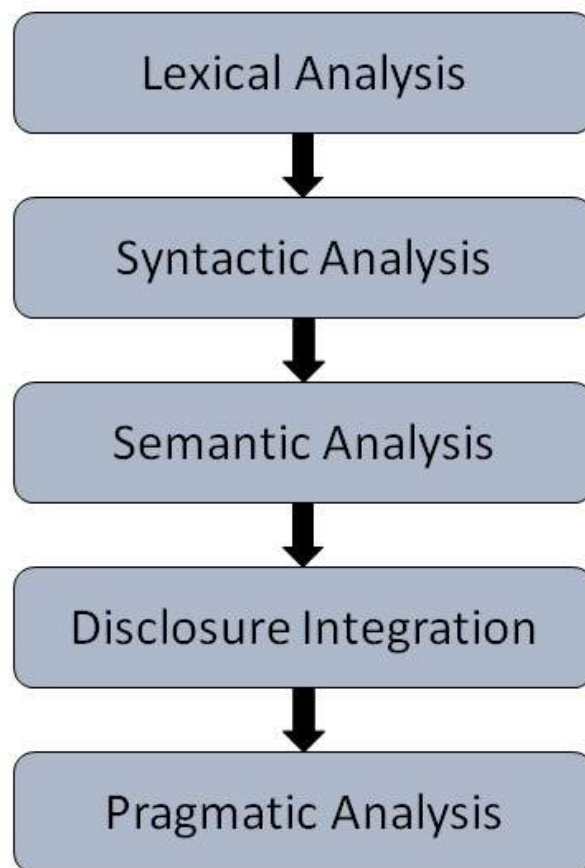


FIGURE 2.1 – Cycle d’analyse d’une phrase

Le point clé consiste ici en ce que la signification soit transmise par chaque niveau de langue et que l’on gagne en compréhension.

Ce processus implique trois problèmes majeurs : le premier se rapporte à la façon de penser, le deuxième à la représentation et à la signification linguistique et le troisième au domaine de connaissance.

Ainsi, un système NLP peut commencer au niveau d’un mot pour déterminer la structure morphologique, la nature ... et peut ensuite avancer au niveau de la phrase pour déterminer l’ordre des mots, grammaire, signification de la phrase entière. Après, il y a le contexte et l’environnement global. Un mot donné ou une phrase peuvent avoir une signification spécifique ou une connotation dans un contexte donné et peuvent être liés à beaucoup d’autres mots et phrases dans le contexte donné.

2.3.1 Phonologie

Ce niveau traite l’interprétation des sons vocaux à travers les mots. En fait, il y a trois types de règles utilisées dans l’analyse phonologique :

1. règles phonétiques, pour les sons dans les mots

2. des règles phonémiques, pour les variations de voix
3. règles prosodiques, pour la fluctuation causée par le stress, les accents et l'intonation à travers une phrase

Dans un système NLP qui accepte un dialogue vocal, les ondes sonores sont analysées et codées dans un signal numérisé pour l'interprétation par des règles diverses ou par la comparaison au modèle de langue particulier étant utilisé.

2.3.2 Morphologie

Ce niveau traite la nature componentielle de mots qui sont composés de morphèmes, les unités les plus petites de signification. C'est une décomposition du sens d'un mot en unités de sens élémentaires. Par exemple, le mot *reconstruction* peut être morphologiquement analysé dans trois morphèmes séparés : le préfixe *re-*, la racine *construc-* et le suffixe *-tion*. Les humains peuvent décomposer un mot inconnu dans ses morphèmes constitutifs pour comprendre sa signification. De même un système NLP peut reconnaître la signification transmise par chaque morphème pour gagner et représenter la signification.

2.3.3 Lexical

L'analyse lexicale consiste à décomposer une chaîne de caractères en entités lexicales et ranger ces entités dans des catégories ; on appelle ce processus la segmentation.

Il existe deux modèles :

- Lexicaux qui est une liste de mots de la langue ; par conséquent, les mots inconnus ne sont pas reconnus. . .
- Statistiques qui est la probabilité des successions de mots.

À ce niveau, les humains aussi bien que les systèmes NLP interprètent la signification de mots individuels. De plus, au niveau lexical, ces mots qui ont plusieurs sens possibles peuvent être remplacés par une représentation sémantique. La nature de la représentation varie selon la théorie sémantique utilisée dans le système NLP.

Le niveau lexical peut exiger un lexique et l'approche particulière prise par un système de NLP qui déterminera quel lexique sera utilisé. Les lexiques peuvent être tout à fait simples, avec seulement les mots, ou peuvent être de plus en plus complexes et contenir des informations sur la classe sémantique du mot.

2.3.4 Syntaxique

L'analyse syntaxique étudie la façon dont les mots se combinent pour former des phrases dans une langue donnée.

Ce niveau d'analyse des mots permet de découvrir la structure grammaticale de la phrase. Ceci exige tant la grammaire qu'un analyseur syntaxique. L'analyse de ce niveau est une représentation de la phrase qui révèle les relations de dépendance structurelles

entre les mots. Il y a diverses grammaires qui peuvent être utilisées et qui auront un impact sur le choix de l'analyseur syntaxique. La syntaxe transmet la signification dans la plupart des langues parce que l'ordre et la dépendance contribuent à la signification.

Par exemple les deux phrases : « le chien a poursuivi le chat. » et « le chat a poursuivi le chien. » diffèrent seulement en termes de syntaxe, et pourtant elles transmettent des significations tout à fait différentes.

2.3.5 Sémantique

L'analyse sémantique établit la signification d'une phrase en utilisant le sens de chaque mot par des formes logiques et/ou des formalismes gérant des formes logiques.

Il existe 3 modèles impliqués :

- Modèles fondés sur les référents, Théorie des Représentations Mentales (Reboul 1998) et son implémentation (Grisvard 2000) ; Représentations Référentielles (Popescu-Belis 2000) ; Domaines de Référence (Salmon-Alt 2001, Landragin 2003)
- Modèles logiques, DRT (Kamp et Reyle 1993) ; UDRT (Reyle 1993) ; SDRT (Asher 1993) ; MDRT (Pineda et Garza 2000)
- Modèles statistiques, comme pour l'analyse lexical, on a la probabilité de succession de mot pour trouver un sens.

Ce niveau de traitement peut inclure la désambiguïsation sémantique de mots avec des sens multiples.

Par exemple, le mot « vienne » peut prendre plusieurs sens différents dans des phrases comme « qu'il vienne par là » (verbe « venir ») et dans d'autres comme « aller à Vienne » (nom propre).

Du point de vue sémantique, le même mot « vienne », peut correspondre à la ville de « Vienne » qui existe en Autriche et en France, ainsi qu'au département de la Vienne. Il y a une multitude de mots ambigus dans toutes les langues, ce qui peut créer des incompréhensions.

Mais dans la perspective de l'analyse sémantique, en situation de communication réelle, on saura faire la différence dans un contexte entre les mots « Vienne » et « vienne ». Outre le V majuscule, la logique grammaticale permet de comprendre que « vienne » est un verbe, tandis que « Vienne » est un nom.

2.3.6 Discours

Tandis que la syntaxe et la sémantique travaillent sur l'analyse d'une phrase, le niveau de discours dans le NLP se concentre sur l'interprétation de l'enchaînement de phrases, dont chacune peut être interprétée séparément. Le discours se focalise sur les propriétés du texte qui se transmettent entre des phrases pour former des connexions. Plusieurs types de traitements de discours peuvent arriver à ce niveau, deux des plus communs étant la résolution d'anaphore et la reconnaissance de structures de discours/texte.

La résolution d'anaphore est le remplacement de mots comme des pronoms, avec l'entité appropriée à laquelle ils se réfèrent. Par exemple, « Ma soeur a adopté un chat. Il n'est pas très beau ».

La reconnaissance de structures de discours/texte détermine les fonctions des phrases dans le texte, comme par exemple thèse, antithèse et synthèse .

2.3.7 Pragmatique

L'analyse pragmatique identifie et traite les actes de langage que l'on catégorise selon leur but : citer, informer, conclure, donner un exemple, décréter, déplorer, objecter, réfuter, concéder, conseiller, distinguer, émouvoir, exagérer, ironiser, minimiser, railler, rassurer et rectifier. Mais elle traite aussi le second degré : présuppositions, implications, sous-entendus, sens littéral et sens argumentatif.

Ce niveau est concerné par l'utilisation de la langue en prenant en compte le contexte et aussi le contenu pour comprendre quel est le but et expliquer les significations non visibles dans le texte. Ceci exige beaucoup de connaissances, y compris la compréhension des intentions, des plans et des buts du dialogue.

2.3.8 Cognitive

C'est un élément récent dans le dialogue homme-machine où l'on souhaite analyser les émotions de l'utilisateur. Cette analyse se fait par vidéo en reconnaissant des patterns du visages ou reconnaissance dans une phrase avec un modèle.

2.3.9 Base de connaissance

Les réponses dans un dialogue peuvent être générées à partir d'une base de connaissances pour répondre à l'utilisateur.

2.4 Natural language Generation

Après la phase d'analyse du natural language understanding, on a le natural language generation. Elle se base sur les résultats des analyses linguistiques (détermination du contenu sémantique de l'énoncé, résolution des références aux objets, identification des actes de langage) pour faire des hypothèses sur les intentions de l'utilisateur.

Ensuite, on a un raisonnement sur ces hypothèses et sur l'intérêt de l'énoncé. L'énoncé courant est mis en rapport avec les énoncés précédents, c'est-à-dire qu'il faut gérer l'historique du dialogue pour trouver les réponses et les réactions pertinentes.

Il existe 3 modèles impliqués :

- Modèle de l'utilisateur avec une liste de préférences linguistiques, pragmatiques et des connaissances supposées.

- Historique du dialogue ou modèle de représentation des contenus échangés, pile des énoncés précédents et de leurs interprétations.
- Modèle de raisonnement sur les contenus, déductions logiques (ou pas).

Voici (figure 2.2) une représentation des phases énoncées précédemment :

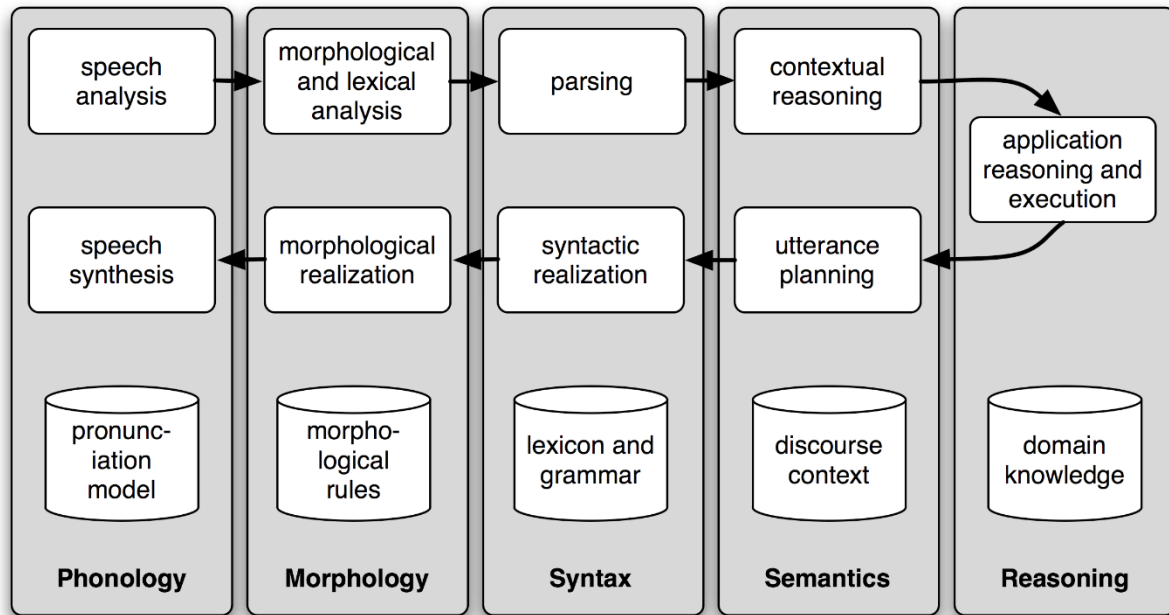


FIGURE 2.2 – Analyse NLP

2.5 Dialogue chatbot

On distingue principalement trois composants dans l'architecture d'un chatbot :

- L'interface utilisateur à travers laquelle les utilisateurs peuvent interagir avec le bot.
- Un moteur qui traite les messages des utilisateurs et qui fonctionne avec une base de données pour stocker les données des utilisateurs et faire appel à des services externes.
- Un moteur de Traitement Automatique du Langage Naturel qui transforme les entrées des utilisateurs en actions exécutables.

