

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

Observatoire de Strasbourg - Avril à Juin 2018

Maître de stage : M. André SCHAAFF

Co-encadrants : M. Thomas BOCH – M. Sébastien DERRIERE

Professeur référent : Mme Karine SERIER

Assesseur : M. Christophe NICOLLE

Alexis Guyot

IQS4-B2



Observatoire astronomique
de Strasbourg

REMERCIEMENTS

Tout d'abord, je tiens à remercier mon maître de stage, M. André SCHAAFF, ingénieur de recherche à l'observatoire astronomique de Strasbourg, pour m'avoir accompagné et guidé avec bienveillance tout au long de ma mission au sein de l'établissement. Par la même occasion, je remercie M. Sébastien DERRIERE, astrophysicien, et M. Thomas BOCH, ingénieur de recherche, mes deux co-encadrants, dont les indications et les connaissances m'ont permis de mieux comprendre les notions astronomiques tout comme le fonctionnement des outils informatiques du service et qui ont de fait contribué à la réussite de mon projet.

Je suis également reconnaissant envers M. Pierre-Alain DUC, directeur de l'observatoire astronomique de Strasbourg, ainsi qu'envers M. Mark ALLEN, directeur du Centre de Données astronomiques de Strasbourg (**CDS**), pour m'avoir permis d'effectuer mon stage dans un endroit aussi incroyable et pour m'avoir permis d'assister à tous les évènements organisés pour les équipes de l'établissement (visites, séance de planétarium, réunions, cafés, séminaires, ...).

De manière générale, j'adresse toute ma gratitude à l'ensemble des employés de l'observatoire pour leur accueil chaleureux, ainsi que pour leur gentillesse, leur accompagnement et leur bienveillance tout au long de mon stage, faisant ainsi de ma première expérience professionnelle un souvenir à la fois marquant et agréable.

Enfin, je tiens à remercier toute l'équipe enseignante de l'IUT Informatique de Dijon pour leur accompagnement et leur dévotion au cours de ces deux années de DUT et plus particulièrement Mme Karine SERIER, enseignante tutrice lors de mon stage, pour avoir été à l'écoute tout au long de ma mission, Mme Patricia MENISSIER, enseignante d'expression-communication, pour avoir pris le temps de répondre à mes questions concernant ce présent document et M. Christophe NICOLLE, professeur des universités, pour m'avoir introduit la problématique du langage naturel en tant que tuteur lors du projet tutoré du troisième semestre et ainsi avoir indirectement contribué à ma participation à ce stage ainsi qu'à sa bonne réalisation.

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

TABLE DES MATIERES

<u>REMERCIEMENTS</u>	3
<u>TABLE DES ILLUSTRATIONS</u>	5
<u>CONVENTION TYPOGRAPHIQUE</u>	5
<u>PREMIÈRE PARTIE : PRÉSENTATION DU CONTEXTE</u>	6
I. INTRODUCTION	6
II. PRÉSENTATION DU SERVICE	8
III. DEFINITION DE LA MISSION	10
1. LA PROBLEMATIQUE AU SEIN DU SERVICE	10
2. LA TACHE A EFFECTUER	11
<u>DEUXIÈME PARTIE : LE DEVELOPPEMENT DU CHATBOT</u>	12
I. METHODE RETENUE	12
1. LES DIFFERENTES SOLUTIONS ENVISAGEES	12
2. LES MATERIELS ET LOGICIELS UTILISES	13
II. APPLICATION DE LA METHODE ET RESULTATS	17
1. LES DIFFERENTES PHASES DE LA REALISATION	17
2. LES DIFFICULTES RENCONTREES	27
III. CONCLUSION	28
1. LES SUGGESTIONS POUR L'ENTREPRISE	28
2. LES LEÇONS TIREES DE CE TRAVAIL	29
<u>LEXIQUE</u>	31
ACRONYMES	31
DEFINITIONS	31
<u>BIBLIOGRAPHIE</u>	32
<u>ANNEXES</u>	33
ANNEXE 1 – DIALOGFLOW	33
ANNEXE 2 – TWIKI	38
ANNEXE 3 – STRUCTURE D'AUTO COMPLETION/PREDICTION DE TEXTE	39
ANNEXE 4 – CAPTURES D'ECRAN DE L'APPLICATION	42
<u>RESUME ET MOTS CLES</u>	48

TABLE DES ILLUSTRATIONS

Figure 1 - Interface du chatbot développé au troisième semestre	6
Figure 2 - Prolongation du projet précédent lors du quatrième semestre	7
Figure 3 - La grande coupole, bâtiment central de l'observatoire de Strasbourg	8
Figure 4 - Accès en ligne aux différents outils du CDS	9
Figure 5 - Mon bureau pendant le stage	9
Figure 6 - Formulaire d'accès aux catalogues présents sur VizieR (un autre formulaire est ensuite nécessaire pour obtenir les données)	10
Figure 7 - L'initialisation des intentions sur l'outil Dialogflow	18
Figure 8 - Console d'évaluation des décisions prises par Dialogflow	19
Figure 9 - En cas d'attente, le chatbot s'adapte	20
Figure 10 - Toujours être clair avec l'utilisateur (Les mots soulignés renvoient vers la page dédiée à ce mot dans les services du CDS)	20
Figure 11 - Auto complétion et prédition de texte	21
Figure 12 - Interface du chatbot sur mobile	21
Figure 13 - Interface sur PC avec fenêtre de prévisualisation	22
Figure 14 - Récupérer la valeur d'une mesure pour un objet	22
Figure 15 - Connaitre les planètes membres d'une galaxie	23
Figure 16 - En cliquant sur le nom, l'utilisateur est directement renvoyé vers la page VizieR du catalogue	23
Figure 17 - Cette étoile en infrarouges	24
Figure 18 - Afficher une étoile	24
Figure 19 - Trouver des objets autour d'un autre	25
Figure 20 - L'objet est recherché et son nom ainsi que son type sont indiqués	25
Figure 21 - Extrait de mon journal de bord	38
Figure 22 - En-tête de ma page Twiki	38
Figure 23 - Utilisations possibles de Twiki	39
Figure 24 - Version simplifiée de la structure	40
Figure 25 - Arborescence du projet	47

CONVENTION TYPOGRAPHIQUE

Un mot dans ce format fait partie du corps de texte.

Un mot dans ce format est défini dans le lexique.

Un mot dans ce format est un mot anglophone non défini dans le lexique

PREMIÈRE PARTIE : PRÉSENTATION DU CONTEXTE

I. INTRODUCTION

Afin de conclure les deux années de DUT, une période de stage est aménagée durant les 12 dernières semaines de la formation afin d'offrir aux étudiants une première expérience professionnelle dans l'informatique. Dans ce cadre-là, j'ai effectué du 3 avril au 22 juin 2018 un stage à l'observatoire astronomique de Strasbourg et plus précisément dans le service du **CDS**, le Centre des Données astronomiques. Ma mission était de développer une nouvelle application permettant aux astronomes d'accéder aux principaux outils du service en utilisant le langage naturel, c'est-à-dire en faisant des phrases comme si on posait la question à une vraie personne, plutôt que de passer par des formulaires parfois longs et peu intuitifs.

A l'origine, j'ai été très intéressé par ce sujet de stage, découvert grâce à une annonce partagée par Maxime Ambard, responsable des stages pour ma promotion, parce que je possédais déjà une première expérience avec le traitement du langage naturel en informatique. En effet, lors du troisième semestre du DUT Informatique, un projet tutoré assez conséquent est mis en œuvre à l'IUT dans le but de nous faire travailler en équipe pendant une centaine d'heures par personne sur une mission informatique, dans les mêmes conditions que dans le monde professionnel. Le tuteur joue alors le rôle du client et travaille à travers une méthode de projet agile avec les étudiants ou des groupes qu'il supervise, qui jouent bien évidemment l'équipe informatique. Lors de ce semestre donc, j'ai eu l'occasion de travailler avec une équipe de 6 autres personnes dans le but de réaliser un **chatbot** à destination de M. Nicolle et de son équipe d'ingénieurs. Un **chatbot** est un robot, aussi appelé en français agent conversationnel, qui est capable de tenir une conversation avec un être humain. On parle donc à cette intelligence artificielle dans un langage naturel, dans le cadre du projet en anglais ou en français, celui-ci analyse et comprend la phrase qu'il vient de récupérer et rédige dans ce même langage une réponse cohérente et pertinente. On peut alors également apprendre à ce **chatbot** à aller récupérer des données sur Internet ou à effectuer des actions bien précises (faire un calcul, afficher une image, ...) afin de répondre aux questions et aux demandes des utilisateurs, en plus de juste pouvoir discuter avec lui. Lors du projet effectué pour M. Nicolle, notre **chatbot** était capable de traduire un mot dans de nombreuses langues, de récupérer des synonymes ou des mots apparentés et de chercher une définition. Finalement, ce projet se sera très bien passé, offrant à la fin un résultat à la fois satisfaisant pour notre équipe et pour notre tuteur.

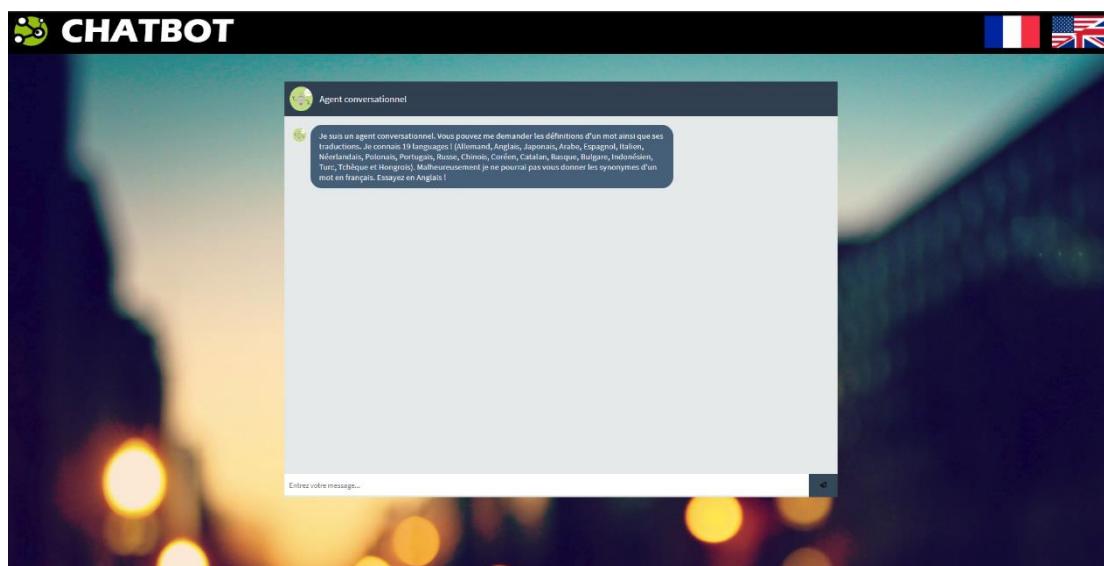


Figure 1 - Interface du chatbot développé au troisième semestre

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

C'est juste après la fin de ce travail que je suis tombé sur l'annonce de l'observatoire de Strasbourg. J'étais donc assez motivé à l'idée de travailler de nouveau sur ce type de projet. J'ai donc répondu à l'offre et quelques jours plus tard je fus pris. Lors du quatrième semestre, j'ai décidé de continuer le projet du semestre dernier pour ajouter de nouvelles fonctionnalités comme la reconnaissance vocale pour poser des questions directement au *chatbot* et de la synthèse vocale pour qu'il puisse nous lire les résultats. En plus de cela, j'ai implémenté la gestion de nouvelles intentions de l'utilisateur comme récupérer la météo ou des informations sur une personne ou une ville par exemple. Le but de ces prolongations était de m'entraîner pour le stage et d'approfondir mes connaissances sur le sujet. J'ai ainsi pu discuter avec André Schaaff, mon futur maître de stage, afin d'orienter mon travail dans un sens qui serait intéressant pour préparer ma mission à Strasbourg. Le 3 avril, j'ai ainsi pu commencer serein mon stage au centre de données astronomiques.