Cette structure ressemble à une architecture classique avec d'un côté un front-end et de l'autre un back-end qui expose une API. La grande différence est la présence du moteur du traitement du langage naturel.

Voici l'architecture d'un chatbot :

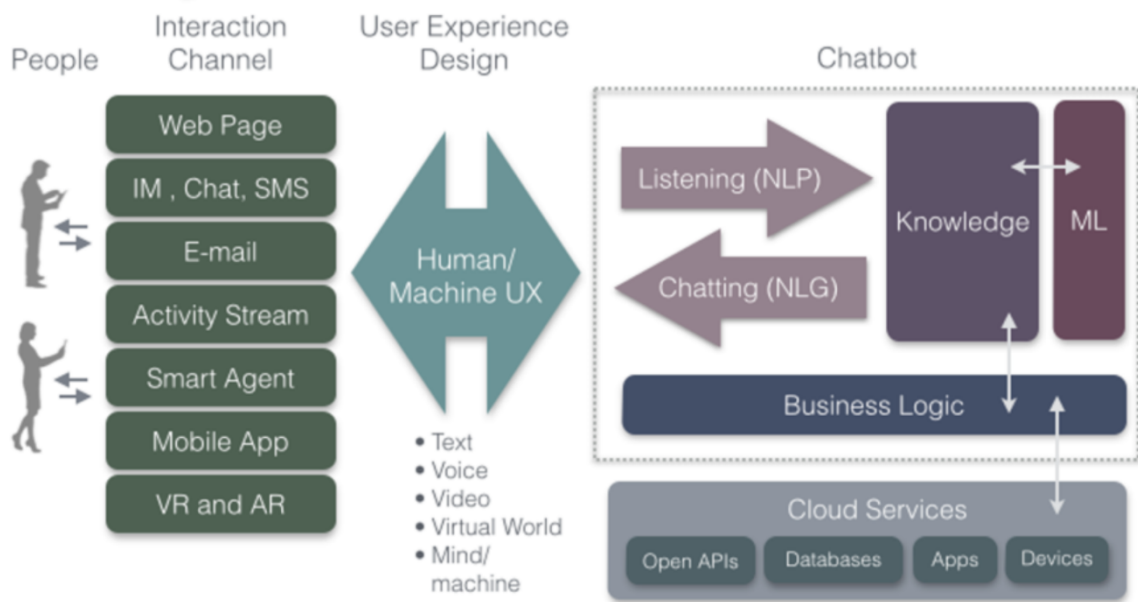


FIGURE 2.3 – Architecture d'un chatbot

On a la partie avec les différents canaux où les utilisateurs communiquent avec le chatbot sous différents formats tels que du texte, par la voix ou encore l'environnement.

Dans le moteur de langue évoqué plus haut, on retrouve ici le NLP qui grâce aux connaissances et aux machine learning permettra au bot de comprendre le message de l'utilisateur. Ensuite il utilisera des règles en s'appuyant sur des services cloud pour formuler une réponse qui sera le NLG.

Chapitre 3

Etat de l'art

Comme nous l'avons vu, comprendre le langage naturel de l'utilisateur nécessite le système NLP décomposé en NLU pour comprendre et NLG pour répondre.

Mais, comment peut-on programmer ce dialogue ? Il existe plusieurs techniques. Nous allons en aborder trois qui sont liées aux dialogues homme-machine. La première fera référence à de l'algorithme sur le TAL (Traitement automatique des langues) en python. La deuxième à L'Artificial Intelligence Markup Language(AIML) qui est un langage hérité de XML et utilisé pour les tout premiers chatbots. Et le troisième, le dialogue géré avec des intents qui commence à émerger notamment utilisé pour Cortana.

Dans un second temps, nous regarderons ce qu'apporte chaque solution. Puis, nous évaluerons leurs impacts dans le paramétrage d'un chatbot.

3.1 Etudes et expérimentation sur le TAL via NLTK

3.1.1 TAL

Le traitement automatique des langues est une discipline qui combine linguistique, informatique et formalisme. Le but de cette discipline est de développer des logiciels pour traiter le langage naturel par ordinateur de façon automatique.

L'objectif est de donner aux ordinateurs les capacités linguistiques d'un être humain. Elle existe depuis les années 50 comme le début de l'IA avec les traductions automatiques.

Pour les niveaux de traitements, il faut se référer aux chapitres concept sur le natural language understanding qui traite des différentes analyses du langage.

Dans un entretien avec un chercheur de Paris X, Marcel Cori explique les techniques mises en œuvre :

« C'est une recherche en linguistique et une recherche en informatique, puisqu'il s'agit de définir des modèles et des algorithmes sur ces modèles de données langagières. ». [8]

Pour parler du TAL et des algorithmes, nous allons nous intéresser du côté du langage python et d'une librairie qui existe pour appliquer le TAL, qui est le Python Natural Language ToolKit (NLTK).

3.1.2 NLTK

Le NLTK est un ensemble de modules python qui implémente des algorithmes pour le TAL. Il permet de faire un traitement des expressions régulières qui sera le cœur de cette méthode pour extraire les mots d'une phrase. Une fois les mots extraits, ils sont étiquetés en suivant des règles pour les catégoriser puis, les analyser en arbre syntaxique.

Segmentation

La segmentation du texte consiste à découper en "mots" ou "segments". Elle est obligatoire pour commencer toutes applications TAL. Elle détermine les unités de base à partir desquelles ces applications pourront travailler.

Dans la librairie on a la fonction qui prend en entrée une chaîne de caractères et une expression régulière et renvoie en sortie, la liste des chaînes qui matchent cette expression régulière.

Une expression régulière spécifie un pattern de caractères, ce pattern sert à identifier certains types de chaînes.

Etiquetage

Après segmentation on associe à chaque mot une étiquette pour l'identifier.

On a 8 grandes catégories : nom, verbe, pronom, préposition, adverbe, conjonction, adjectif et article.

Avec d'autres modèles comme Penn Treebank, on a 45 étiquettes ou encore Susanne avec 353 étiquettes.

Modèle Penn Treebank : [9]

CC	Coord Conjunctn	<i>and,but,or</i>	NN	Nom, sing. or mass	<i>dog</i>
CD	Cardinal number	<i>one,two</i>	NNS	Nom, plural	<i>dogs</i>
DT	Article	<i>the,some</i>	NNP	Proper nom, sing.	<i>Edinburgh</i>
EX	Existential there	<i>there</i>	NNPS	Proper nom, plural	<i>Orkneys</i>
FW	Foreign Word	<i>mon dieu</i>	PDT	Prearticle	<i>all, both</i>
IN	Préposition	<i>of,in,by</i>	POS	Possessive ending	<i>'s</i>
JJ	Adjectif	<i>big</i>	PP	Personal pronom	<i>I,you,she</i>
JJR	Adj., comparative	<i>bigger</i>	PP\$	Possessive pronom	<i>my,one's</i>
JJS	Adj., superlative	<i>biggest</i>	RB	Adverbe	<i>quickly</i>
LS	List item marker	<i>1,One</i>	RBR	Adverbe, comparative	<i>faster</i>
MD	Modal	<i>can,should</i>	RBS	Adverbe, superlative	<i>fastest</i>

FIGURE 3.1 – Modèle1 Penn Treebank

RP	Particle	<i>up,off</i>	WP\$	Possessive-Wh	<i>whose</i>
SYM	Symbol	<i>+,%,&</i>	WRB	Wh-adverbe	<i>how,where</i>
TO	"to"	<i>to</i>	\$	Dollar sign	<i>\$</i>
UH	Interjection	<i>oh, oops</i>	#	Pound sign	<i>#</i>
VB	verbe, base form	<i>eat</i>	"	Left quote	<i>' , "</i>
VBD	verbe, past tense	<i>ate</i>	"	Right quote	<i>' , "</i>
VBG	verbe, gerund	<i>eating</i>	(Left paren	<i>(</i>
VCN	verbe, past part	<i>eaten</i>)	Right paren	<i>)</i>
VBP	Verbe, non-3sg, pres	<i>eat</i>	,	Comma	<i>,</i>
VBZ	Verbe, 3sg, pres	<i>eats</i>	.	Sent-final punct	<i>. ! ?</i>
WDT	Wh-article	<i>which,that</i>	:	Mid-sent punct.	<i>: ; — ...</i>
WP	Wh-pronom	<i>what,who</i>			

FIGURE 3.2 – Modèle2 Penn Treebank

Analyse

Après l'étiquetage, on fait une analyse syntaxique ce qui donne, un arbre en sortie après décomposition de la phrase.

On peut retrouver un exemple (figure 3.3)qui explique le TAL [10] avec une phrase « Jean a mangé des pommes », le résultat est le suivant de l'analyse est le suivant :

U1 : Jean U2 : a mangé U3 : des U4 : pommes

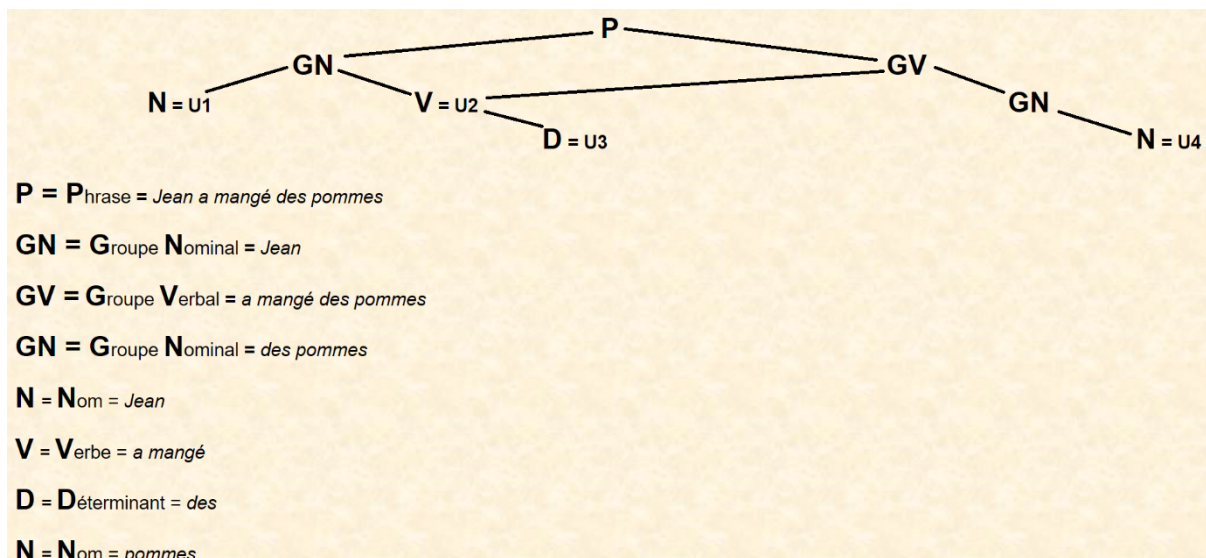


FIGURE 3.3 – Analyse syntaxique

3.2 Etudes et expérimentation sur l'AIML

Le robot baptisé A.L.I.C.E créé en 1995 utilise l'AIML qui se sert des fichiers dont le contenu est conforme aux normes XML. Tout fichier AIML donne une liste de schémas auxquels une phrase peut correspondre et celle-ci propose une autre liste de réponses possibles à cette dernière.

Exemple simplifié de fichier AIML en français [11] :

```
<pattern>Raconte-moi une blague </pattern> <template>Une poule rencontre une
autre poule. - Tu viens, lui dit-elle, on va prendre un ver ? </template>
```

Ainsi, si l'utilisateur demande précisément « Raconte-moi une blague » au chatbot, la réponse renvoyée sera « Une poule rencontre une autre poule. - Tu viens, lui dit-elle, on va prendre un ver ? Il est possible d'ajouter plusieurs réponses (template) pour un même schéma (pattern).

3.2.1 Catégories

Alice était ainsi le premier chatbot à coupler l'AIML avec une base connaissance. En effet, après la phrase de l'utilisateur, le système matchait un pattern avec la base connaissance pour ensuite fournir une réponse. [12]

Chaque catégorie est une partie de la base de connaissance que contient le chatbot.

Exemple simplifié de fichier AIML en français avec plusieurs catégories [11] :

```
<category><pattern>COMMENT CA VA ?</pattern><template>bien et vous ? </tem-
plate></category>
```



```
<category><pattern>QUEL EST TON NOM?</pattern><template>je m'appelle
Alice</template></category>
```

3.2.2 Récursivité

AIML implémente la récursivité avec la balise `<srai>` ce qui permet d'interpréter plus d'entrées de la part de l'utilisateur, nous allons voir quelques exemples pour mieux comprendre.

Réduction symbolique

La réduction symbolique se réfère au processus de simplifier des formes grammaticales complexes en formes plus simples. Par exemple, nous avons tendance à préférer des modèles comme "QUI EST SOCRATES" aux comme "VOUS SAVEZ QUI SOCRATES EST" en stockant des informations biographiques sur Socrates.

```
<category><pattern>WHO IS ALAN TURING?</pattern><template>Alan Tu-
ring was a British mathematician, cryptographer, and computer scientist often credited
as the founder of modern Computer Science. </template></category>
```

```
<category><pattern>WHO IS ALBERT SABIN?</pattern><template>Albert Sa-
bin was the researcher who developedthe vaccine that is the main defense against polio.
</template></category>
```

```
<category><pattern>DO YOU KNOW WHO * IS?</pattern><template><srai>
WHO IS <star/></srai></template></category>
```

L'expression étoile '*' sera reprise dans la balise star qui fera ensuite référence à Alan Turing ou Albert Sabin que l'on peut représenter (figure 3.4) dans les étapes E et F.

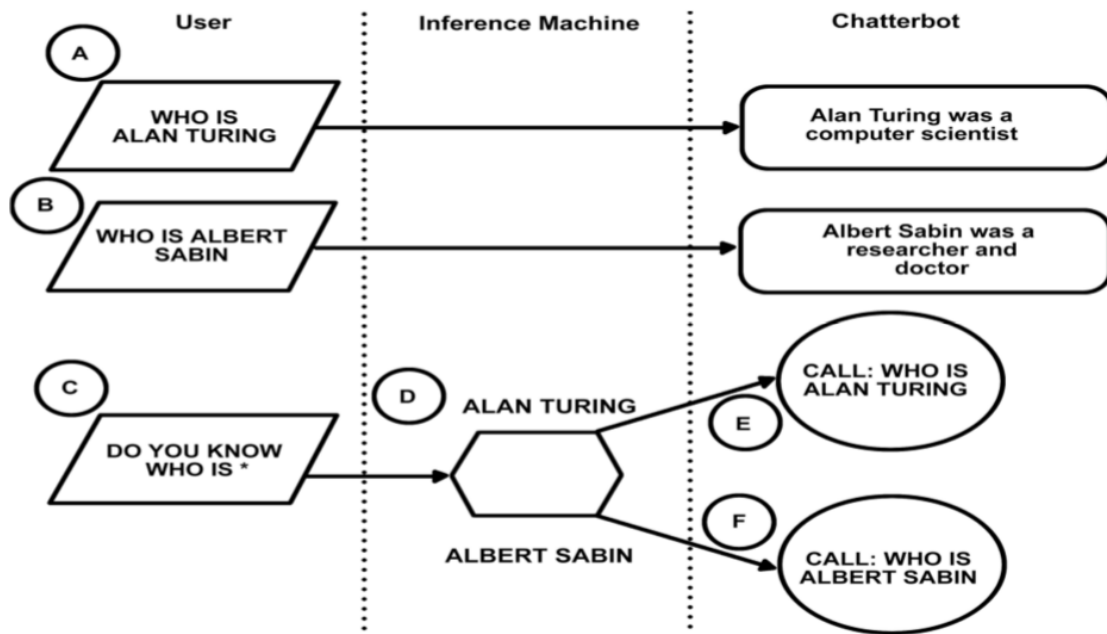


FIGURE 3.4 – Réduction symbolique AIML

Diviser pour conquérir

Une phrase peut être sous-divisée en plusieurs sous-phrases. Par exemple, on peut dire bye ou bien bye suivi de ce que l'on veut.

En prenant cet exemple, on a le fichier ci-dessous :

```
<category><pattern> BYE </pattern><template> Goodbye friend! </template></category>
```

```
<category><pattern> BYE * </pattern><template><srai> BYE </srai></template></category>
```

Dans la deuxième catégorie, si on a bye avec un ou plusieurs mots qui le précèdent alors on appellera le pattern BYE (figure 3.5).

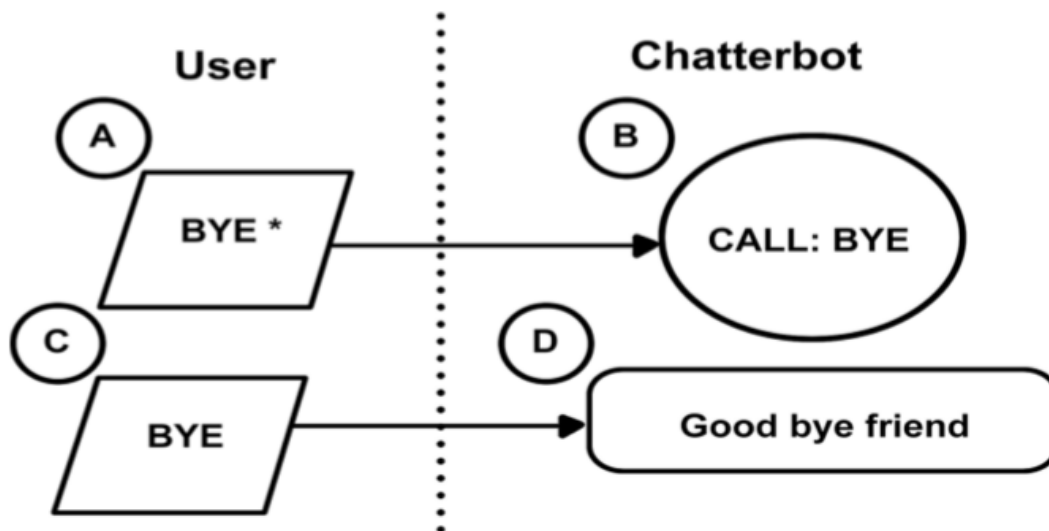


FIGURE 3.5 – Diviser pour conquérir AIML

Synonyme

En AIML, on ne peut avoir qu'un pattern par catégorie ; du coup, on met en place des synonymes qui font référence à un mot.

Exemple d'un fichier avec des synonymes :

```
<category>
<pattern>HELLO</pattern>
< template>Hi there!</template>
< /category>
<category>
< pattern>HI</pattern>
< template><srai>HELLO</srai></template>
< /category>
<category> < pattern>HI THERE</pattern>
< template><srai>HELLO</srai></template>
< /category>
```

Il existe encore d'autres balises et principes liés à l'AIML mais l'essentiel a été présenté, le principe étant basé sur des patterns de reconnaissance du message de l'utilisateur.

Dans une explication sur le principe de l'AIML, il est considéré que lorsque l'on prend la grammaire et la sémantique, le nombre de choses qu'une personne peut finalement dire n'est pas si grande.

Dans un livre de Steven Pinker, il dit la chose suivante : "Say you have ten choices for the first word to begin a sentence, ten choices for the second word (yielding 100 two-word beginnings), ten choices for the third word (yielding a thousand three-word beginnings),

and so on. (Ten is in fact the approximate geometric mean of the number of word choices available at each point in assembling a grammatical and sensible sentence). [13]

3.3 Etudes et expérimentation sur les intents

Les derniers chatbots utilisent le modèle sur les intents qui identifie l'intention d'un utilisateur selon son énoncé pour déclencher une action possible en prenant des entités ou non comme paramètres.

Nous allons expliquer les termes importants puis expliquer avec un exemple issu d'un ouvrage.

Les Intents (intentions) : il s'agit de la fonction que l'application exécute quand l'utilisateur dit quelque chose. Il existe différentes manières de demander la même chose, une intent peut être déclenchée par différents énoncés. Par exemple, "Je veux écouter une chanson de Rihanna" ou "Écouter Rihanna " devrait lancer une musique de l'artiste Rihanna.

Les Entités : elles représentent un concept. Dans l'exemple précédent sur la musique, Rihanna est une entité qui aurait pu être n'importe quelle autre artiste avec toujours la même fonction, celle d'écouter une musique.

Les Utterances (énoncés) : ce seront tous les énoncés type liés à un intent. Plus on gère de cas possibles et plus le chatbot est susceptible de comprendre à quel intent l'utterance fait référence, on peut alors parler de machine learning.

3.3.1 Exemple d'intent

Dans l'ouvrage [14], l'explication sur le modèle d'intent se base sur Cortana qui utilise une centaine d'intents avec 25M d'utterances. Dans la partie du livre où il existe ce modèle, il prend pour exemple la phrase suivante « Am I free today at noon ? » (Suis-je libre aujourd'hui à midi ?) en demandant à un chatbot de regarder son calendrier.