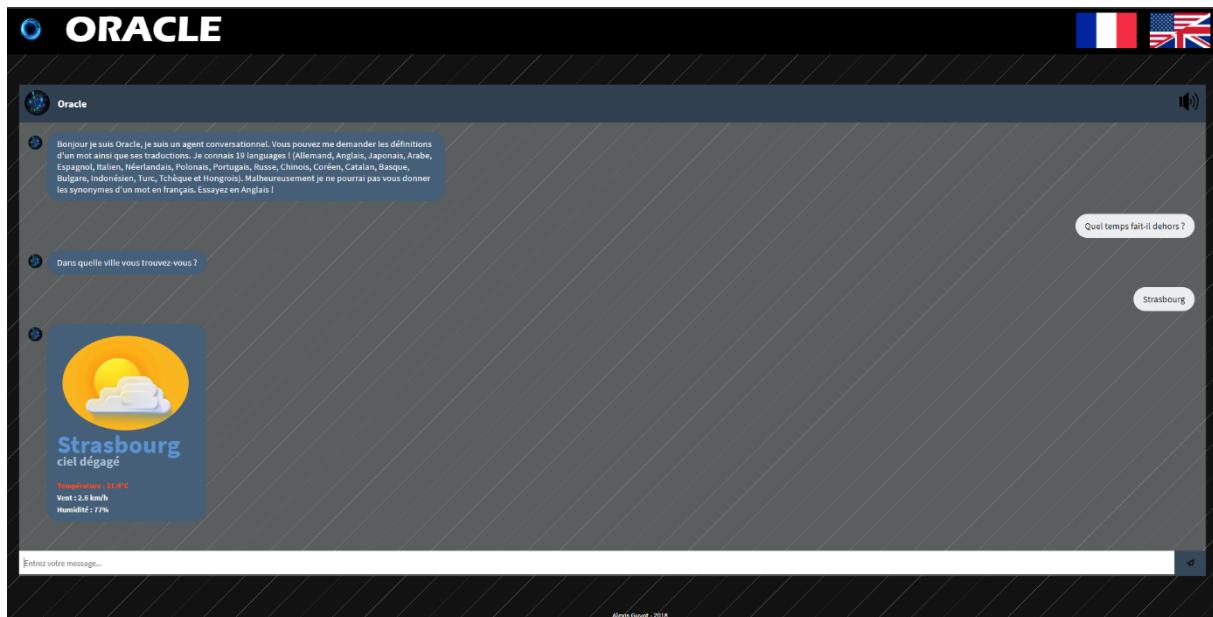


Figure 2 - Prolongation du projet précédent lors du quatrième semestre

II. PRESENTATION DU SERVICE

L'observatoire astronomique de Strasbourg, situé en plein cœur du campus universitaire, est un établissement construit en 1881 par l'empire allemand, dans le but d'exposer leur intérêt et leur avancement dans le domaine scientifique. Dans cette optique-là, ils construisirent trois bâtiments très modernes pour l'époque, reliés par des couloirs fermés pour protéger les astronomes et les instruments. Cependant, malgré tous ces efforts, l'observatoire souffrait d'un obstacle particulièrement gênant pour les astronomes : la météo. En effet, la ville de Strasbourg n'est pas réputée pour être l'endroit le plus ensoleillé de la planète et une bonne observation du ciel n'est possible que si celui-ci est dégagé. De fait, les observations dans un but professionnel s'arrêtèrent en 1960 (des observations amateurs sont encore proposées de nos jours). Pour faire face à ce problème, l'observatoire décida de se séparer en deux équipes, une spécialisée dans la recherche astronomique, l'équipe **GALHECOS**, et une spécialisée dans l'archivage, le traitement et la diffusion de données astronomiques. Ainsi, en 1972, le service du centre des données stellaires, aujourd'hui renommé en centre des données astronomiques de Strasbourg, fut créé.



Figure 3 - La grande coupole, bâtiment central de l'observatoire de Strasbourg

De nos jours, l'observatoire est dirigé par Pierre-Alain DUC et héberge la troisième plus grosse lunette astronomique de France ainsi que le planétarium de la ville. Malgré son statut de lieu historique et contrairement à d'autres établissements du genre, il héberge toujours des équipes de recherche. Il est d'ailleurs reconnu en tant qu'unité mixte de recherche, partagée entre le **CNRS** et l'Université de Strasbourg. De fait, il accueille et gère également les étudiants du parcours astrophysique du master physique de la ville. En tout, environ 80 personnes travaillent à l'observatoire, principalement des chercheurs (enseignants ou non), des documentalistes, des ingénieurs et des astronomes. Chaque année, plusieurs stagiaires sont engagés par l'équipe recherche et développement de l'observatoire pour les assister dans le développement ou l'amélioration d'outils offerts par le **CDS**.

Le Centre de Données Astronomiques, appelé dans la suite de ce rapport « **CDS** », désigne l'équipe de recherche travaillant à l'archivage, le traitement et la mise à disposition de données astronomiques pour le reste du monde. Il est composé d'environ un tiers d'informaticiens, d'un tiers de documentalistes et d'un tiers d'astronomes et est dirigé par M. Mark Allen. Ensemble, ils veillent au bon fonctionnement des trois principales plateformes mises à disposition par l'observatoire pour les astronomes du monde entier, à savoir Simbad, VizieR et Aladin, mais aussi des nombreux outils gravitant autour. Rapidement, Simbad correspond à une base de données « traditionnelle », stockant un ensemble de données associées à un ensemble d'objets astronomiques situés en dehors du système solaire ; VizieR correspond à une base de données contenant des catalogues (publications) astronomiques informatisés, une sorte de bibliothèque où les livres sont synthétisés pour ne garder que les données et leurs descriptions ; Et Aladin est un service proposant une visualisation du ciel, une sorte de « Google Map » de l'espace. Une présentation plus détaillée des outils et plateformes sera faite dans une partie dédiée [ci-après](#).



Figure 4 - Accès en ligne aux différents outils du CDS

Les principales fonctions de ce service étant de transformer des publications rédigées en un ensemble de données informatiquement traitables, le **CDS** se trouve au centre des enjeux autour des données astronomiques et de leur partage. De fait, il s'impose comme un acteur très important de l'**IVOA**, *l'International Virtual Observatory Alliance*, une association internationale qui développe des standards et des normalisations dans le traitement et la diffusion des données afin d'assurer l'interopérabilité des services astronomiques partout dans le monde. On peut ainsi comparer cette alliance au *World Wide Web Consortium*, l'organisme en charge du bon fonctionnement du Web.

J'ai eu l'occasion de travailler lors de mon stage dans la bibliothèque historique de l'observatoire, au milieu des milliers de revues stockées au cours de l'histoire et de quelques anciens outils d'astronomie. Dans celle-ci, un bureau a été aménagé pour que les stagiaires et les étudiants du parcours astrophysique puissent travailler dans le calme tout en étant intégrés au sein du même bâtiment qu'une grosse partie de l'équipe de recherche. J'étais accompagné de trois autres stagiaires de DUT Informatique. Même si nos missions étaient différentes, cela nous a permis de nous entre-aider au besoin, mais aussi tout simplement de profiter d'une bonne ambiance tout au long du stage.



Figure 5 - Mon bureau pendant le stage

III. DEFINITION DE LA MISSION

1. LA PROBLEMATIQUE AU SEIN DU SERVICE

Comme dit plusieurs fois depuis le début de ce rapport, l'essence même du **CDS** est d'offrir aux astronomes du monde entier un ensemble de données fiables et pérennes qu'ils pourront utiliser pour leurs travaux. Aujourd'hui, le service met déjà à disposition du reste du monde un ensemble d'outils, qui sont pour la plupart questionnables à travers des formulaires dans lesquels il est possible de préciser certains paramètres, de poser des conditions et de choisir des formats de sortie pour les données trouvées.

Figure 6 - Formulaire d'accès aux catalogues présents sur VizieR (un autre formulaire est ensuite nécessaire pour obtenir les données)

La problématique ayant amené à la création de mon stage était donc de proposer un nouveau moyen de communiquer avec toutes ces plateformes de données, de manière plus intuitive et moins « formelle ». Parmi les moyens envisagés pour faire cela, un outil permettant à un utilisateur de réaliser une requête en langage naturel a été privilégié. On entend par « requête en langage naturel » une question de la forme « *What is the effective temperature of Sirius* » plutôt qu'un ensemble de champs où il faudrait préciser « *Sirius* » comme nom d'objet et « *effective temperature* » comme mesure demandée. Il s'agit en effet d'un moyen de communiquer avec la machine en pleine expansion de nos jours, en grande partie grâce aux géants du Web, Google, Apple, Facebook, Amazon et même Microsoft. En effet, ceux-ci proposent tous aujourd'hui des outils populaires centrés autour de la compréhension du langage naturel : Google a son assistant, présent dans de nombreux produits de la firme comme le récent Google Home, et a dévoilé très récemment Google Duplex, une évolution du Google Assistant capable de passer un coup de téléphone à notre place et de discuter dans un langage naturel d'une qualité impressionnante ; Apple possède l'incontournable Siri, l'assistant intelligent présent sur les appareils de la marque à la pomme ; Facebook propose depuis quelques temps sur la plate-forme Messenger la possibilité de créer des bots utilisant le langage naturel pour communiquer avec des entreprises ou des services par exemple ; Amazon propose Alexa, un équivalent au Google Home (un appareil doté d'un micro et de haut-parleurs pour communiquer avec un assistant intelligent) ; Et enfin Microsoft propose Cortana pour accompagner les utilisateurs de Windows et Xiaolce, uniquement sur le marché chinois, une intelligence artificielle créée pour tenir une conversation avec un être humain et qui aujourd'hui fonctionne plutôt bien (au point de pouvoir créer et raconter des histoires cohérentes de près de 10 minutes à des enfants) et qui est très populaire au pays du soleil levant. A la vue de la popularité du concept et de leurs besoins, les ingénieurs du **CDS** ont donc décidé qu'un outil de ce type pourrait être un plus pour l'observatoire, utilisé dans une interface plus interactive et moins austère. L'outil étant principalement destiné à des professionnels, il était toutefois très important que la précision, la transparence du processus de raisonnement et des causes d'éventuels échecs ne soient pas sacrifiés au profit de ce point, au risque que l'utilisateur se retrouve perdu ou méfiant par rapport aux résultats.

Avant le mien, deux stages autour du traitement du langage naturel et de la définition de ses possibilités d'utilisation ont été menés à bien. Aux termes de ceux-ci, un premier prototype a ainsi été créé,

mais était loin d'être suffisant pour une potentielle mise en ligne et utilisation courante. Des améliorations au niveau de l'accompagnement de l'utilisateur mais aussi et surtout au niveau de la compréhension des requêtes et de la langue étaient nécessaires. Un troisième stage sur le domaine a donc été mis en place, le mien.

2. LA TACHE A EFFECTUER

En tant que stagiaire, ma mission était de continuer les travaux et investigations menés par mes prédécesseurs, à savoir effectuer un état de l'art en rapport avec le traitement du langage naturel. En effet, ce qui intéresse avant tout le **CDS** au travers des stages autour de ce thème est surtout de savoir si l'idée qu'ils ont eue est réalisable, viable et utile, ou non. L'objectif premier du stage était donc de répondre à cette simple question : Peut-on faire quelque chose avec ces besoins, et si oui jusqu'où peut-on aller ? Une fois cet état de l'art terminé, le **CDS** attendait de moi que je leur fasse une ou plusieurs propositions de concepts et d'outils à utiliser et qu'à partir de ceux-ci, je développe mon propre prototype et que j'en assure les tests. Celui-ci pouvait être au choix une version différente ou une évoluée de celui de mon prédécesseur, selon mes préférences et les idées que j'avais derrière la tête après étude approfondie du sujet. Finalement, j'ai décidé de tout reprendre depuis le début, le code du prototype laissé n'étant pas assez commenté et documenté afin que je puisse le reprendre moi-même. Ainsi, à partir de ces consignes, mon stage s'est articulé autour de quatre grands axes :

- Me familiariser avec l'existant, en particulier les trois outils principaux du **CDS**, Simbad, VizieR et Aladin : connaître leur contenu, leurs avantages, leurs inconvénients, comment les questionner ...
- Rechercher, étudier et comparer un certain nombre de solutions de traitement du langage naturel, puis mettre en place la solution privilégiée.
- A partir des résultats retournés après compréhension de la requête en langage naturel, articuler correctement les différents outils pour obtenir l'information demandée par l'utilisateur.
- Proposer une interface à la fois complète, intuitive, lisible et proposant un accompagnement efficace de l'utilisateur dans ses démarches.

DEUXIÈME PARTIE : LE DEVELOPPEMENT DU CHATBOT

I. METHODE RETENUE

1. LES DIFFERENTES SOLUTIONS ENVISAGEES

Dans cette partie, j'ai décidé de mettre en avant mon raisonnement concernant le choix du moteur de traitement du langage naturel. En effet, ce point me paraît être le plus pertinent car il s'agit de l'outil central du projet. En effet, sans une bonne compréhension des phrases entrées par l'utilisateur, on ne peut pas lui proposer de solution pertinente. De plus, il s'agit du point ayant le plus nécessité au cours de mon projet un état de l'art et une comparaison entre les solutions possibles. J'aurais également pu parler du choix de passer par un système de chat pour communiquer avec l'outil, mais celui-ci s'est fait naturellement puisque par habitude j'ai commencé à travailler avec une interface de ce type, comme lors de mes précédents projets, et que celle-ci a plu à mon maître de stage et à mes co-encadrants.

Après avoir lu de nombreux articles, publications, documentations d'outils de **NLU**¹, j'ai mis en avant un certain nombre de paramètres essentiels à prendre en compte pour comparer les différentes solutions. Tout d'abord, il fallait que l'outil soit gratuit et que son code source soit libre ou que l'outil soit au moins utilisable pour un projet professionnel. S'il se trouvait être une application en ligne requérable depuis un code extérieur, il fallait que le nombre de requêtes par jour ne soit pas trop limité, pour éviter les cas où un astronome ne pourrait plus utiliser l'outil à cause d'une limite journalière atteinte. En tant que développeur, il m'était aussi indispensable que l'outil possède une documentation complète et bien rédigée. Il fallait de même que son installation soit rapide, pas trop difficile (quitte à faire) et que celui-ci soit utilisable le plus rapidement possible, afin de ne pas perdre trop de temps pour le reste du stage. Les deux grandes techniques utilisées en traitement du langage naturel étant le **matching**, le fait d'associer un ensemble de mots-clés avec une idée, et le **machine learning**, une technique d'intelligence artificielle se basant sur l'apprentissage par expérience, le meilleur scénario était que l'outil utilise les deux, et pas juste la première, un peu moins efficace et surtout imposant une trop petite liberté à l'improvisation. Un gros enjeu du langage naturel est aussi la prise en compte du contexte. On entend par là que quand une personne demande « Quelle est la température de Sirius² ? » puis « Et sa position dans le ciel ? », l'application soit capable de comprendre que dans la deuxième question le sujet est toujours « Sirius ». Il s'agit d'un mécanisme très naturel chez l'Homme, mais complètement absent chez la machine. Si l'outil choisi pouvait proposer un système pour prendre en compte ce concept, cela serait un plus non négligeable.