Dans cette question qui est une uttérance de type : « Am I entityat entity ? » le système va faire référence à la fonction read From Calendar avec les entités free today et noon ce qui donne le schéma suivant :

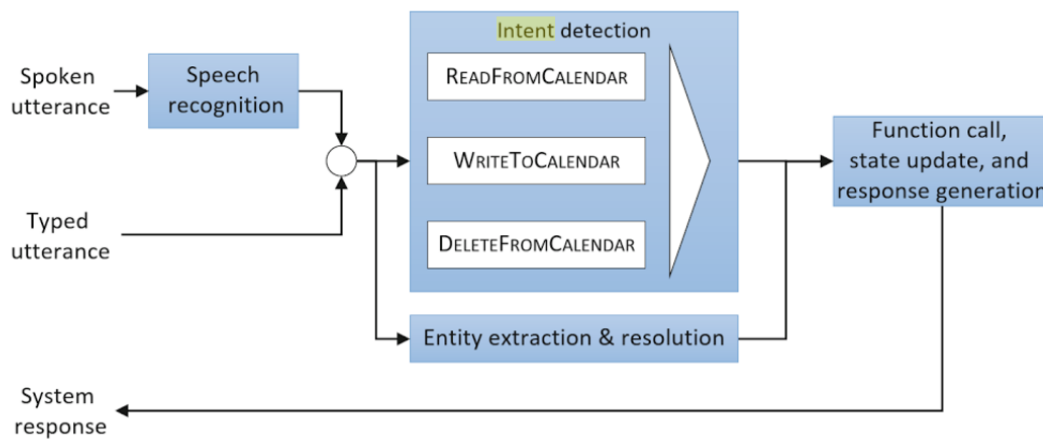


FIGURE 3.6 – Système d’appel d’intent

3.3.2 Modèle

Le modèle d’intent est expliqué par le nombre d’utterances permettant de prédéfinir un intent. Un modèle mathématique a été réalisé pour formaliser ce principe. Il a été modélisé par Wang YY, Deng L et Acero A [15] et Tur G, Mori RD[15] : “Intent i is a model of the form $P_i(y|x)$, where x are the words in the utterance and y is a binary variable where $y = 1$ indicates that the intent is present in the utterance and $y = 0$ indicates not.

For a given utterance x a set of intents P , the most likely intent i^* can be selected as :” [12] page 3-4

$$i^* = \arg \max_{i \in \mathcal{I}} P_i(y = 1|x)$$

FIGURE 3.7 – Modèle mathématique des intents

Ainsi, on atteint la valeur maximale de i si l’intent est présent dans l’utterance et qu’il y a le moins de mots possibles avec un modèle.

3.3.3 Utilisation du modèle

Pour comprendre comment des technologies utilisent le modèle d’intent, nous prendrons comme référent la start-up française recast.ai [17] qui explique très bien et facilement la manière de créer un chatbot en suivant ce modèle. Il faut savoir que les principes qui seront évoqués sont repris par d’autres sociétés et que ce sera donc à titre d’exemple.

Il y a quatre étapes pour créer un chatbot : l’apprentissage, la construction de la logique, l’exécution/déploiement et l’entraînement.

Apprendre

A la création d'un chatbot évidemment, il y a un nom à donner puis le ou les langues qui vont être comprises.

Ensuite, on commence à mettre en place des intents qui vont définir des actions réalisables par le chatbot. Les intents généralement en place sont greetings pour bonjour, hello, etc comme utterance et none lorsque le message ne correspond à aucun intent avec un message type : je ne comprends pas.

Quand un intent est créé, on lui rajoute des utterances, exemple ci-dessous d'un intent order avec une liste d'utterance :

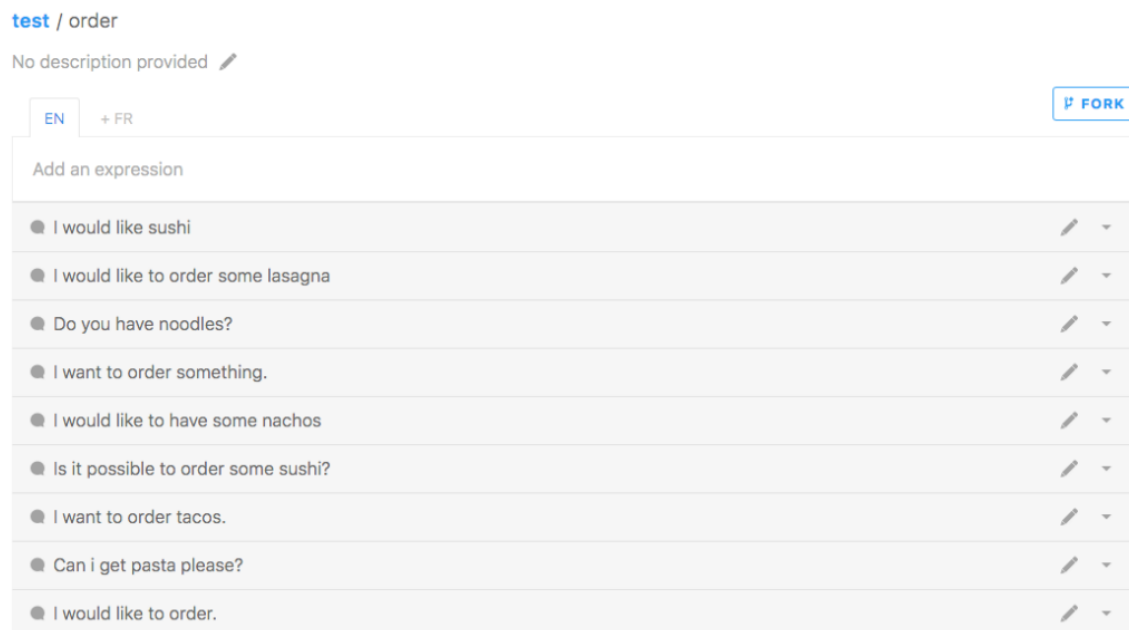


FIGURE 3.8 – Liste d'utterances pour l'intent order

Pour avoir plus de compréhension de la part de l'utilisateur, on peut mettre en place des entités comme exemple avec « I would like sushi » en mettant sushi dans une catégorie food.

La dernière étape est le test des utterances qu'on lui a apprises avec un tchat interne. En reprenant l'exemple d'une commande sushi, cela donne un JSON qui contient les informations sur le message. Voici le test :



FIGURE 3.9 – Test de l'intent order

En entrant I want to order some sushi, on a une correspondance à 93% avec l'intent order qui se rapproche le plus avec « I want to order something ».

Il reconnaît notamment sushi comme une entité food à 83%.

Construire

Quand tous les intents sont bien paramétrés, il reste à définir un enchaînement des actions pour arriver à une des tâches précises de l'utilisateur.

En prenant l'exemple du départ d'une commande, on peut se retrouver avec un processus semblable à :

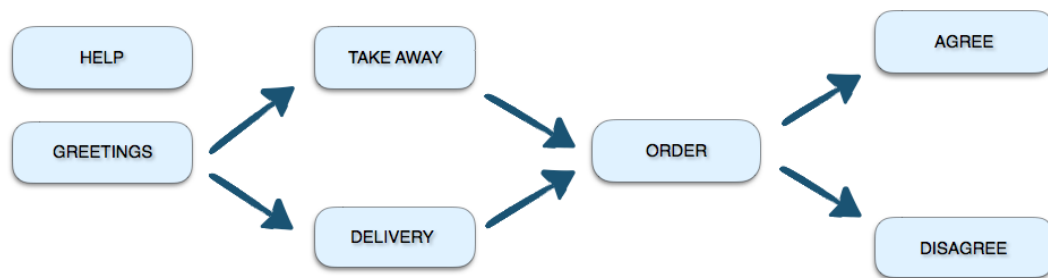


FIGURE 3.10 – Dialogue flow

Cette partie se fait avec un outil interne mais il peut bien évidemment se faire par la programmation en définissant l'ordre des exécutions et les questions du chatbot à chaque étape. On peut ensuite tester de nouveau le chatbot.

Exécuter

Comme vu dans la partie concept d'un chatbot, il faut mettre « l'outil à disposition » sur un ou plusieurs canaux ; pour ça, il existe plusieurs librairies qui permettent une connexion vers Messenger, SMS, MAIL, etc.

Entraîner

Après le déploiement sur un canal, on peut continuer à le faire évoluer en entrant d'avantage d'intent, d'utterance et de complexification avec des entités. On peut notamment récupérer ou voir les énoncés entrés par les utilisateurs pour perfectionner le chatbot.

3.4 Evaluations

Nous avons diverses méthodes de création d'un chatbot qui sont les plus répandues. Comme nous l'avons dit, le problème reste la compréhension réelle de l'utilisateur par le bot. Elle passe par deux gros problèmes : premièrement par les mots utilisés qui doivent être compris avec une bonne orthographe et deuxièmement par le sens d'une phrase.

Avec le nltk, l'orthographe peut poser problème pour l'analyse sémantique ; 90% de succès de l'analyse pour l'étiquetage quand il n'y a aucune faute donc moins s'il y en a. Pour le sens si l'étiquetage est réussi, alors l'arbre représenté sera bon et ce sera au programme ensuite de faire la compréhension du sens de la phrase même si cela ne sera pas toujours exact.

Avec l'aiml, l'orthographe pose beaucoup de problèmes car elle fait qu'on ne peut rentrer dans aucune catégorie, pour le sens l'aiml est fait pour orienter l'utilisateur autant dans les questions que dans les réponses donc il n'y a aucun problème.

Pour les intents, en écrivant des phrases type, on peut faire quelques erreurs mais se rapprocher le plus possible d'un intent et avec plus d'utterance on réduit les risques. Quant au sens, c'est mieux en proposant à l'utilisateur ces actions possibles et en l'aiguillant dans un domaine plus fermé que le nltk ou l'aiml avec des processus de dialogue.

Un chatbot est avant tout une conversation et ces méthodes la gèrent différemment. Le nltk ne fait que l'analyse de la phrase, il faut ensuite programmer une conversation, un échange, des réponses et une logique. L'aiml quant à lui ne fait que la conversation avec des entrées et des sorties pré-programmées avec de la logique vue précédemment. Pour finir, l'intent gère tout avec des outils simples et efficaces pour paramétrer un bot.

Un chatbot se veut intelligent dans ses réponses ; pour le nltk et l'intent tout dépend de l'algo après analyse ou matching avec un intent donc c'est compliqué à évaluer contrairement à l'aiml qui n'en a quasiment aucune. Le machine learning et la base de connaissances sont une part de l'intelligence et elles peuvent être utilisées dans ces méthodes mais le mémoire ne détaille pas ces sujets donc on n'en tiendra pas compte.

3.5 Etat de l'art des solutions

n prenant les critères évoqués dans la partie précédente, on peut facilement enlever l'aiml qui n'offre que très peu de possibilités en termes de programmation et de logique. Ensuite pour le nltk et les intents, on peut faire de l'algorithme dans les deux méthodes. On pourrait choisir le nltk, si on choisissait de faire un dialogue avec un robot physique, le python étant un langage utilisé pour les programmer.

L'intent semble être la meilleure solution car simple à programmer avec des outils comme celle de la start up recast.ai. Evidemment, il y en a d'autres comme IBM, Facebook, Microsoft ou encore Amazon qui s'y mettent beaucoup.

La solution des intents est nouvelle et innovante. Elle évolue beaucoup et colle parfaitement avec l'ère actuelle des différents canaux de discussion. Nous allons nous y intéresser davantage pour savoir si elle permet un paramétrage d'un chatbot intéressant tout en évaluant ses limites avec la partie compréhension du dialogue comme axe principal de notre problématique.