Durant mes recherches, j'ai été confronté à de nombreux outils, mais seuls 4 ont retenu mon attention lors du premier tri, parce qu'ils étaient gratuits et qu'ils semblaient implémenter les principales fonctionnalités attendues. Parmi eux, on retrouve Wit.ai, l'outil de **NLU** possédé par Facebook, Luis.ai, l'outil de Microsoft, Rasa Stack, une bibliothèque développée par des indépendants, et Dialogflow, l'outil de Google. Il est également intéressant de noter que lors du troisième semestre à l'IUT, mon projet tutoré m'avait fait utiliser Rasa Stack pour le développement de notre chatbot, et que pour le quatrième j'avais été mené à utiliser Luis.ai. Ces deux outils ne m'étaient donc pas inconnus. En plus de cela, j'avais à ma disposition le travail effectué par mes prédécesseurs, qui avaient fait le choix de quasi entièrement redévelopper un outil de **NLU** à partir d'une bibliothèque open source créée et diffusée par l'université de Stanford. J'ai alors établi le tableau comparatif suivant :

¹ Natural Language Understanding – Compréhension/Traitement du langage naturel en français

² Sirius est une étoile.

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

	Wit.ai	Luis.ai	Rasa Stack	Dialogflow	Travaux préd.
Gratuit et libre	Oui	Oui	Oui	Oui	Oui
Méthode d'utilisation	API requêteable	API requêteable	Pas open source mais utilisable pour projets pro	API requêteable	Oui
Limite	Non, sous réserve d'une utilisation raisonnable	Oui (1000 requêtes par mois)	Aucune	Presque aucune*	Aucune
Documentation	Bonne	Bonne	Bonne	Très bonne	Aucune
Mise en place	Simple	Simple mais longue	Longue et compliquée	Simple	Déjà mis en place
Techniques utilisées	Un peu des deux, mais plus matching après quelques tests	Un peu des deux, mais plus matching après quelques tests	Les deux	Les deux	Matching
Contexte	Difficile d'utilisation	Non	Non	Oui	Non
Fonctionnalités supplémentaires	Aucune	Aucune	Apprentissage de comportements	Nombreuses : smalltalks, sauvegarde et restauration, ...	Aucune

* Une limite est présente pour Dialogflow mais uniquement pour éviter les surcharges des serveurs de Google. Celle-ci est extensible gratuitement et à la demande si la justification donnée est pertinente. Par défaut, elle est de 180 requêtes par minute. Il n'y a pas de limite journalière ou mensuelle pour les requêtes textuelles.

Après comparaison, j'ai donc décidé de choisir Dialogflow, l'outil de Google, car il vérifiait quasiment tous mes critères de sélection. Concernant la méthode d'utilisation, il est vrai qu'il est toujours plus appréciable lorsqu'on développe une application d'avoir une totale maîtrise et un total contrôle de tous les outils utilisés. De fait, dépendre d'une application tierce pour un élément aussi gros pour le projet que la compréhension du langage pourrait paraître être un point négatif, mais mes maîtres de stage n'avaient rien contre. Cependant, dans un souci d'anticipation, j'ai veillé lors du développement de l'application à ce que le changement du moteur de traitement du langage naturel ne soit pas un problème, en rendant le reste du code ouvert à l'extension (on peut l'utiliser différemment, avec un autre outil par exemple), mais fermé à la modification (En cas de changement, pas besoin de revérifier ou modifier tout le code, les appels à l'outil de **NLU** ne se faisant qu'à un seul endroit). Vous pourrez trouver dans la partie « [annexes](#) » un document de synthèse sur Dialogflow que j'ai rédigé et fourni à mes maîtres de stage pour qu'ils comprennent l'utilité et les avantages de cet outil.

2. LES MATERIELS ET LOGICIELS UTILISÉS

A. LE MATERIEL



UBUNTU

J'ai effectué mon stage sur un poste de travail sous Linux Ubuntu, sur lequel je possédais des droits administrateurs. Je ne possédais pas de contrôle/blocage pour certains sites Internet comme cela peut être le cas parfois dans les établissements scolaires. Cela m'a permis d'avoir une grande liberté concernant ma façon de travailler, en pouvant être complètement libre concernant les logiciels utilisés. L'utilisation de Linux plutôt que de Windows n'a pas spécialement été un problème, sauf pour certaines applications qui n'étaient pas disponibles. Cela a même été intéressant pour pouvoir tester comment fonctionnait mon programme sous différentes distributions (Linux avec mon poste de travail, Windows avec mon ordinateur portable et Android avec mon téléphone portable).

B. LES LOGICIELS/OUTILS EN LIGNE



NETBEANS

Netbeans est un environnement de développement permettant de créer des applications dans de nombreux langages. Il est notamment possible de développer des projets web et de les relier à un serveur accessible par toutes les machines présentes sur le même réseau que l'ordinateur de développement (Ils étaient donc accessibles dès que mon ordinateur était allumé et le programme lancé). J'ai choisi d'utiliser ce logiciel parce qu'il est gratuit, disponible sous Linux, que mon prédécesseur l'avait lui aussi utilisé et que j'avais déjà eu de nombreuses occasions de m'en servir à l'IUT, il m'était donc familier. Il s'est avéré être un bon choix du fait de sa capacité à pouvoir rendre accessible les applications web aux membres du réseau. Ce point a en effet facilité la vie à mes encadrants qui pouvaient ainsi juger de mon avancement sans avoir à toujours venir me voir.



TWIKI

Twiki est une plate-forme de travail collaboratif utilisée par les membres du **CDS**. Sur celle-ci, tous les stagiaires possèdent leur propre page dédiée. L'objectif de celle-ci était de garder du début à la fin de notre stage des traces de notre avancement (grâce à un journal de bord), de nos difficultés, de nos objectifs, ... Vous pourrez trouver en [annexes](#) quelques captures du logiciel.



DIALOGFLOW

Dialogflow est une plate-forme de traitement du langage naturel possédée depuis 2016 par Google. Elle offre un ensemble de fonctionnalités en rapport avec le traitement du langage naturel. Celles-ci sont accompagnées et renforcées par un algorithme de machine learning spécifique à chaque projet (comprenez ici que l'algorithme s'adapte à l'expérience gagnée par le bot au fur et à mesure de son utilisation). L'outil est hébergé dans les serveurs de Google et est accessible à travers un système de requêtes HTTP³. Parmi les fonctionnalités très intéressantes proposées par Dialogflow, celle qui m'a fait choisir cet outil est la gestion du contexte. Par contexte, on entend que si un utilisateur demande « quel est le prix du scooter rouge », et tout de suite après « et celui du bleu ? », l'outil est capable de comprendre qu'on parle toujours d'un scooter. Une présentation bien plus complète et détaillée des avantages de l'outil est disponible dans la partie « [Annexes](#) ». Dialogflow est l'outil qui nous servira à comprendre ce que veut l'utilisateur lorsqu'il entre son message en langage naturel.

³ Protocole de communication entre un programme hébergé sur un serveur et un autre sur l'ordinateur de l'utilisateur.

C. LES OUTILS DU CDS



SIMBAD

Simbad est une base de données créée, alimentée et maintenue par le **CDS**. Elle contient en 2018 près de 9 millions d'objets astronomiques (étoiles, galaxies, nébuleuses, ...) situés en dehors du système solaire et est requêtée environ 500 000 fois par jour. Quand un objet est entré dans Simbad, les différentes mesures faites sur lui sont renseignées et sont donc accessibles rapidement pour les astronomes. Pour accéder à celles-ci, une interface web est proposée. Grâce à plusieurs critères qui lui sont proposés (nom de l'objet, position dans le ciel, référence bibliographique, ...), l'utilisateur peut récupérer les mesures entrées dans la base. Il existe également un moyen plus technique utilisant l'**ADQL**, une version modifiée du **SQL**⁴ adaptée aux besoins de l'astronomie. Grâce à l'**ADQL**, on peut récupérer dans une application des données de Simbad structurées dans un format compréhensible par la machine (comme le **JSON** ou le **XML**). Contrairement à la base VizieR, qui contient elle aussi des données sous une forme différente, Simbad se requête plus facilement et rapidement par une machine, et est donc très utile pour les outils informatiques manipulant les propriétés les plus courantes en astronomie. Lorsqu'un utilisateur demande une mesure en langage naturel à notre outil, il y a donc de très grandes chances qu'on puisse trouver ce qu'il recherche dans Simbad, très rapidement.



VIZIER

VizieR est la deuxième base de données gérée par le **CDS**. Elle a été créée en 1992 et contrairement à Simbad, elle ne contient pas directement des données brutes mais des catalogues. En astronomie, un catalogue est une version informatisée d'une publication faite par un organisme et contenant des recherches d'astronomes, des données récupérées par des outils au sol ou dans l'espace, Par « informatisée », on entend que les données présentes sont regroupées dans des tableaux plutôt que noyées dans des lignes de texte. Dans un sens, VizieR peut être vue comme une base de données contenant des bases de données. Elle présente l'avantage d'être beaucoup plus facilement utilisable par les hommes que par les machines. En effet, la base n'est rien d'autre qu'une bibliothèque de livres où les pages sont des tableaux de données. La base est ainsi très complète et permet d'obtenir des propriétés plus spécifiques et pointues sur un objet particulier ou un type d'objet en général. Cependant, comme pour les livres dont la façon de présenter l'histoire varie d'un exemplaire à l'autre, le contenu des tables peut varier d'un catalogue à l'autre, ce qui rend son traitement informatique plus compliqué. Heureusement, des moyens ont été mis en œuvre par le **CDS** au fil des années, mais récupérer une donnée précise dans VizieR se trouve quand même être plus long que sur Simbad. La base reste toutefois utilisable pour l'outil de langage naturel, mais doit être considérée dans un second temps, uniquement si la recherche n'a donné aucun résultat sur la base précédente.

⁴ Langage informatique utilisé pour récupérer des données dans une base de données « classique ».



ALADIN(-LITE)

Pour présenter Aladin de manière vulgarisée, on pourrait simplement dire qu'il s'agit d'un « Google Map » du ciel. En effet, comme sur l'application de cartographie, il est possible de se balader sur la voûte céleste et de découvrir les étoiles, galaxies et autres nébuleuses. L'application permet également de changer la longueur d'onde étudiée, puisque certaines étoiles sont invisibles à l'œil nu.

Dans le cadre de mon projet, la plate-forme Aladin-Lite, la version en ligne du logiciel utilisable depuis un navigateur, me permet d'afficher une image de l'objet astronomique demandé par l'utilisateur.

D. LES LANGAGES INFORMATIQUES

Au début du stage, j'ai proposé l'idée de développer le chatbot comme une application web (accessible depuis un navigateur) plutôt que comme une application bureau (un logiciel qu'on installerait et lancerait ensuite), et ce pour plusieurs raisons. Tout d'abord pour faciliter son installation et son utilisation. En effet, aucun problème avec une application web puisqu'il suffit de posséder un navigateur internet (Chrome, Firefox, ...) pour pouvoir avoir accès aux fonctionnalités. Cela le rend également de fait accessible sur les appareils mobiles comme fixes. Ensuite, parce que l'outil allait puiser ses connaissances dans des sources qui sont toutes situées sur le web, que rien ne se trouve en local sur l'ordinateur. Les différents langages de ce domaine de l'informatique permettent facilement ce genre d'actions. Enfin, parce que le prototype proposé par mon prédécesseur était déjà une application web. Partir sur ce type d'application me permettait ainsi de pouvoir me resservir de certaines parties de son travail. Pour développer l'outil, j'ai donc utilisé les langages suivants :



HTML/CSS

Le HTML n'est pas à proprement parler un langage de programmation mais un langage dit de balisage. Si on compare un site web avec une construction LEGO, le HTML est la boîte de jouets qui contient les petites briques. C'est en assemblant les « briques » du HTML, les balises, qu'on construit la structure d'une page web.

Le CSS est un langage de description. On s'en sert pour décrire l'apparence des éléments d'une page web (leur couleur, leur taille ...).



JAVASCRIPT

Javascript est le langage de programmation web le plus utilisé. Il permet de rendre les pages dynamiques et de gérer les interactions avec l'utilisateur. C'est un langage asynchrone, c'est-à-dire qu'il n'attend pas forcément qu'une instruction soit totalement terminée pour passer à une suivante. Cela permet de gérer beaucoup plus facilement les événements déclenchés par l'utilisateur et surtout les requêtes vers un serveur pour récupérer des données (très utile dans notre cas, puisque les connaissances sont toutes sur des serveurs). En effet, cela permet de ne pas bloquer l'application tant que les résultats ne sont pas arrivés et ainsi de pouvoir afficher des animations par exemple.



JQUERY

JQuery n'est pas un langage de programmation mais j'ai décidé d'en parler dans cette partie car il s'utilise en général en tant que complément à Javascript. Il s'agit en réalité d'une bibliothèque, une sorte d'extension de ce dernier proposant plus de fonctionnalités, comme la possibilité de faire de l'auto complétion ou des listes déroulantes par exemple, plus facilement et rapidement qu'avec le langage seul. Elle permet ainsi de simplifier le code.

II. APPLICATION DE LA METHODE ET RESULTATS

1. LES DIFFERENTES PHASES DE LA REALISATION

A. ORGANISATION

La méthode agile a été choisie pour mener à bien ce projet. Il s'agit d'une organisation particulière du travail utilisée pour réaliser une tâche assez conséquente. Dans celle-ci, le client est directement impliqué dans le développement du projet et veille et participe régulièrement à son évolution à travers des réunions régulières avec l'équipe en charge de la réalisation. Ainsi, cette dernière profite de retours réguliers ce qui lui assure de livrer à la fin un produit qui plait et convient au client. L'autre caractéristique principale de la méthode agile concerne son rapport au produit final. En effet, alors qu'une méthode classique privilégiera une anticipation complète des besoins, formalisée dans le cahier des charges, offrant ainsi la possibilité de réaliser le produit d'une traite, cette méthode fonctionne de manière incrémentale. Lors du premier cycle (temps écoulé entre deux réunions avec le client), l'équipe développe une version minimale du projet sur laquelle viendra se greffer de nouvelles fonctionnalités et de nouveaux besoins chaque cycle, et ce de manière itérative. En effet, à chaque réunion avec le client, une liste d'objectifs, pouvant être des modifications sur le produit existant ou de toutes nouvelles fonctionnalités, est dressée et servira de fil directeur durant le cycle suivant. Ainsi, de réunion en réunion, le produit se rapproche de l'objectif final du client.

Plusieurs raisons ont motivé le choix de cette méthode pour le projet. Tout d'abord, parce qu'elle correspondait totalement à la nature du stage. En effet, comme décrit dans la partie « [Définition de la mission](#) », mes encadrants n'avaient pas d'idée précise d'un outil utilisant le langage naturel. Ils voulaient avant tout savoir ce qui était faisable et jusqu'où il était possible d'aller. Ce type d'objectif correspond totalement avec le fonctionnement de la méthode agile : A chaque cycle on essaye d'aller plus loin. Ensuite, un autre intérêt de cette méthode, en particulier lorsque des stagiaires vivant leur première expérience professionnelle sont impliqués, est l'encadrement et l'accompagnement. Effectivement, utiliser une organisation agile implique que le client, ici mon maître de stage, soit présent pour orienter le travail de l'équipe de travail, ici moi-même. De fait, il était possible pour lui de vérifier si tout se passait bien de mon côté, si je travaillais bien, etc., et pour moi de pouvoir profiter de ces moments pour poser des questions par exemple. Enfin, le dernier avantage de cette méthode est sa souplesse, la proximité instaurée entre le client et l'équipe de développement. En effet, cela m'a permis de pouvoir moi-même proposer des améliorations et des correctifs lors des réunions et de pouvoir en discuter avec mes encadrants. Ainsi, même si les décisions finales étaient prises par eux, la flexibilité de la méthode me permettait d'avoir une place dans ce projet et de ne pas juste être une « machine à développer ».

Concernant les limites de la méthode agile dans le cadre de mon projet, on peut revenir sur le fait que le client soit pas mal impliqué tout au long de la réalisation. Bien qu'il s'agisse avant tout d'un avantage pour les raisons citées ci-dessus, ce point peut s'avérer être un inconvénient dans certains cas. On peut notamment penser au fait que mes trois encadrants (mon maître de stage et mes deux cotuteurs) avaient tous des emplois du temps différents et que parfois trouver un créneau pour les réunions qui convenait à tout le monde n'était pas forcément facile.

Mon stage s'est étalé en tout sur 6 cycles d'une dizaine de jours ouvrables en moyenne à chaque fois. Les deux premiers cycles concernaient la familiarisation avec l'environnement, l'état de l'art autour du langage naturel, la recherche d'un moteur de **NLU** et la création de prototypes pour effectuer des tests. À la suite de cela se sont enchainés deux cycles de développement afin d'obtenir une application utilisable implémentant les fonctionnalités minimales définies par mes encadrants avant le début du stage. Celles-ci étaient exprimées à travers une dizaine de messages possibles qui devaient être compris par le portail. Vous pourrez les découvrir dans la partie [annexe](#) (dernière capture de Twiki). Enfin, mon stage s'est terminé sur deux cycles marqués par de nombreuses présentations à des acteurs extérieurs au projet. En effet, André

Schaaff a présenté mon travail lors d'une réunion semestrielle de l'IVOA à Victoria au Canada et j'ai pu moi-même expliquer le but du projet et animer des démonstrations pour plusieurs employés de l'observatoire pendant cette période. Le but était de recueillir des conseils et des idées d'améliorations des futurs utilisateurs, ceux-ci possédant des spécialités et des besoins différents les uns et les autres (astronomes, informaticiens, documentalistes ...). Il s'agissait également de profiter de ces regards extérieurs pour identifier et corriger un maximum de bugs. Après chaque présentation, je pouvais aussi ajouter de nouvelles fonctionnalités, souvent plus poussées donc demandant plus de travail, mais proposées directement par les utilisateurs. Au terme de mon stage, j'ai ainsi pu fournir un produit implémentant les fonctionnalités décrites dans la partie suivante.

B. LES FONCTIONNALITES IMPLEMENTEES

LA COMPREHENSION DU LANGAGE NATUREL

Point de départ de chaque interaction entre l'utilisateur et la machine, la compréhension du langage naturel est effectuée dans le cadre de ce projet par Dialogflow. Cet outil, présenté à de nombreuses reprises dans les parties précédentes, agit comme un filtre au lancement du processus de recherche. Du côté de son entrée, il va recevoir du programme une phrase en langage naturel puis va l'analyser et la comprendre grâce ses algorithmes de **matching** et de **machine learning**. Une fois cela fait, il va retourner à ce même programme les résultats de son analyse. Ceux-ci seront constitués entre autres d'une intention, ce que veut l'utilisateur, et d'une ou plusieurs entités, ces paramètres importants pour la récupération des données (le nom d'une étoile, d'une mesure, d'un catalogue, ...). Pour cette fonctionnalité, mon travail a essentiellement consisté à paramétriser l'outil de Google. J'ai dû pour chaque intention réfléchir à un ensemble le plus exhaustif possible de manières d'exprimer la volonté en question. Il a ensuite fallu que j'entre toutes ces phrases dans un corpus de travail pour l'algorithme. Plus ce corpus est gros, plus les décisions prises par l'outil sont fiables. En moyenne, j'essayais donc dans un premier temps de fournir une dizaine de phrases différentes. Afin de conclure l'initialisation des intentions, il suffisait ensuite de faire le travail de **NLU** à la place de l'intelligence artificielle sur les membres du corpus. Une fois cela fait, les algorithmes se lançaient et l'outil prenait le relais.

The screenshot shows the Dialogflow web interface. On the left, the sidebar includes sections for 'astroboy' (language: en, en-AU), 'Intents' (selected), 'Entities', 'Fulfillment', 'Integrations', 'Training', 'History', 'Analytics', 'Prebuilt Agents', and 'Small Talk'. The main area displays the configuration for the 'get_measure' intent. At the top, there's a title 'get_measure' with a 'SAVE' button and a three-dot menu. Below the title is a table for parameters:

PARAMETER NAME	ENTITY	RESOLVED VALUE
meas	@meas	Parallax
meas	@meas	proper motion
oid	@oid	Arcturus

Below the table are several examples of user queries:

- 99 Parallax and proper motion of Arcturus
- 99 What is the parallax, the spectral type, the position, the proper motion, the redshift, the magnitudes and the distance of Sirius
- 99 Galactic coordinates and spectral type of Arcturus
- 99 Tell me everything about T Tau
- 99 Could you search the proper motion of pulsars
- 99 Tell me everything about pulsars
- 99 Parallax of T Tau

To the right of the intent configuration, there's a 'Try it now' button and a microphone icon. Below that, a message says 'Please use test console above to try a sentence.' and a link 'See how it works in Google Assistant.'

Figure 7 - L'initialisation des intentions sur l'outil Dialogflow

Cependant, mon travail sur l'outil Dialogflow ne s'arrêtait pas là. En effet, afin d'améliorer la fiabilité et l'efficacité de la prise de décision, il faut par la suite régulièrement commenter les résultats du traitement du langage naturel. Dialogflow propose ainsi un outil d'entraînement où il est possible pour chaque entrée d'indiquer à l'intelligence artificielle que sa compréhension de la phrase est correcte et d'éventuellement proposer une correction si ce n'est pas le cas. Une fois la prise de décision approuvée, le

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

message de l'utilisateur est automatiquement ajouté au corpus d'exemples utilisés pour la prise de décision, ce qui au fil du temps l'aide à s'améliorer (plus de matière pour comprendre comment découper les phrases, plus de façons de dire la même chose reconnues, ...). Ainsi, j'ai pu effectuer ces actions tout au long du stage pour améliorer la compréhension du langage naturel.

The screenshot shows the Dialogflow interface. At the top, it says "Hello" and "Jun 7 50 REQUESTS 0 NO MATCH". There is a blue "APPROVE" button. Below this, under "USER SAYS", there is a box containing "how are you". Under "INTENT", it says "smalltalk.greetings.how_are_you". In the middle section, under "USER SAYS", there is a box containing "Parallax and proper motion of Electra". Below this, there is a table with three rows:

PARAMETER NAME	ENTITY	RESOLVED VALUE	
meas	@meas	Parallax	X
meas	@meas	proper motion	X

At the bottom of this section, under "INTENT", it says "get_measure". On the right side of the interface, there is a sidebar with various icons and a dropdown menu.

Figure 8 - Console d'évaluation des décisions prises par Dialogflow

Concernant les fonctionnalités liées au langage naturel, le **chatbot** est également capable de gérer les banalités d'une conversation comme les salutations, les remerciements, les insultes, ... L'intérêt de ce point est d'offrir des discussions plus vivantes entre l'utilisateur et la machine. Comme abordé lors de la [comparaison entre les moteurs de NLU](#), Dialogflow propose également la possibilité de gérer un système de contexte gardant en mémoire les précédentes entités retournées pendant 20 minutes ou pendant 5 messages envoyés par l'utilisateur. Cela offre une meilleure fluidité d'utilisation de l'outil, en évitant les répétitions excessives d'informations d'un message à l'autre. Enfin, le **chatbot** est capable de gérer correctement des précisions de termes astronomiques comme par exemple « coordonnées galactiques », qui sont un type de coordonnées dans le ciel, au lieu de « coordonnées » qui est la désignation générale pour la position d'un objet. Mais aussi l'ambiguïté parfois pas évidente à lever concernant la différence entre nom d'objet et type d'objet. En effet, il arrive régulièrement qu'une étoile donne son nom à un type générique désignant d'autres étoiles qui lui ressemblent. Par exemple, l'étoile « T Tau » a donné son nom à un type qui s'appelle « T Tauri ». Dialogflow étant un outil trop générique pour gérer ce type de subtilités, il a fallu que je rajoute moi-même des fonctions Javascript afin de lever au maximum ces ambiguïtés du langage astronomique.

UNE EXPERIENCE UTILISATEUR AGREABLE ET COMPLETE

Là où le sujet de mon stage mettait un point d'honneur était sur le fait que le **chatbot** devait nécessairement proposer des solutions pour accompagner l'expérience de l'utilisateur et ainsi empêcher qu'il se retrouve à se débrouiller seul sans savoir comment régler son problème. En effet, le traitement du langage naturel se basant sur un système probabiliste, où une intelligence artificielle doit prendre une décision qui aura un impact sur le résultat, il est impossible d'affirmer avec certitude que toutes les réponses retournées à l'utilisateur seront pertinentes et cohérentes. Il est donc indispensable que celui-ci comprenne le cheminement qui a amené à la production de la réponse qu'il a sous les yeux, qu'elle soit bonne ou mauvaise. Ceci est d'autant plus vrai que l'outil est à destination de professionnels qui utiliseront les données retournées par l'outil à des fins de recherche. Il est donc très important que l'utilisateur sache d'où les valeurs affichées viennent, les calculs qui ont éventuellement été effectués et qu'il puisse avoir un accès rapide à la source des données pour aller vérifier lui-même s'il est dubitatif devant les résultats.

Ceci passe par une certaine transparence dans les réponses du **chatbot**, celui-ci devant présenter à chaque fois de manière explicite l'intention de l'utilisateur qu'il a discerné ainsi que les paramètres qu'il a pris en considération pour construire son message. La source des données est également toujours visuellement identifiable grâce à la présence des logos des services du CDS et des liens vers ceux-ci sont disséminés dans la réponse du **chatbot**. Quand une erreur survient à un moment dans le processus, celle-ci est résumée dans l'interface sous la forme d'un message du **chatbot** qui explique les raisons de l'échec. Si celle-ci relève d'une erreur technique propre au code de l'outil, une version plus technique est affichée dans la console du navigateur afin d'orienter les développeurs. Si la réponse met du temps à arriver, une animation est également affichée pour indiquer à l'utilisateur que le processus pour répondre à sa requête est en cours et que l'application n'a pas planté.