Dans ce dialogue, avec le NLP, la partie NLU est gérée par le moteur de l'outil sélectionné, on ne pourra que choisir les intents pour orienter, les utterances pour augmenter la compréhension et les entités pour plus de choix. La partie NLU sera programmée et on pourra ajouter une base de connaissances et du machine learning.

L'outil utilisé pour les intents sera LUIS (Language Understanding Intelligent Service) couplé avec Microsoft bot framework pour la partie programmation. Le choix de ces outils m'a été suggéré au cours de mon stage car il s'est déroulé dans une entreprise filiale de Microsoft et donc j'ai eu la chance d'avoir de la documentation, des présentations sur ce domaine et des gens qualifiés.

Chapitre 4

Implémentation d'un agent conversationnel

Le dernier chapitre nous a permis d'identifier diverses méthodes existantes. En fonction des besoins que nous avons mis en avant, nous avons pu retenir celle des intents. Cette méthode consiste à choisir le bon intent par rapport à la phrase d'un utilisateur pour fournir la meilleure réponse possible. L'intent permettra aussi à une application de déclencher un processus de dialogue pour répondre à une tâche possible du chatbot en utilisant tous les services externes disponibles pour ajouter de l'intelligence dans la réflexion.

4.1 Méthodologie

Pour étudier le paramétrage d'un chatbot avec la méthode choisie, nous utiliserons LUIS pour gérer le langage naturel couplé avec Microsoft bot framework pour programmer. Un scénario réel d'un chatbot effectuant un service sera développé et nous allons montrer quelles ont été les réflexions dans la phase de création par rapport à la compréhension du langage.

4.1.1 Microsoft bot framework

Microsoft bot framework est une architecture pour créer des chatbots, les connecter sur des canaux de discussions, les tester sur un émulateur, et les déployer sur les plateformes botframework pour les rendre accessibles et azure pour offrir davantage de services liés au machine learning ou à l'analytics.

Le développement se fera en C# sous Visual Studio 2015 et l'exécution du chatbot et les tests seront faits avec Bot Framework Channel Emulator.

4.1.2 LUIS

LUIS est un service de Microsoft Azure pour comprendre le langage naturel. Il gère les intents comme expliqué dans la partie état de l'art.

Diagramme de LUIS :

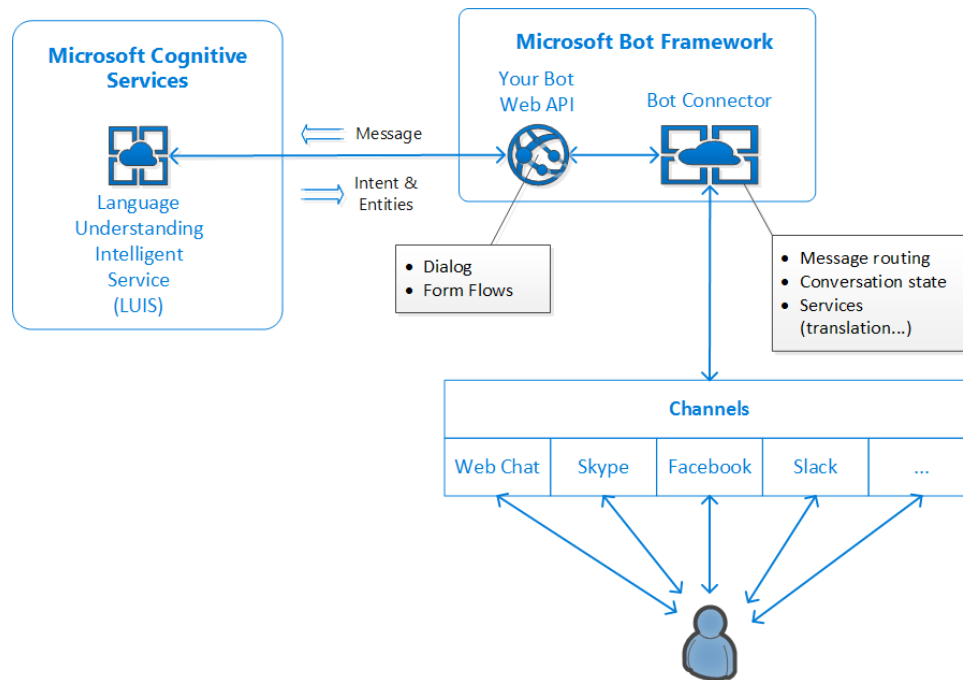


FIGURE 4.1 – LUIS architecture

Il y a deux autres services utiles, mais non utilisés dans ma solution qui seraient pourtant très utiles dans la phase de compréhension en amont de celle avant l'utilisation de LUIS :

- API vérification orthographique Bing : pour une correction orthographique avec un renvoi de meilleurs résultats.

Exemple : Je voudari commander une piza donne : je voudrais commander une pizza.
Le résultat en JSON :

```

{
  "_type": "SpellCheck",
  "FlaggedTokens": [
    {
      "Offset": 3,
      "Token": "voudari",
      "Type": "UnknownToken",
      "Suggestions": [
        {
          "suggestion": "voudrais",
          "Score": 0.939707755698091
        },
        {
          "suggestion": "voudrai",
          "Score": 0.824891719862266
        }
      ]
    }
  ],
  {
    "Offset": 25,
    "Token": "piza",
    "Type": "UnknownToken",
    "Suggestions": [
      {
        "suggestion": "pizza",
        "Score": 0.939707755698091
      }
    ]
  }
]
}

```

FIGURE 4.2 – Résultat JSON intent order

On peut voir qu'il vérifie voudari qui est mal orthographié et qui correspond plus à voudrais que voudrai. Et que piza est aussi mal orthographié car il manque un 'z'.

- API Analyse linguistique segmente une phrase en balise et sert à l'analyse morphosyntaxiques. Cette méthode a été vue dans la partie état de l'art sur le nltk. Pour l'exemple, je prendrai le même mais en anglais car en français il est pour le moment mal géré.

Exemple : I would like to order pizza :

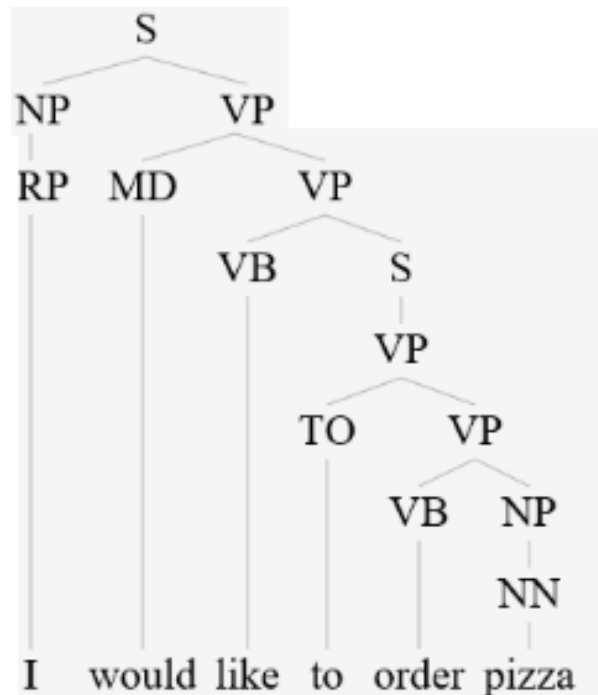


FIGURE 4.3 – Arbre Analyse syntaxique

S : sentence (phrase) NP : noun phrase (phrase nominal) VP : verbe phrase (phrase verbale) RP : particle (particule) MD : modal VB : verb (verbe) TO : to NN : noun (nom)

Le lien entre les différents outils repose sur des clés.

Pour notre chatbot programmé avec Microsoft bot framework, il faut l'enregistrer sur <https://dev.botframework.com/bots> pour obtenir une clé et un mot de passe qui seront ensuite à enregistrer dans le fichier config du projet. Ces informations seront demandées pour debug avec l'émulateur en localhost.

Bot framework channel emulator :

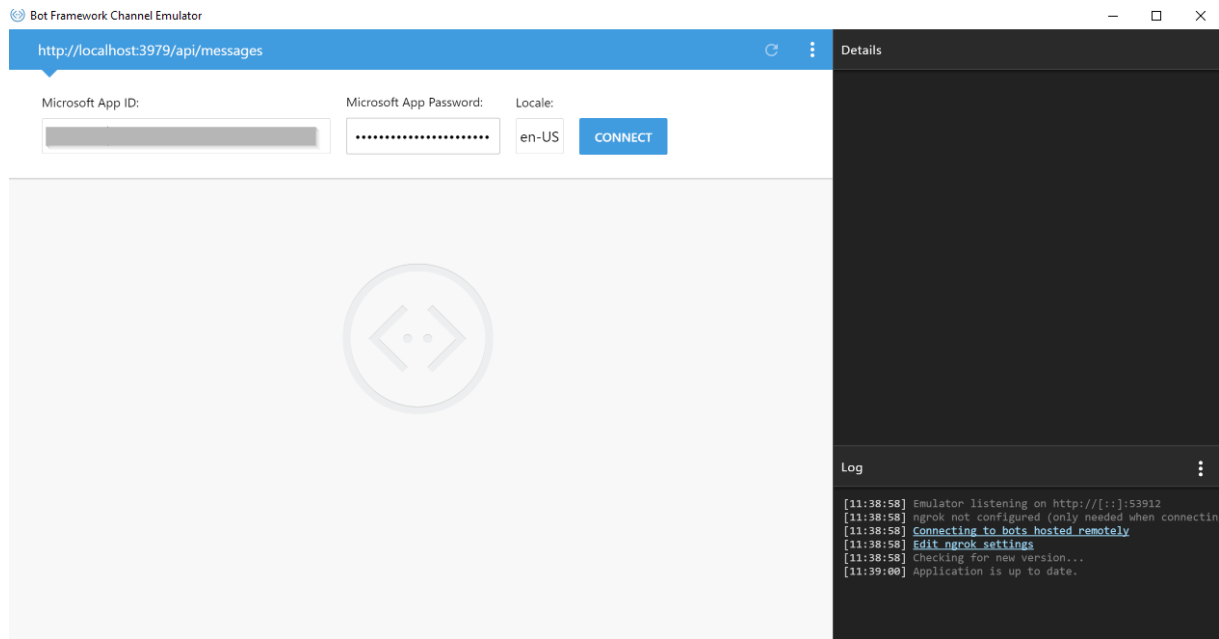


FIGURE 4.4 – Bot framework emulator

Pour LUIS, la clé est générée sur « <https://www.luis.ai/applications> », on construit son modèle et on l'utilise via la clé dans le programme. Ce point sera vu plus en détail dans la partie développement.

4.2 Cas concret : incident dans CRM

Les chatbots exécutent des tâches en répondant à des services ; lors de mon stage, un besoin était d'en implémenter un pour une entreprise d'assurance fictive nommée Oltiva pour des démonstrations clients avec une interaction entre Dynamics CRM et un chatbot.

Dynamics CRM (Customer Relationship Management) est un logiciel de gestion de la relation client pour gérer les ventes, le marketing et les services d'une entreprise. Il permet de suivre l'évolution d'un contact en futur client jusqu'à la vente.