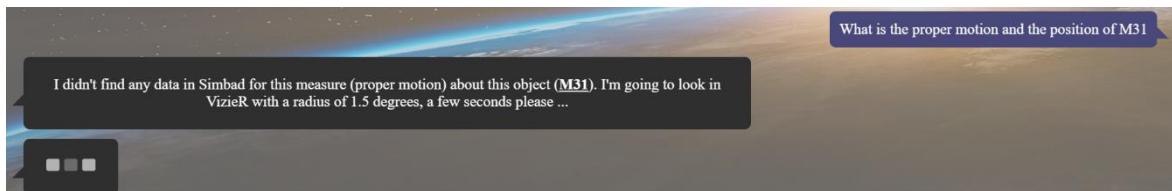


Figure 9 - En cas d'attente, le chatbot s'adapte

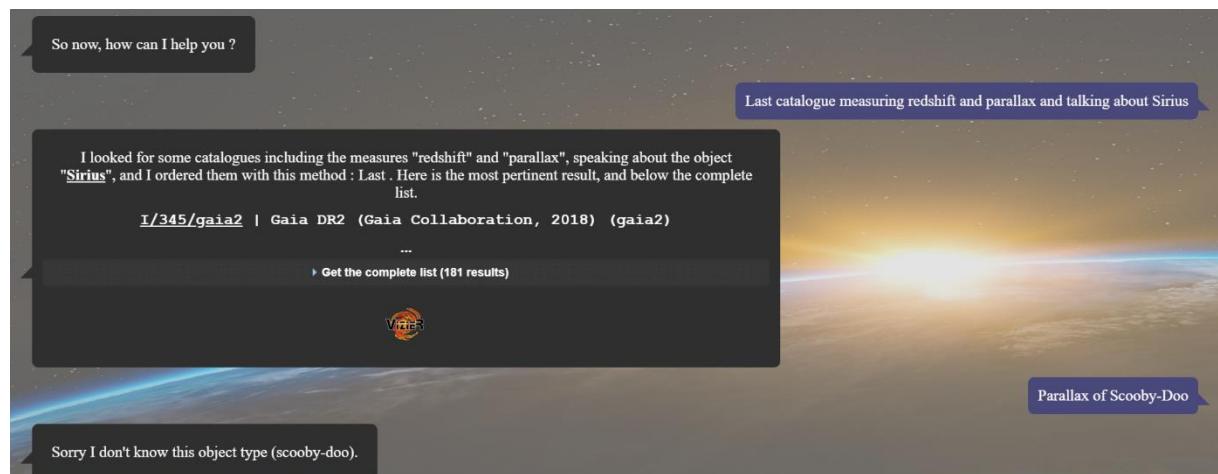


Figure 10 - Toujours être clair avec l'utilisateur (Les mots soulignés renvoient vers la page dédiée à ce mot dans les services du CDS)

Mais ceci passe également par une prise en charge de l'utilisateur avant même qu'il envoie son message. En effet, on peut essayer de l'orienter directement dans la bonne direction grâce à certaines fonctionnalités. Parmi elles, j'ai proposé d'ajouter un message d'accueil proposant une version raccourcie de ce que je suis en train de faire dans cette partie du rapport, à savoir décrire les fonctionnalités techniques prises en charge par l'outil. Pour chaque, un ou plusieurs modèles de phrases types sont proposées, pour aider l'utilisateur à construire un message qui le fera obtenir ce qu'il souhaite. En cliquant sur ces modèles, des exemples sont directement ajoutés dans la barre de recherche et l'utilisateur n'a plus qu'à confirmer pour obtenir un exemple de résultat. Par mimétisme, il pourra ainsi se servir de l'outil comme il le souhaite. En plus de cela, nous avons décidé d'ajouter un système d'auto complétion proche de celui proposé par les smartphones, c'est-à-dire suggérant une version complète du mot en cours d'écriture et pas de la phrase comme cela peut être le cas dans un moteur de recherche. Ce système est couplé à un algorithme de prédiction de texte qui s'active quand l'utilisateur entre un espace et s'apprete à écrire un nouveau mot. Des suggestions de mots suivants lui sont proposées. L'algorithme et la structure ont entièrement été conçus par moi-même puisque mes recherches sur le sujet ne m'ont pas permises de trouver une structure générique efficace et répondant à nos attentes. Une description plus technique de ce point, avec une description de ses avantages et de ses limites, est présente en [annexe](#). Un historique des dernières recherches de l'utilisateur est accessible

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

rapidement avec les touches « Haut » et « Bas » pour pouvoir réutiliser des phrases proches sans avoir à tout retaper.

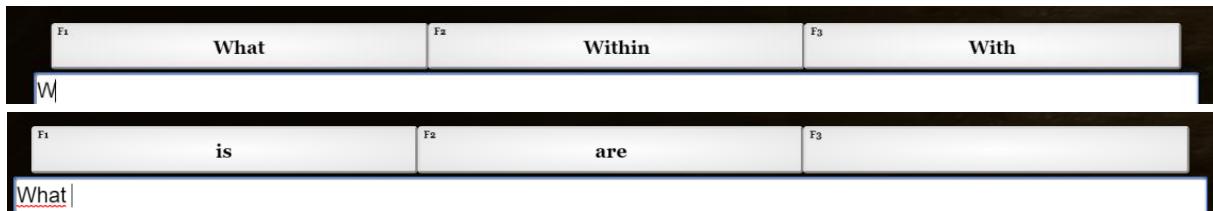


Figure 11 - Auto complétion et prédition de texte

Au niveau de l'interface, j'ai proposé une disposition mono-page, c'est-à-dire que toutes les parties essentielles du *chatbot* sont accessibles sur l'écran d'accueil sans avoir à descendre dans la page. Une zone textuelle s'étendant sur toute la largeur et 80% de la hauteur de l'écran accueille les messages de l'utilisateur et du *chatbot*. Quand trop de messages sont présents pour pouvoir tous les afficher, les plus anciens sont poussés vers la sortie de la zone par les plus récents mais restent évidemment accessible grâce à une barre de défilement, comme sur une interface de « chat » traditionnelle. Cette interface s'adapte à l'écran de l'utilisateur, qu'il soit sur ordinateur, smartphone ou tablette, ce qui permet une utilisation sur tous les supports. Pour ne pas dépayser les astronomes, les résultats récupérés sont affichés en respectant la notation scientifique utilisée sur les services du CDS et optent pour une police de caractère identique.

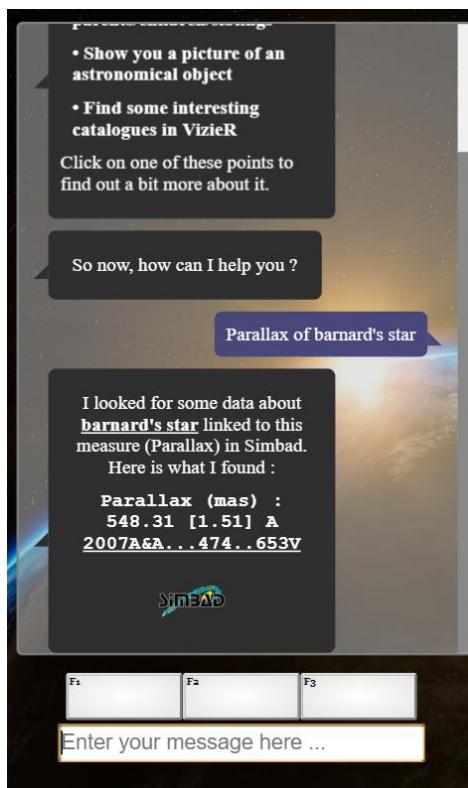


Figure 12 - Interface du chatbot sur mobile

J'ai également proposé pour accélérer et fluidifier l'expérience utilisateur de proposer un système de prévisualisation lorsque l'utilisateur survole un nom d'objet présent dans la base de données Simbad. Quand il le fait, une petite fenêtre suivant la souris apparaît pour afficher un maximum des informations principales sur l'objet pointé. Quand l'utilisateur clique sur le nom de l'objet, une recherche pour obtenir toutes les informations disponibles sur Simbad sur celui-ci est lancée et les résultats affichés dans un nouveau message du *chatbot*.

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

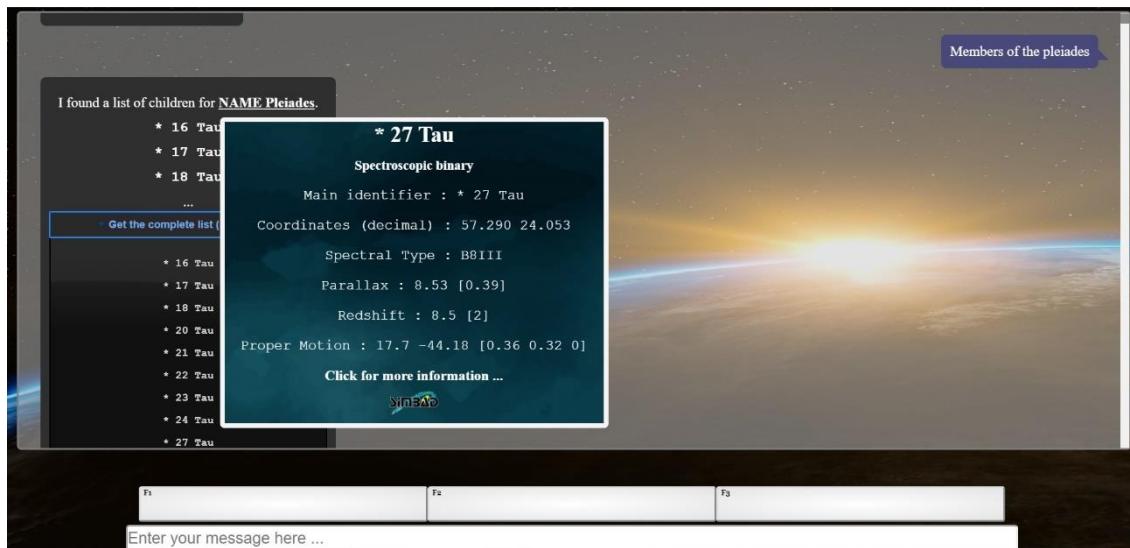


Figure 13 - Interface sur PC avec fenêtre de prévisualisation

RECUPERER DES DONNEES SUR LES OBJETS DU CIEL

Il s'agit de la fonctionnalité principale exprimée dans une grosse partie des phrases fournies au début de mon stage par l'équipe du CDS ([annexe 2](#)). Le principe de cette intention est de récupérer les valeurs associées à une mesure sur un objet astronomique enregistré dans les bases Simbad et VizieR. Quand l'utilisateur demande une valeur, elle est par défaut recherchée sur la base Simbad, pour les raisons évoquées dans la partie de [présentation des outils du CDS](#). Si aucun résultat n'est trouvé, une recherche identique sera menée sur VizieR. Cependant, si l'utilisateur le souhaite, il peut demander explicitement au *chatbot* d'effectuer la recherche sur la deuxième base. Dans sa demande, il peut préciser plusieurs mesures dans une même phrase et les demander sur un nom ou sur un type d'objet sans problème d'ambiguïté. Pour éviter la frustration de l'utilisateur et quelques problèmes inutiles, un système est mis en place pour limiter l'impact des fautes d'orthographe ou de frappe. Celui-ci utilise le concept de la distance de Levenshtein entre deux mots. Il s'agit simplement du nombre de caractère qu'il faudrait ajouter, supprimer ou modifier pour passer d'un mot à un autre. Par exemple la distance de Levenshtein entre « mot » et « moi » vaut 1, puisqu'il suffit de modifier la dernière lettre de « mot » pour passer à « moi ». Cela permet d'obtenir des résultats même si l'utilisateur se trompe ou écrit trop vite. Ce système n'est cependant pas appliqué sur les noms ou types d'objets, puisque plusieurs étoiles peuvent avoir des désignations très proches les unes des autres et que cela créerait des dysfonctionnements. Concrètement, cette intention est donc un moyen alternatif de récupérer les données stockées au CDS, utilisant le langage naturel plutôt qu'un langage informatique de requêtage classique comme le **SQL** ou **l'ADQL** en astronomie.

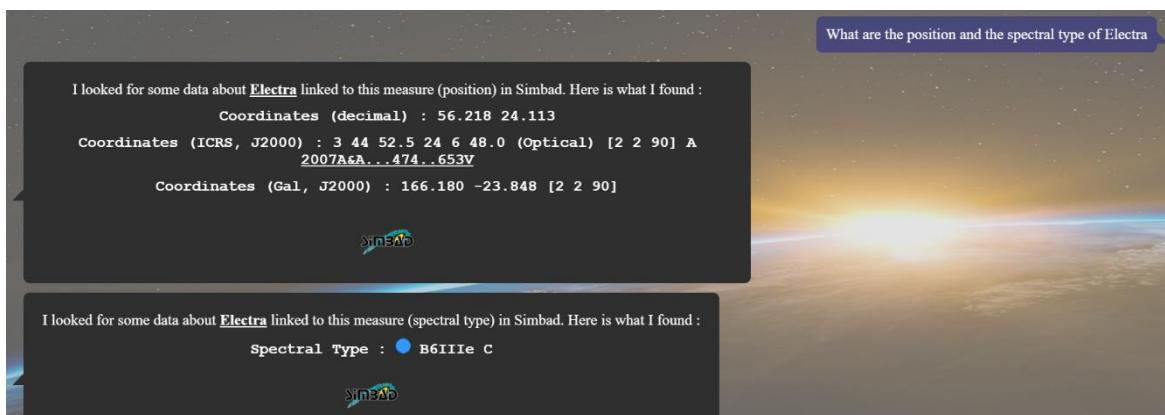


Figure 14 - Récupérer la valeur d'une mesure pour un objet

Concernant la récupération des données, d'autres intentions liées à celle-ci ont été ajoutées pour récupérer des liens hiérarchiques entre les objets. Pour comprendre ce concept, je vais prendre un exemple proche de nous. La Terre fait partie de la Voie Lactée, la planète est donc hiérarchiquement une enfant de la galaxie. De manière réciproque, la Voie Lactée est parent de la Terre. Même principe avec les autres planètes de notre galaxie. Hiérarchiquement, la Terre et Mars sont frères et sœurs. Le chatbot permet de récupérer une liste d'objets astronomiques contenus dans un autre. On peut même préciser le type d'objet qui nous intéresse et ainsi connaître indépendamment les planètes et les étoiles contenues dans une galaxie, ou réciproquement le nom de la galaxie ou de la nébuleuse dans laquelle se trouvent l'objet que l'astronome étudie. Les résultats d'une requête de ce type sont rangés dans une liste et tous proposent l'option de prévisualisation. Ainsi, il est très facile et très rapide de comparer deux objets d'une même galaxie par exemple.



Figure 15 - Connaitre les planètes membres d'une galaxie

FACILITER L'UTILISATION DES SERVICES DU CDS

Lors d'un travail de recherche, les astronomes peuvent être intéressés par le fait d'obtenir une liste de catalogues à étudier sur leur sujet de recherche. En effet, il en existe aujourd'hui un peu plus de 16000 répertoriés dans VizieR et pouvoir avoir accès en un clic à ceux faisant mention des paramètres qui s'imposent pouvait être un confort supplémentaire non négligeable pour eux. Une fonctionnalité a donc été ajoutée pour leur permettre faire cela. Ils peuvent dans ce type de demande préciser le nom générique d'un catalogue, une ou plusieurs mesures étudiées ainsi qu'un nom ou un type d'objet particulier, tout en précisant s'ils souhaitent obtenir une liste triée par ordre de popularité ou par ordre chronologique.

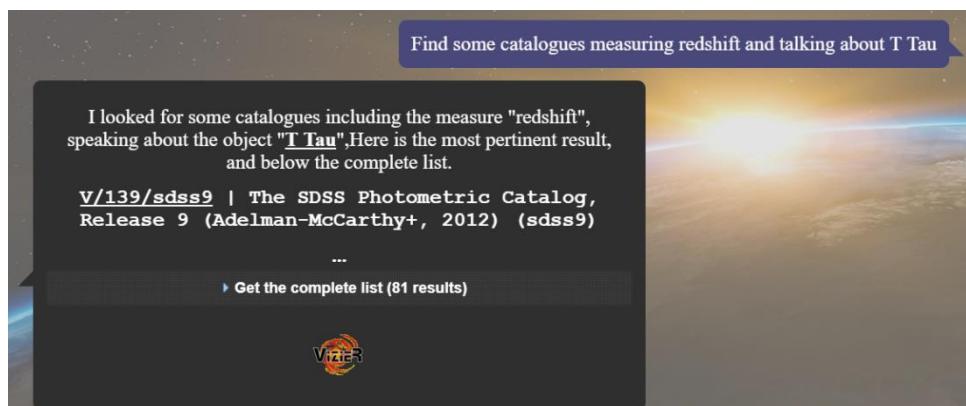


Figure 16 - En cliquant sur le nom, l'utilisateur est directement renvoyé vers la page VizieR du catalogue

VISUALISER ET DECOUVRIR LE CIEL

Dernière fonctionnalité implémentée et sûrement la plus visuelle et intéressante pour le commun des mortels, le fait de pouvoir visualiser un objet astronomique directement depuis l'interface du chatbot. A l'aide d'une simple demande, on peut faire en sorte que l'outil nous montre une image d'un objet précis, en donnant son nom ou d'un type d'objet en général (une étoile, une nébuleuse, une galaxie, ...). Il est également possible d'ajouter des filtres pour ne garder qu'une longueur d'onde précise à observer. Rapide rappel vulgarisé de physique, la longueur d'onde est une mesure de distance utilisée pour représenter l'écart entre deux « bosses » d'une onde. Grâce à cette mesure, on peut déterminer si une onde est infrarouge, ultraviolette, radio, ... La lumière, qui est pour simplifier une onde un peu spéciale, possède un type défini de cette manière-là. L'œil humain est capable de détecter des longueurs d'onde allant du bleu, à environ 400nm au rouge à environ 800nm. On appelle cette plage le domaine du visible mais d'autres longueurs d'onde en dehors de cet intervalle existent et font partie du domaine de l'invisible à l'œil nu. Ainsi, dans l'espace, tous les objets n'émettent pas que de la lumière visible. En appliquant ces filtres, les infrarouges, ultraviolets et autres ondes invisibles vont être convertis en lumière visible et ainsi afficher l'objet d'une manière complètement différente. Ils permettront également d'observer certaines étoiles qui n'émettent que des ondes dans le domaine de l'invisible. Cette fonctionnalité est possible grâce au service Aladin-Lite proposé par le CDS. En effet, la version en ligne d'Aladin propose un widget pouvant être inclus dans une page web de la même façon que les petites cartes Google Maps que vous avez peut-être déjà pu observer sur certains sites.

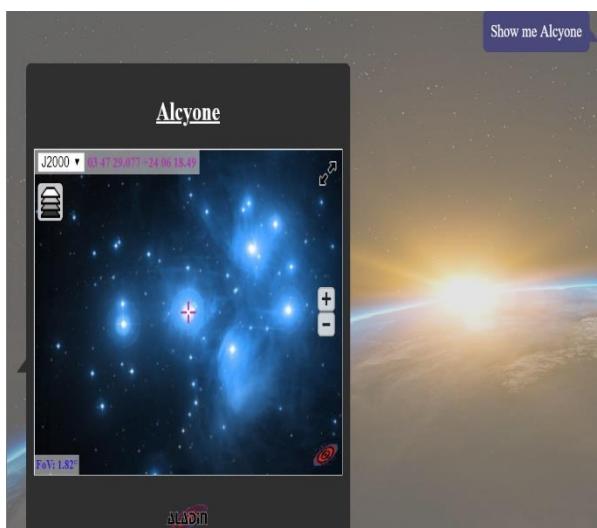


Figure 18 - Afficher une étoile

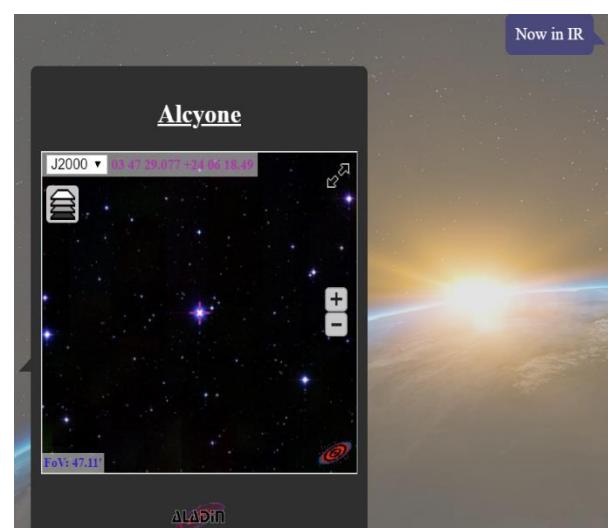


Figure 17 - Cette étoile en infrarouges

Dans le domaine de l'image également, il est possible d'utiliser ces fenêtres Aladin-Lite pour se déplacer dans le ciel. Si l'utilisateur tombe sur quelque chose qui l'intéresse et veut savoir de quel objet il s'agit, il lui suffit de le demander.

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

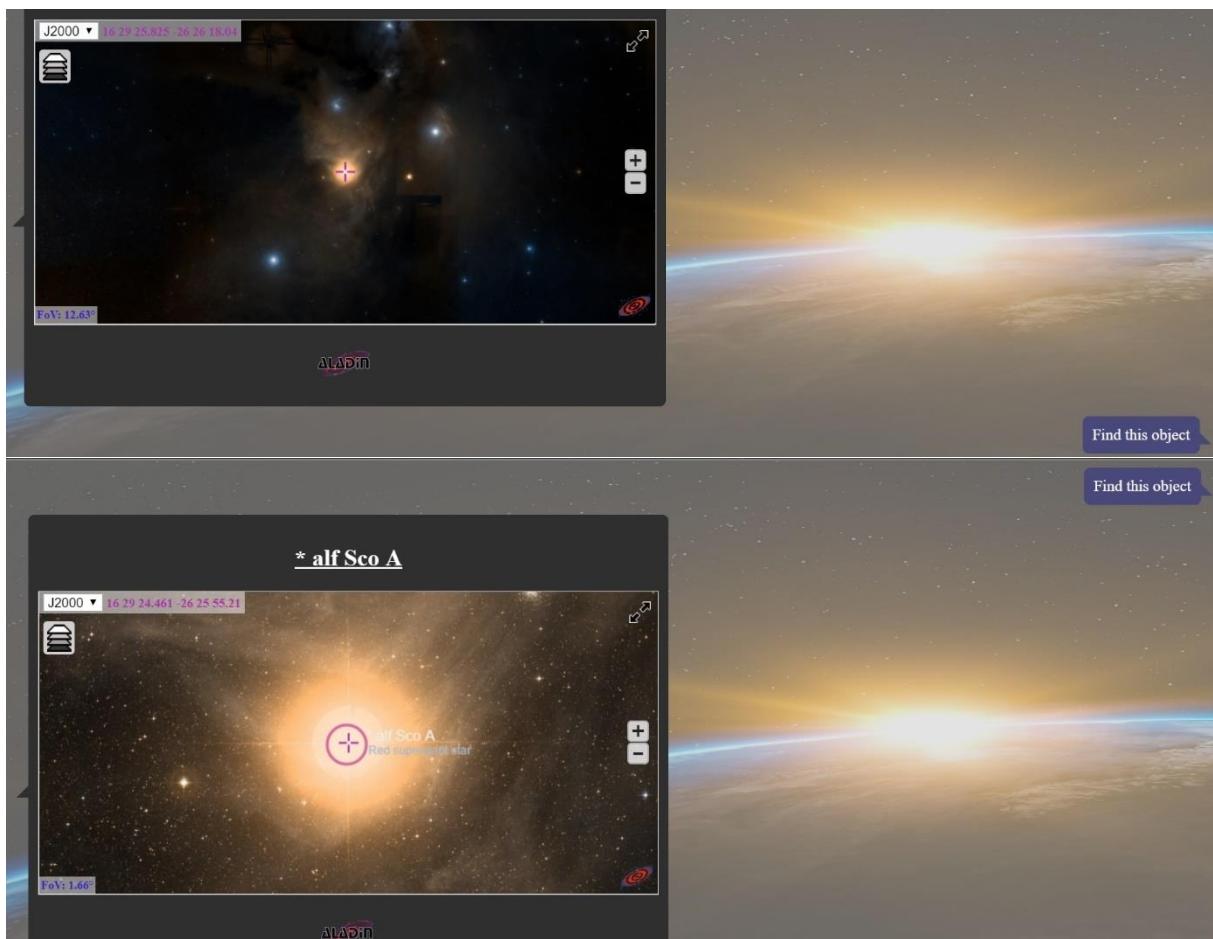


Figure 20 - L'objet est recherché et son nom ainsi que son type sont indiqués

Enfin, la dernière fonctionnalité en rapport avec la visualisation du ciel consiste à rechercher des objets astronomiques autour d'un autre. Le rayon de recherche peut être précisé ou non par l'utilisateur et les objets trouvés sont à la fois cliquables directement sur l'image, ouvrant ainsi un petit tableau récapitulatif des données disponibles, et répertoriés dans une liste où passer la souris sur leur nom offre une prévisualisation. Si l'utilisateur recherche un type d'objet en particulier, par exemple seulement les étoiles, il peut le préciser et la liste ne comportera que les objets de ce type.