Le scénario est le suivant :

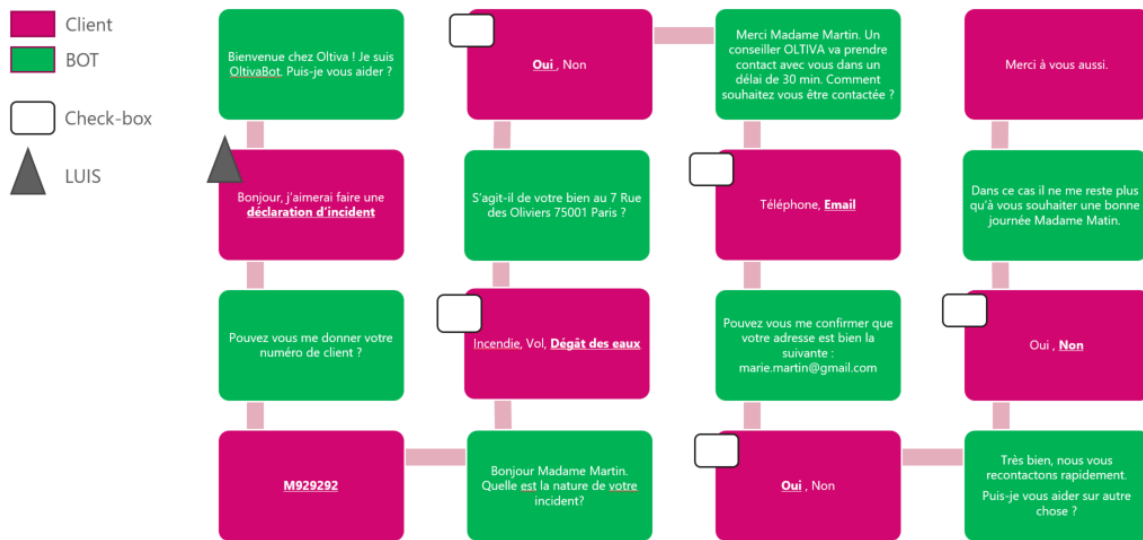


FIGURE 4.5 – Scénario incident CRM

L'utilisateur veut déclarer un incident, le chatbot demande son identifiant qui est un champ dans CRM pour récupérer toutes les informations du contact. Ensuite le chatbot propose les incidents d'habitation qu'il connaît : ce sont les incendies, les vols et les dégâts des eaux. Après avoir choisi, on lui demande si son adresse renseignée dans CRM est la bonne. Ensuite, il lui est proposé la façon dont il veut être contacté par un agent réel (mail ou téléphone) quand son incident sera pris en charge. Il confirme ces informations et s'il n'a pas d'autres questions, alors la conversation est finie.

4.3 Développement

4.3.1 Architecture

Mon architecture est la suivante :

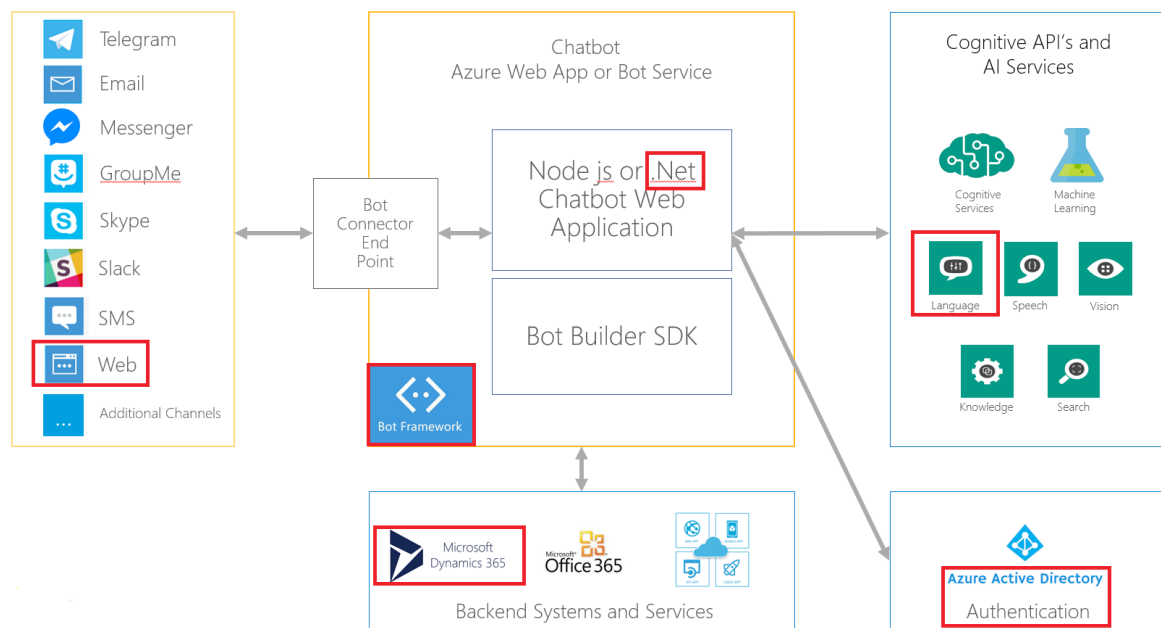


FIGURE 4.6 – Architecture solution chatbot

On y retrouve le web comme canal qui sera une iFrame présente dans le CRM d'Oltiva. Il sera paramétré sur le site bot framework du bot enregistré.

L'utilisation du Microsoft bot framework qui gère la connexion avec le chatbot est écrite en .NET. On a besoin du sdk, d'une clé et d'un mot de passe (fourni après enregistrement du bot sur le site).

On aura une connexion Dynamics CRM en utilisant le sdk CRM (XRM). Il faut le nom de domaine du CRM et un compte actif (administrateur) avec id et mdp qui seront encryptés.

Il y aura l'utilisation du service azure language (LUIS). On publie sur azure le chatbot pour être ensuite utilisable sur les canaux via Visual Studio. On enregistre notre bot sur LUIS pour obtenir notre modèle et l'utiliser.

Pour revenir au scénario, il est modifié en mettant en avant les interfaces avec CRM :

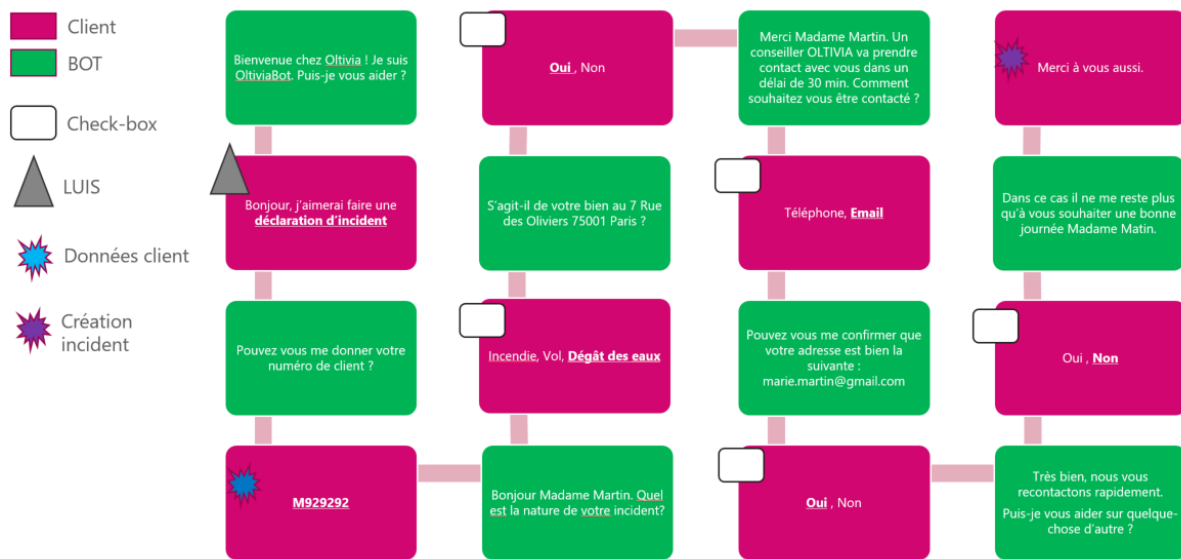


FIGURE 4.7 – Scénario incident CRM avec interfaces

LUIS n'intervient qu'au début pour entrer dans un dialogue bien précis avec un certain intent. On aura une connexion avec Dynamics au moment de l'entrée de son identifiant et on la fermera après la conversation pour créer un incident dans CRM avec les informations récupérées.

4.3.2 Paramétrage

Il faut avant tout les intents. On a bonjour, incident pour le scénario et none qui est obligatoire quand aucun autre ne correspond.

The screenshot shows the 'Intents' management page in CRMoltiva. The left sidebar contains navigation links: Settings, Dashboard, Intents (selected), Entities, Prebuilt domains, Features, Train & Test, and Publish App. The main content area displays a table of intents.

Intent Name	Utterances	
Bonjour	2	Edit Delete
Incident	5	Edit Delete
None	13	

FIGURE 4.8 – Intents chatbot CRM

On va s'intéresser sur intent avec les utterances suivantes :

Incident

Here you are in full control of this intent; you can manage its utterances, used entities and suggested utterances ... [Learn more](#)

Utterances (5) Entities in use Suggested utterances

Type a new utterance & press Enter ...		×
<div> <div>Save</div> <div>✕ Discard</div> <div>🗑 Delete</div> <div>Reassign Intent ▾</div> </div> <div>Labels view (Ctrl+E): Entities ▾</div> <div>Search in utterances ... 🔍</div> <div>⌵</div>		
<input type="checkbox"/> Utterance text	Predicted Intent	
<input type="checkbox"/> j ' aimerais faire une déclaration d ' incident	1 Incident	
<input type="checkbox"/> incident	1 Incident	
<input type="checkbox"/> déclarer un incident	1 Incident	
<input type="checkbox"/> ce serait pour faire une déclaration d ' incident	1 Incident	
<input type="checkbox"/> je voudrais déclarer un incident	1 Incident	

FIGURE 4.9 – Utterances intent incident chatbot CRM

On pourrait en rajouter davantage en allant dans l'onglet Suggested utterances qui sont les phrases rentrées par des utilisateurs du chatbot.

Dans le projet, on a un modèle MVC. Le modèle est l'utilisateur qui déclare son incident. Les vues sont les forms flows qui représentent les questions que pose le chatbot dans le scénario. Et le contrôleur gère la messagerie.

Les forms flows sont des dialogues générés automatiquement par le chatbot. L'avantage est d'orienter l'utilisateur tout en générant le ou les questions sur des foms flows appelés précédemment. A la fin, on peut appeler une méthode pour réaliser une action. Un exemple pour le premier, c'est de demander l'id utilisateur. Ensuite, on se connecte à CRM et on fait une requête sur l'id puis on instancie notre modèle.

Le contrôleur va appeler une classe qui contient le modèle LUIS pour correspondre avec un l'intent incident. Quand ce sera fait, on aura un enchaînement de méthodes pour représenter le scénario avec un appel successif des différents form flows. Voir Annexe A.

Les forms flows sont modulaires, ils ne dépendent en aucun cas des autres.