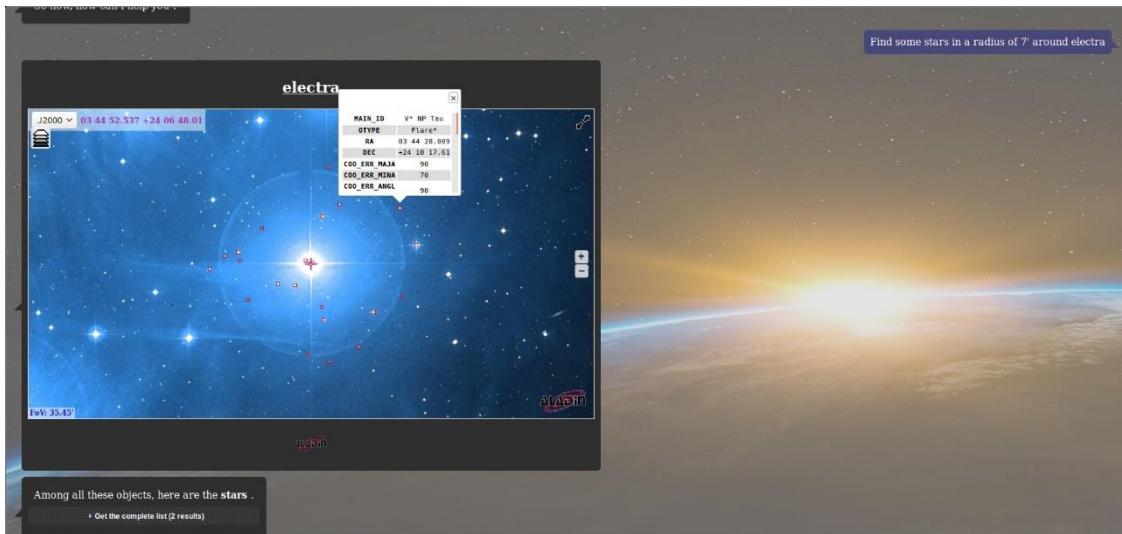
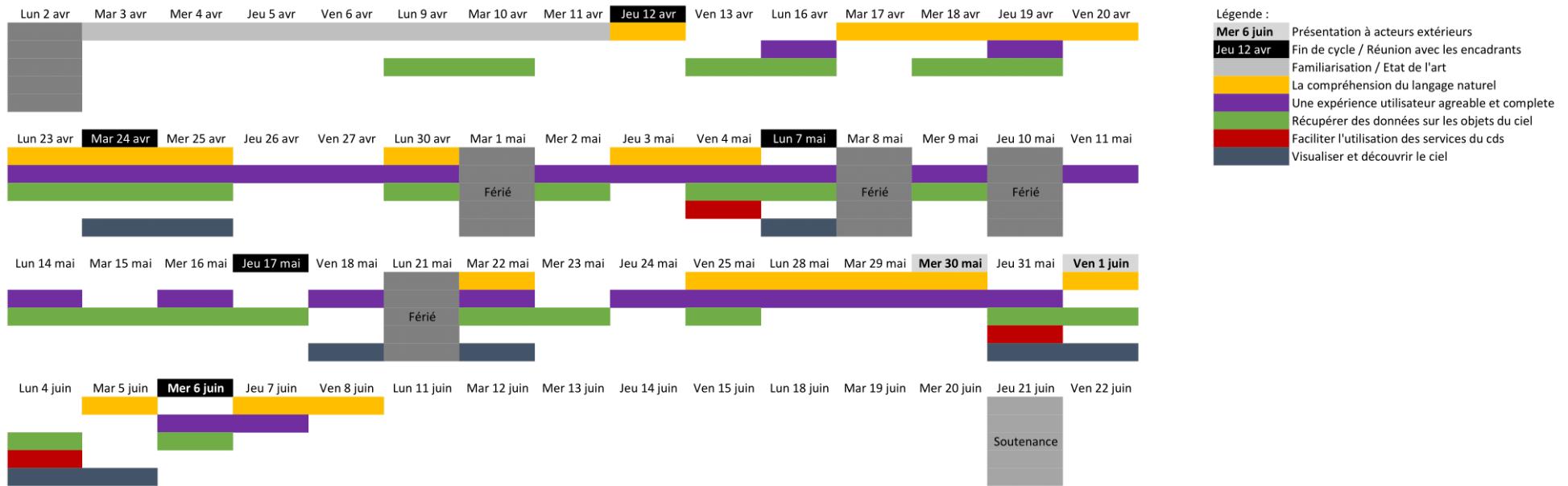


Figure 19 - Trouver des objets autour d'un autre

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

C. PLANNING DE REALISATION



Chaque couleur correspond à une des fonctionnalités décrites dans la partie précédente. Aucune indication n'est donnée à partir du 11 juin parce que ce planning a été réalisé le 8 juin 2018.

2. LES DIFFICULTES RENCONTREES

Au cours de ce projet, j'ai pu remarquer que la principale difficulté résidait dans la compréhension de mon environnement et de mes interlocuteurs. En effet, l'astronomie est un domaine des sciences très pointu dans lequel je n'avais que très peu de connaissances au début du stage. Il a ainsi fallu apprendre très vite certains concepts essentiels pour pouvoir communiquer avec les astronomes du service et cerner correctement les enjeux de certaines fonctionnalités pour eux. En effet, mon travail principal tout le long de ce stage a été d'apprendre à une machine à comprendre le langage humain. Or, tout le monde sait à quel point il est difficile de transmettre des connaissances que nous-mêmes ne maîtrisons pas. Il était par conséquent inenvisageable de ne pas s'imprégnier de ces connaissances, au risque de ne pas comprendre certaines demandes qui m'étaient faites ou de fournir un outil qui ne comprendrait pas ce qui lui est demandé. Pour remédier à cela, j'ai lu de nombreux articles sur Internet autour de concepts de la matière qui m'étaient inconnus, j'ai posé des questions, que je préparais à l'avance pour éviter de les déranger le reste du temps, à mes encadrants quand je les croisais et j'ai assisté aux séminaires organisés par l'établissement le vendredi en fin de matinée pour présenter des travaux de recherche en astronomie.

Mais outre le langage astronomique, il a également fallu comprendre vite et bien comment fonctionnaient les différents services proposés par le **CDS** et discerner quels étaient leurs avantages et leurs inconvénients. Dès le début du stage, il a été de fait assez important d'être impliqué et d'écouter attentivement les indications et présentations de mes encadrants pour ne pas être perdu. J'ai également dû faire preuve de beaucoup d'autonomie en allant faire des essais de mon côté pour mieux comprendre la structure des deux bases de données, la façon de les interroger et le format des données à la sortie.

Enfin, je vais conclure cette partie en présentant le seul problème informatique rencontré assez important pour mériter son paragraphe dans ce rapport, à savoir le système entourant la récupération des données sur la base VizieR. J'ai déjà pu aborder lors de la [présentation du service](#) la raison même de ce problème, à savoir la structure de la base. En effet, celle-ci s'avère être constituée de tables, comme toutes les bases de données, mais qui représentent en réalité des publications scientifiques. Or, pour faciliter l'accès aux données et leur conservation, ces publications sont elles-mêmes découpées en tables. VizieR est donc une base de données contenant elle-même d'autres bases de données. Sauf que la particularité est que la structure des catalogues peut varier de l'un à l'autre. Pour tout de même garantir une certaine cohérence entre les tables des différentes, un système de métadonnées a été mis en place, pour décrire de la même manière tout le contenu de VizieR. Ainsi, deux colonnes situées dans deux catalogues différents peuvent mesurer la même mesure, prenons ici la température, mais l'une pourrait s'appeler « Temp » alors que l'autre « Tempe. ». Le seul moyen de comprendre qu'il s'agit de la même mesure dans les deux cas est de regarder les métadonnées des colonnes et de remarquer que celles-ci décrivent la température d'un objet astronomique. Ceci veut aussi dire que deux publications peuvent totalement parler du même objet mais être dissociées. Autrement dit, un certain nombre de catalogues peuvent être choisis pour un même objet. Après avoir longuement cherché en long et en large un moyen efficace de récupérer les données sur VizieR, j'ai demandé de l'aide à Thomas BOCH et Sébastien DERRIERE, mes deux co-encadrants, afin de mettre en place une stratégie. La solution était donc de récupérer les coordonnées dans le ciel de l'objet recherché ainsi que les métadonnées représentant les mesures demandées. Avec ces informations, il fallait ensuite récupérer la liste des catalogues dont le contenu décrit par les métadonnées faisait mention des mesures recherchées dans la bonne région du ciel. Après avoir isolé le catalogue le plus visité sur le service et le catalogue le plus récent, afin d'obtenir des résultats les plus pertinents possibles, l'étape d'après était de récupérer la structure de ceux-ci grâce à une table spécialement construite à cet effet dans VizieR. Enfin, une fois les noms des colonnes identifiés, il suffisait de questionner la base grâce à l'outil prévu à cet effet, en lui fournissant une requête ADQL correcte. Une solution pas simple mais qui a le mérite d'avoir nécessité un gros travail de réflexion et de maîtrise des différentes technologies du service. C'est aussi pour cette raison que récupérer une donnée sur VizieR est un processus plus long et donc considéré uniquement en cas de demande explicite de l'utilisateur ou en cas d'échec sur Simbad.

Outre son intérêt en tant que casse-tête, j'ai trouvé intéressant de présenter VizieR et des difficultés rencontrées lors de l'accès à ses données afin d'illustrer une dernière fois le fait que la principale difficulté de ce stage a été la compréhension de l'environnement, des outils existants et des « clients », à savoir mon maître de stage et les autres employés de l'observatoire. Techniquement parlant, les connaissances acquises au long de mon DUT ont été suffisantes pour effectuer un grand nombre de tâches. Lorsque je ne savais pas faire quelque chose, j'allais juste lire attentivement la documentation du langage ou de la bibliothèque sur Internet et je trouvais plus ou moins rapidement une solution à mon problème. C'est en grande partie pour cette raison que j'ai si peu de vrais problèmes techniques à décrire dans cette partie. Mon stage m'a permis de me rendre compte que mon principal obstacle dans ma future vie professionnelle n'allait pas forcément être de l'ordre technique, mais certainement plus de l'ordre humain et de l'ordre de la compréhension du milieu et des acteurs pour lesquels je serai amené à travailler.

III. CONCLUSION

1. LES SUGGESTIONS POUR L'ENTREPRISE

En prenant un peu de recul par rapport à tout ce que j'avais pu faire pendant mon stage, j'ai pu discerner quelques points qui pourraient être intéressants à travailler ou à modifier afin de prolonger le travail sur le *chatbot*.

Tout d'abord, je pense qu'il pourrait être très bénéfique à plus ou moins court terme de changer le moteur de traitement du langage naturel, Dialogflow. En effet, comme vous aurez pu le remarquer en parcourant ce rapport, l'outil a été développé, est maintenu et est hébergé par Google. Même si cela ne posait pas de problème particulier à mon maître de stage, utiliser un tel outil contraint énormément le **CDS**. Dans un premier temps, si l'outil venait à ne plus marcher pour une quelconque raison, tout le *chatbot* deviendrait inutilisable. Ensuite, si un mauvais fonctionnement de l'assistant était dû à Dialogflow, le service n'aurait pas le pouvoir de corriger le problème, juste de le reporter aux équipes de Google et attendre qu'ils s'en occupent. Enfin, si un jour l'entreprise venait à décider de ne plus offrir gratuitement les fonctionnalités de son moteur de NLU, c'est tout le *chatbot* qui s'en verrait affecté. A la vue de ces problèmes assez majeurs, on pourrait alors se demander pourquoi avoir choisi d'utiliser Dialogflow dans un premier temps. D'abord, il faut savoir que j'ai quand même eu ce raisonnement dès le moment où il a fallu faire un choix et que j'ai pu en discuter à plusieurs reprises avec M. Schaaff. Cependant, il faut aussi garder à l'esprit que mon stage était avant tout une mission de découverte et d'investigations pour savoir ce qu'il était possible de faire pour le **CDS** avec du langage naturel. Finalement, peu importe le moteur de NLU choisi, ils feront tous la même chose dans les grandes lignes, à savoir prendre en entrée un message de l'utilisateur et renvoyer en sortie une analyse de celle-ci. Ceci veut ainsi dire que même si celui-ci venait à être changé par la suite, tout le travail de recherche effectué autour de ce domaine, toutes les connexions avec les outils du **CDS**, etc., seront toujours d'actualité et mon stage aura quand même atteint son objectif. Dans cette optique-là, j'ai également organisé mon code de sorte à le rendre un maximum fermé à la modification. Pour ce faire, j'ai créé un ensemble de fonctions dont le seul but était de découper les résultats renvoyés par le moteur de NLU. Ainsi, si un jour celui-ci venait à être changé au profit d'un autre créé de toute pièce par le service par exemple, seules quelques lignes de code auraient à être changées.

Dans un second temps, je conseillerais aux équipes de l'observatoire d'affiner mon système de récupération des données sur VizieR. En effet, pour rechercher un catalogue parlant d'un objet dont l'utilisateur a précisé le nom dans cette base, il faut passer par sa position dans le ciel. Et à cause de cette étape, plusieurs imprécisions peuvent venir fausser les résultats. Tout d'abord, parce que la plupart des objets ne restent pas fixes mais se déplacent lentement dans le ciel. En fonction de la date à laquelle l'objet a été entré dans la base, celui-ci a totalement pu légèrement bouger et sa position aujourd'hui peut correspondre à celle d'un tout autre objet positionné à cet endroit par le passé, ce qui fausse complètement les résultats. Ensuite, parce que tous les objets astronomiques ne font pas la même taille et que si le rayon de recherche

précisé autour des coordonnées est trop grand, les mesures d'autres objets pourraient interférer. Avec le système mis en place actuellement, les gros objets répertoriés ne devraient pas poser de problème mais pour des demandes plus précises, les résultats des requêtes à ce service pourraient être faux. J'ai décidé de mentionner ce problème en tant que suggestion mais il n'est pas impossible que d'ici la fin du stage celui-ci soit corrigé. En effet, au moment où j'écris ces lignes il me reste encore deux semaines de stage. Cependant, même si c'est une possibilité, ce n'est pas une certitude car d'autres finitions plus importantes devront être implémentées avant de travailler sur ce point.