4.4 Résultat

Si on teste notre modèle sur la plateforme avec un intent qui n'existe pas mais en lien avec un incident « je voudrais faire une déclaration d'incident », on aurait :

Test your application

Use this tool to test the current and published versions of your application, to check if you are progressing on the right track ... [Learn more](#)

Train Application

Last train: May 3, 2017 5:10:19 PM | Last publish: May 3, 2017 5:11:52 PM

Interactive Testing Batch Testing

☐ Enable published model

Labels view (Ctrl+E) Entities [Reset console](#)

Type a test utterance & press Enter →

je voudrais faire une déclaration d ' incident

Current version results

Top scoring intent
Incident (1)

Other intents
None (0.12) Bonjour (0.02)

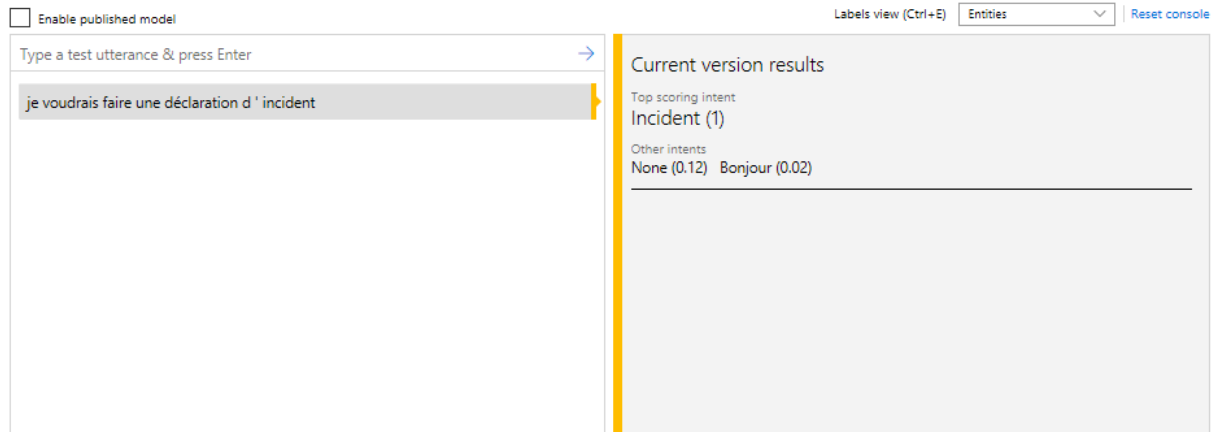
The screenshot shows the LUIS Interactive Testing interface. On the left, there's a text input field with the placeholder "Type a test utterance & press Enter" and a blue arrow button. Below it, the test utterance "je voudrais faire une déclaration d ' incident" is entered. On the right, the "Current version results" are displayed. The "Top scoring intent" is "Incident (1)". Under "Other intents", "None (0.12)" and "Bonjour (0.02)" are listed. The interface also includes a checkbox for "Enable published model", a "Labels view (Ctrl+E)" button, a dropdown menu for "Entities", and a "Reset console" link.

FIGURE 4.10 – Test intent incident bon

Cela correspond à une précision de 100% pour incident, 12% pour None et 2% pour Bonjour.

La même phrase avec une faute d'orthographe sur incident :

Type a test utterance & press Enter →

je voudrais déclarer un inciden

Current version results

Top scoring intent
None (0.63)

Other intents
Bonjour (0.03) Incident (0)

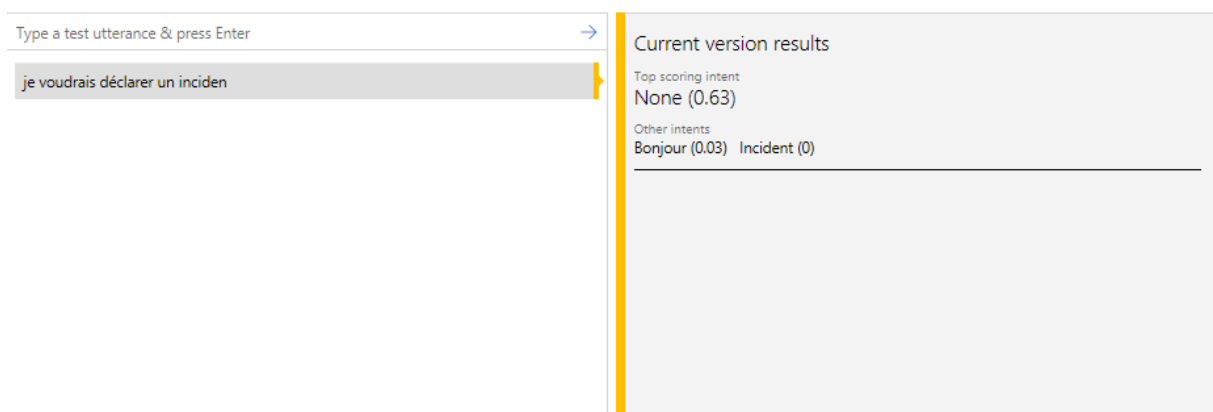
The screenshot shows the LUIS Interactive Testing interface with a misspelled test utterance. The text input field contains "je voudrais déclarer un inciden". The "Current version results" show that the "Top scoring intent" is "None (0.63)". Under "Other intents", "Bonjour (0.03)" and "Incident (0)" are listed. The interface elements are the same as in Figure 4.10.

FIGURE 4.11 – Test intent incident mauvais

Il correspond à None à 63%, Bonjour à 3% et Incident à 0%. Ce problème est résolu soit en entrant cette phrase dans les utterances de l'intent incident soit avec une correction orthographique dans le programme avant l'appel du modèle LUIS.

Le résultat du chatbot dans CRM donne ce résultat :

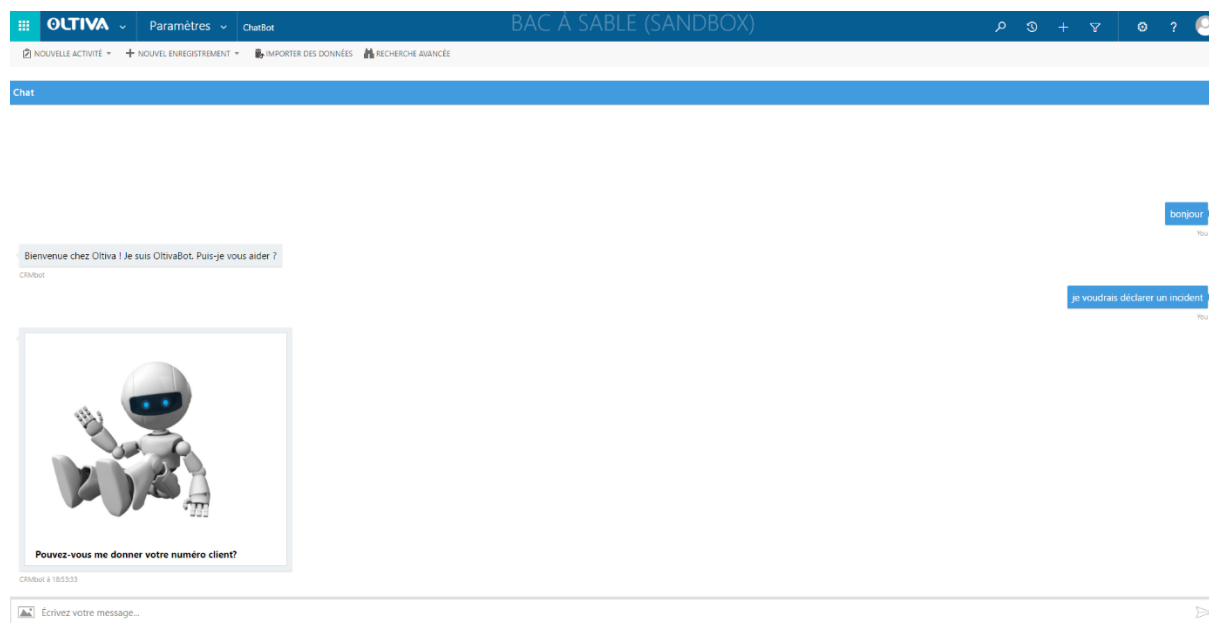


FIGURE 4.12 – Chatbot CRM

Vous pouvez retrouver la suite du dialogue dans l’annexe B.

4.5 Discussion

Notre solution ne comporte que deux intents donc il n’y a pas de gros problème pour commencer une conversation. Avec la compréhension du langage naturel, on aurait pu ajouter des outils d’analyse en place comme la correction orthographique et l’analyse sémantique. Ce choix a été fait car étant une démonstration client, on a peu de risque d’erreur possible lors de la manipulation. Ces outils restent une évolution possible et ils auraient été obligatoires si le chatbot avait davantage de tâches possibles liées aux incidents par exemple.

Le chatbot est fonctionnel avec un CRM mais on peut imaginer des connexions avec n’importe quel autre service et l’ utiliser sur la plateforme que l’on veut. Il remplit la tâche d’ajouter un incident en remplissant les informations de l’utilisateur pour ensuite être repris par un agent réel. Dans ce cas, le chatbot fait gagner du temps à une entreprise avec un service clientèle tout en étant simple d’utilisation.

Pour un chatbot en domaine fermé, la compréhension du langage nous a permis de paramétrer au mieux notre solution avec notre intent et les utterances qui étaient le NLU. On a fourni des forms flows pour plus de choix et de modularité. Le NLG utilise les infos de CRM qui constituent une base de connaissances pour offrir des dialogues dynamiques. On peut faire du machine learning en ajoutant différents utterances en s’inspirant de que rentrent les utilisateurs.

4.6 Limite de la solution

LUIS n'est utilisé qu'en imput de la solution ; lorsque l'on rentre dans le dialogue avec les forms flows, c'est à nous de contrôler ensuite les réponses de l'utilisateur étape par étape.

Comme on l'a dit, on est dans un domaine fermé, ce qui limite les problèmes liés au langage naturel car le dialogue est très orienté. S'orienter vers un domaine plus ouvert accroît les risques d'incompréhension avec notre solution actuelle car la complexité est nettement plus grande.

Les intents et les outils externes pour le compléter permettent d'améliorer les performances de compréhension. Le langage naturel est compris mis à part le discours et le sens des phrases qui sont plus dans des chatbots purement conversationnels sans but réel.

Il n'y a pas vraiment de NLG dans la solution car l'utilisateur souhaite déclarer un incident ; on entre dans l'intent et après il y a un enchaînement de questions que le chatbot pose. Les seuls retours qui sont faits dans ce dialogue ont lieu lorsque les informations ne sont pas correctes, alors il lui est demandé de les saisir.

Chapitre 5

Bilan

5.1 Conclusion

La compréhension de l'automatisation du langage théorique existe par le NLP découpé en deux parties : le NLU qui utilise le TAL pour toute la partie analyse des phrases et du dialogue ainsi que le NLG pour la partie raisonnement après analyse.

Cette compréhension a été utile dans la phase de conception du scénario pour obtenir un dialogue simple et efficace. Pour la partie paramétrage, les intents simplifient le langage naturel avec un outil qui permet une meilleure correspondance avec ce qui est possible de faire et les utterances améliorent cette performance avec la phase de machine learning. Le NLU est géré mis à part le discours et le sens des phrases qui eux sont à gérer avec des algorithmes car il n'existe pas d'outils pour les traiter. Quant au NLG, une base de connaissances, le CRM et le machine Learning avec la plateforme ont suffi.

Cette solution est encore en constante évolution notamment avec les différents langages, l'anglais est très bien maîtrisé alors que le reste beaucoup moins voire pas du tout dans les analyses syntaxiques par exemple.

Au vu de cette solution, la compréhension a permis la bonne conceptualisation, modélisation et le bon paramétrage d'un bot de type chatbot. Cette compréhension a été la plus utile pour nous en amont du paramétrage et sur le NLP.

Maintenant, si on veut parler d'une IA qui sera un domaine ouvert, la compréhension pour le paramétrage restera la même que sur cette solution pour le NLP tout du moins. Le NLG sera la partie principale d'une IA avec une réflexion primordiale où la compréhension du langage trouvera très vite une limite.

5.2 Evolution possibles

Durant ce mémoire, nous nous sommes focalisés sur un dialogue écrit mais bien évidemment celui oral est tout aussi important et utilise les mêmes principes à la différence qu'on a une partie de retranscription de ce qui est dit pour ensuite analyse. Cela aurait un impact supplémentaire sur le paramétrage.

On pourrait avoir une plus grande diversité des langues avec par exemple une phase où on analyserait le langage de l'utilisateur dès le début d'un dialogue même si cela reste compliqué vu les problèmes actuels pour passer de l'une à l'autre. La partie NLU serait encore plus complexe avec une compréhension du langage pour le paramétrage.

Il y a aussi un dialogue avec un bot réel où des interactions physiques pourraient avoir lieu avec une discussion vocale pour le NLU et NLG, par exemple pour déplacer un objet d'un point A à un point B de la meilleure des façons.

Une partie non évoquée, c'est le cognitif avec une détection des expressions d'une personne en plus de ce qu'il pourrait dire. On aurait une partie raisonnement plus poussée pour comprendre les sentiments de la personne et répondre plus intelligemment.

Pour rendre un bot encore plus intelligent, on pourrait utiliser un réseau neuronal artificiel qui est un ensemble d'algorithme pour formuler des hypothèses et faire un raisonnement avec des informations issues du deep learning pour un apprentissage automatique.

Toutes les évolutions possibles présentées pourraient être dans un même bot et on se rapprocherait d'un bot plus proche de l'homme.

Bibliographie

- [1] “Test de turing.” <https://sites.google.com/site/int3llig3nc3artifici3ll3/le-test-de-turing>.
- [2] S. Lucci and D. Kopec, *Artificial Intelligence in the 21st Century, Second Edition*.
- [3] A. Kearney, “Technology and innovation for the future of production : Accelerating value creation,” *world economic forum*, March 2017.
- [4] Avanade, *Deep dive chatbots*, 2017.
- [5] “Comprendre les chatbots.” <https://www.fabernovel.com/insights/tech/comprendre-les-chatbots>.
- [6] “Les chatbots sur messenger.” <http://www.numerama.com/business/162745-facebook-lance-linvasion-des-bots-dans-messenger.html>.
- [7] J. Claude Tarby, “Thèse gestion automatique du dialogue homme-machine à partir de spécifications conceptuelles,” *Université des Sciences Sociales - Toulouse*, 1993.
- [8] “Entretien avec marcel cori professeur de tal à l’université paris x.” <http://www.technolangua.net/imprimer.php3?idarticle=274>.
- [9] C. Gardent, “Traitement des langues naturelles (tal),” 2011.
- [10] M. Amine, “Glossaire de linguistique computationnelle,” 1995.
- [11] “Fichier aiml.” <http://alicebot.wikidot.com/aiml:fr-fr:tsiewlan:humor-ed-aiml>.
- [12] R. Wallace, *The Elements of AIML Style*,. A.I Foundation, 2001.
- [13] D. R. S. Wallace, “Aiml overview,” 2000.
- [14] M. G.G.Lee, H.K Kim and J.-H. Kim, *Natural language Dialog Systems and Intelligent Assistant*.
- [15] A. A. Wang YY, Deng L, “Spoken language understanding,” 2005.
- [16] M. R. Tur G, *Spoken language understanding – systems for extracting semantic information from speech*.
- [17] “Recast.ai.” <https://blog.recast.ai/build-your-first-bot-with-recast-ai/>.

Table des figures

1.1	Développement de l'IA et ses états	14
1.2	Différents agents conversationnels actuels	16
1.3	Complexité d'un bot	17
2.1	Cycle d'analyse d'une phrase	22
2.2	Analyse NLP	26
2.3	Architecture d'un chatbot	27
3.1	Modèle1 Penn Treebank	31
3.2	Modèle2 Penn Treebank	31
3.3	Analyse syntaxique	32
3.4	Réduction symbolique AIML	34
3.5	Diviser pour conquérir AIML	35
3.6	Système d'appel d'intent	37
3.7	Modèle mathématique des intents	37
3.8	Liste d'utterances pour l'intent order	38
3.9	Test de l'intent order	39
3.10	Dialogue flow	40
4.1	LUIS architecture	44
4.2	Résultat JSON intent order	45
4.3	Arbre Analyse syntaxique	46
4.4	Bot framework emulator	47
4.5	Scénario incident CRM	48
4.6	Architecture solution chatbot	49
4.7	Scénario incident CRM avec interfaces	50
4.8	Intents chatbot CRM	50
4.9	Utterances intent incident chatbot CRM	51
4.10	Test intent incident bon	52
4.11	Test intent incident mauvais	52
4.12	Chatbot CRM	53

Annexe A

Code traitement des intents

```
using System;
using CRMbot.Models;
using Microsoft.Bot.Builder.FormFlow;
using Microsoft.Bot.Builder.Luis;
using Microsoft.Bot.Builder.Dialogs;
using System.Threading.Tasks;
using Microsoft.Bot.Builder.Luis.Models;

namespace CRMbot.Dialogs
{
    [LuisModel("████████████████████", "████████████████████")]
    [Serializable]
    2 references
    public class LUISDialog : LuisDialog<CustomerModel>
    {
        private Func<IForm<CustomerModel>> buildForm;

        1 reference
        public LUISDialog(Func<IForm<CustomerModel>> buildForm) [...]

        [LuisIntent("")]
        0 references
        public async Task None(IDialogContext context, LuisResult result) [...]

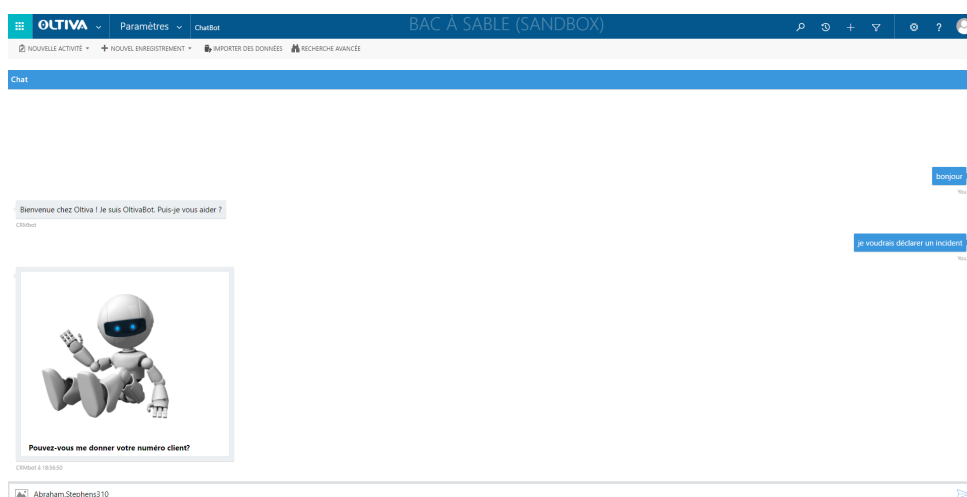
        [LuisIntent("Bonjour")]
        0 references
        public async Task Bonjour(IDialogContext context, LuisResult result) [...]

        #region INCIDENT
        [LuisIntent("Incident")]
        0 references
        public async Task CallCustomer(IDialogContext context, LuisResult result)
        {
            // use customerModel form flow
            var customerform = new FormDialog<CustomerModel>(new CustomerModel(), CustomerModel.BuildForm, FormOptions.PromptInStart);
            context.Call(customerform, CallCase);
        }
    }
}
```

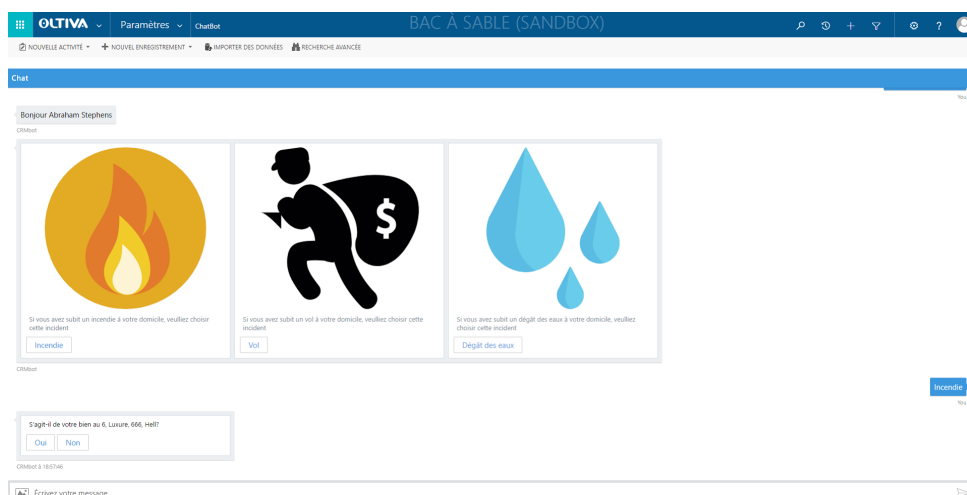

Annexe B

Dialogue avec le chatbot

Début du dialogue, intent bonjour, intent incident et demande du numéro client :



Choix de l'incident :



Confirmation de l'adresse, fausse donc demande des informations :

The screenshot shows the Oltiva chatbot interface. The header includes the Oltiva logo, navigation links (Paramètres, Chatbot), and the title 'BAC À SABLE (SANDBOX)'. Below the header, there are links for 'NOUVELLE ACTIVITÉ', 'NOUVEL ENREGISTREMENT', 'RAPORTER DES DONNÉES', and 'RECHERCHE AVANCÉE'. The chat area displays a series of questions from the chatbot and user responses:

- Chatbot: "S'agit-il de votre bien au 6, Lurue, 666, Heil?"
User: "Non"
- Chatbot: "Votre code postal de résidence s'il vous plaît?"
User: "50000"
- Chatbot: "Merci, votre ville?"
User: "Hartono"
- Chatbot: "Et pour finir, votre adresse s'il vous plaît?"
User: "2 rue du moulin"
- Chatbot: "Merci Un conseiller Oltiva va prendre contact avec vous dans un délai de 30 min. Comment souhaitez vous être contacté?"
User: "Téléphone"

The chatbot's name 'Oltiva' is visible below each question. At the bottom, there is a text input field with the placeholder 'Ecrivez votre message...' and a send button.

Choix de la manière d'être contacté, téléphone choisi donc confirmation du téléphone dans CRM. Pas d'autres questions donc fin de la conversation.

This screenshot continues the chatbot conversation from the previous one:

- Chatbot: "Merci Un conseiller Oltiva va prendre contact avec vous dans un délai de 30 min. Comment souhaitez vous être contacté?"
User: "Téléphone"
- Chatbot: "Pouvez vous me confirmer que votre numéro de téléphone est bien le suivant : 0614358371"
User: "Oui"
- Chatbot: "Très bien, nous vous recontactons rapidement. Puis - je vous aider sur quelque - chose d'autre?"
User: "Non"
- Chatbot: "Dans ce cas il ne me reste plus qu'à vous souhaiter une bonne journée Abraham Stephens."

The chatbot's name 'Oltiva' is visible below each question. The final message from the chatbot is a closing statement. At the bottom, there is a text input field with the placeholder 'Ecrivez votre message...' and a send button.