En termes de prolongations cette fois-ci, je pense qu'il serait pertinent de coupler le **chatbot** avec un système de reconnaissance vocale. En effet, il faut garder à l'esprit qu'un des objectifs principaux de cet outil est de faciliter et d'accélérer l'accès aux différents services du **CDS**. Or, même si pour certaines fonctionnalités il est évident que l'assistant est plus rapide qu'une recherche « manuelle » directement dans le service, pour d'autres il est parfois plus difficile d'être aussi catégorique. Par exemple, pour obtenir une mesure dans la base Simbad. A travers un champ de formulaire, il suffit juste de rentrer le nom d'un objet pour obtenir tout un tas d'informations et de mesures. Dans ce cas, un astronome va-t-il aller sur le **chatbot** pour écrire « Position of Sirius » ou va-t-il directement aller sur Simbad taper « Sirius » et chercher parmi la liste des mesures proposées ? Evidemment cela dépendra des préférences de chacun, mais pas seulement. On peut en effet penser que la rapidité d'accès à l'information va énormément jouer. Or, le système actuel ne joue pas forcément en la faveur du **chatbot**, puisque pour la même information il faudra taper au moins deux mots de plus. Cependant, avec un système de reconnaissance vocale, l'expérience utilisateur se retrouve positivement impactée : plus besoin d'écrire quoi que ce soit, il suffit juste de le dire puis l'assistant prend le relais jusqu'à ce qu'il trouve l'information, ce qui à l'échelle humaine arrive très vite. Après avoir pu travailler sur une fonctionnalité de ce genre durant mon projet tutoré du quatrième semestre, je peux affirmer que l'utilisation de l'outil devient dans tous les cas bien plus fluide et l'interactivité proposée donne encore plus envie de l'essayer.

Ces idées d'améliorations mises à part, le travail effectué tout au long du stage est fonctionnel et a été testé par moi-même mais aussi par d'autres employés du **CDS**. En toute logique, les fonctionnalités implémentées ne devraient pas nécessiter de modifications même si de nouvelles venaient à être ajoutées dans le futur.

2. LES LEÇONS TIREES DE CE TRAVAIL

Ce stage à l'observatoire de Strasbourg m'a été bénéfique sur de nombreux points. Tout d'abord d'un point de vue personnel, il représente un moment important de ma vie puisqu'il s'agit de ma première vraie expérience professionnelle, mes interactions avec le monde du travail se résumant avant cela à quelques missions de courtes durées l'été pour se faire de l'argent de poche. Et sur ce point je suis plus que satisfait puisque j'ai été accueilli chaleureusement et avec considération. L'ambiance de travail à l'observatoire était excellente et donnait vraiment envie de se lever chaque matin pour aller travailler. Comme conclu lors de la partie sur les [difficultés rencontrées](#), j'ai également pu me rendre compte d'où venait la difficulté principale lors de projets informatiques professionnels, ce qui me servira forcément de nouveau un jour.

Mais c'est surtout d'un point de vue professionnel que le stage est pour moi une réussite. D'abord, parce qu'il m'a permis d'approfondir pendant douze semaines mes connaissances en développement web et notamment en développement Javascript, un langage plutôt populaire que je maîtrise aujourd'hui bien mieux qu'au début du stage. Ensuite, parce que participer à cette mission gravitant autour du langage naturel m'a permis d'enchaîner un deuxième gros projet dans le domaine, le premier étant celui commencé en tant que projet tutoré au troisième semestre et approfondi pendant celui du quatrième, me permettant ainsi de pouvoir mentionner une expérience assez conséquente pour une personne de mon niveau d'étude dans ce domaine de l'intelligence artificielle. Réaliser ce stage m'a également poussé à m'intéresser beaucoup plus à la théorie derrière les outils et ainsi apprendre et comprendre quelques bases de ce domaine de l'informatique comme

le *machine learning* par exemple. Ceci est donc une première approche intéressante pour la suite de mes études. Et d'ailleurs en parlant de cela, j'aurais difficilement pu faire plus pertinent envers mon projet professionnel que de travailler pour un service recherche et développement, mis à part peut-être directement travailler dans un laboratoire, puisque j'envisage actuellement de devenir enseignant-chercheur plus tard. J'ai donc extrêmement apprécié passer ces douze semaines dans cet environnement, à pouvoir participer chaque semaine à la présentation de sujets de recherche lors de séminaires ou tout simplement en participant à un projet qui tout le long s'est voulu expérimental et orienté recherche.

Pour résumer, le stage à l'observatoire restera pour moi une expérience très enrichissante qui m'aura permis de confirmer mes envies de continuer vers la voie de la recherche. J'ai également pu prendre un peu plus confiance en moi et en mes capacités et il m'aura apporté quelques éléments intéressants pour me démarquer dans mon futur professionnel, comme le fait que mon travail ait été présenté à une communauté scientifique mondiale par exemple, ce qui n'est pas rien. C'est pour toutes ces raisons que je tiens à conclure mon rapport en remerciant de nouveau toute l'équipe de l'observatoire astronomique de Strasbourg.

LEXIQUE

ACRONYMES

- **ADQL** : Astronomical Data Query Language – Version améliorée du SQL prenant en charge certaines fonctionnalités propres en astronomie, notamment la gestion des points dans le ciel, du calcul de la distance, ...
- **CDS** : Centres des Données astronomiques de Strasbourg
- **CNRS** : Centre National de la Recherche Scientifique – Plus grand organisme public français de recherche scientifique.
- **GALHECOS** : Galaxies, High Energy, Cosmology, Compact Objects & Stars – Equipe de recherche astronomique de l'observatoire de Strasbourg.
- **IVOA** : International Virtual Observatory Alliance – L'alliance internationale pour l'observatoire virtuel, une organisation scientifique mondiale visant à organiser et faciliter la virtualisation des données astronomiques.
- **JSON** : JavaScript Object Notation – Format utilisé pour structurer les données. Comme son nom l'indique, très facile de manipulation avec le langage Javascript.
- **NLU** : Natural language Understanding – Traitement du langage naturel
- **SQL** : Structured Query Language – Langage informatique permettant de récupérer grâce à des requêtes des données dans une base relationnelle, c'est-à-dire une base constituée de tableaux de données reliés entre eux par des clés d'identification.
- **XML** : eXtreme Markup Language – Autre format utilisé pour structurer les données, utilisant des balises comme le langage HTML.

DEFINITIONS

- **Chatbot** : En français « agent conversationnel », un chatbot est un programme informatique capable de tenir une conversation avec un être humain dans un langage qui lui est naturel.
- **Langage naturel** : Langue parlée par des êtres humains, par opposition au langage formel, autre désignation du langage informatique.
- **Machine learning** : Le concept général du machine learning est d'apprendre à une intelligence artificielle à évoluer au cours de son utilisation et de son expérience avec l'utilisateur. Cette adaptation au contexte se base sur une analyse de données pouvant provenir d'une base de données ou de capteurs. Il existe plusieurs types d'apprentissage par machine learning, dont deux principaux : l'apprentissage supervisé, qui nécessite l'action d'un « expert » (la personne qui apprend les connaissances à l'IA), et l'apprentissage non supervisé, où l'IA classe et « comprend » toutes les données de base qu'il possède de lui-même. Dialogflow se base sur un apprentissage supervisé.
- **Matching** : « Correspondance » en français, il s'agit du fait d'associer un mot-clé ou un groupe de mots à une idée.

BIBLIOGRAPHIE

Qu'est-ce qu'un chatbot ? : [article Figaro](#)

Google Duplex : [article sciences et avenir](#)

Documentation Dialogflow : <https://dialogflow.com/docs/machine-learning>

Documentation Rasa NLU (traitement du langage naturel) : <https://nlu.rasa.com/>

Documentation Rasa Core (machine learning) : <https://core.rasa.com/>

Documentation Wit.ai : <https://wit.ai/docs>

Documentation Luis.ai : <https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/Home>

Site **CDS** : <http://cdsweb.u-strasbg.fr/index-fr.gml>

Portail **CDS** : <http://cdsportal.u-strasbg.fr/>

Portail Simbad : <http://simbad.u-strasbg.fr/>

Portail VizieR : <http://vizier.u-strasbg.fr/>

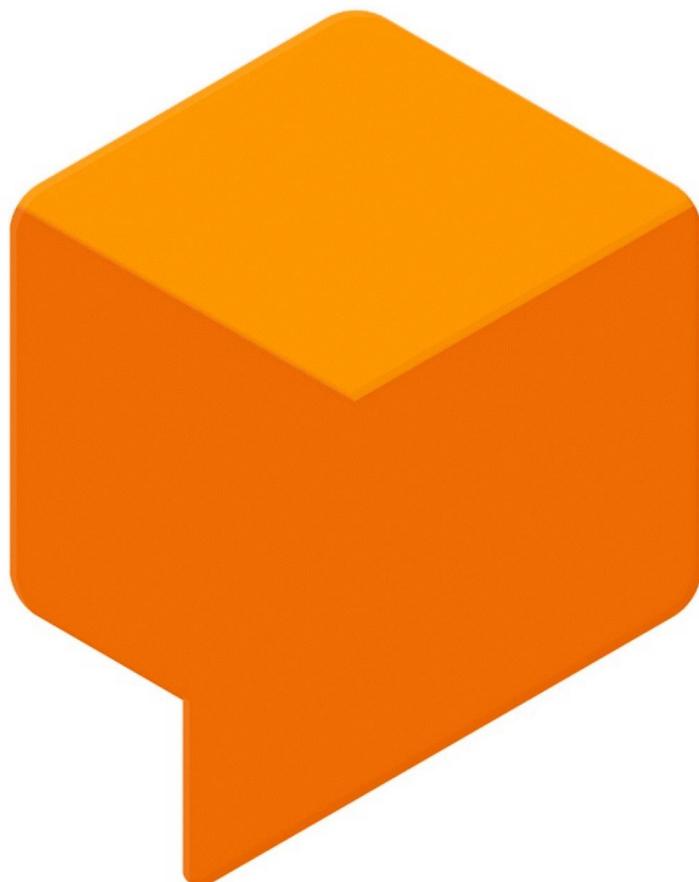
Portail Aladin-Lite : <http://aladin.u-strasbg.fr/>

Documentation jQuery : <http://api.jquery.com/>

Machine learning dans le langage naturel : <https://arxiv.org/pdf/1606.01541.pdf>

ANNEXES

ANNEXE 1 – DIALOGFLOW



DIALOGFLOW

LE MACHINE LEARNING AU SERVICE DU TRAITEMENT DU
LANGAGE NATUREL

Dialogflow – Alexis Guyot

I. INTRODUCTION

Dialogflow est une plate-forme de traitement du langage naturel possédée par Google. A l'origine, l'entreprise derrière l'outil s'appelait Speaktoit, était indépendante, a été créée en 2010. Elle s'est faite remarquer en créant l'Assistant en 2011, soit la même année que la création de Siri, un assistant artificiel disponible sur toutes les plateformes. En 2014, l'entreprise a mis à disposition du monde le moteur derrière leur assistant, et l'a appelé api.ai. En 2016, Google rachète Speaktoit pour travailler sur leur assistant. La plateforme de NLU¹ est renommée en octobre 2017 « Dialogflow ».

Dialogflow offre un ensemble de fonctionnalités en rapport avec le traitement du langage naturel. Celles-ci sont accompagnées et renforcées par un algorithme de machine learning spécifique à chaque projet (comprenez ici que l'algorithme s'adapte à l'expérience gagnée par le bot au cours de sa vie). Le moteur derrière Dialogflow est bien évidemment hébergé chez Google, et la communication avec celui-ci se fait par requêtes HTTP. Nous allons par la suite détailler ces fonctionnalités, et voir en quoi celles-ci s'avèrent intéressantes dans le cadre de notre projet.

II. LES FONCTIONNALITÉS PROPOSÉES

1. Intentions/Entités

Comme pratiquement tout moteur de traitement du langage naturel, Dialogflow propose une compréhension du langage basée sur un système d'intentions et d'entités. L'intention, comme son nom le suggère bien, est l'attente qui a poussé l'utilisateur à formuler sa requête à l'oral. Par exemple, l'intention cachée derrière une phrase du type « Quelle est la température effective de Sirius ? » est d'obtenir une mesure concernant un objet. Les entités correspondent aux éléments clés de la demande, ceux qui auront une importance pour formuler une réponse pertinente et cohérente. Dans notre précédent exemple, on discerne deux entités : « température effective » qui correspond à la mesure attendue, et « Sirius » qui est le nom de l'objet à étudier. Ces deux termes sont indispensables pour la formulation d'une réponse. Dialogflow décrit donc une phrase en catégorisant son intention, et en mettant de côté ses entités. Après traitement de la phrase d'exemple, le bot gardera seulement à l'esprit que la personne qui lui parle veut obtenir une mesure, la température effective, d'un objet qui s'appelle Sirius. Il stockera ensuite ce constat dans une structure JSON qu'il retournera à l'utilisateur.

Afin d'effectuer ce travail de caractérisation d'une phrase, Dialogflow utilise une combinaison de deux méthodes de NLU : le matching et le machine learning. Le matching est la plus vieille méthode connue de traitement du langage, et est historiquement la première utilisée, dès le premier chatbot ELIZA créé en 1966. Une liste de mots-clés est pré-enregistrée par le développeur, et le bot va vérifier pour chaque phrase entrée par l'utilisateur si celle-ci contient un des mots qu'il connaît. Chaque mot-clé est également lié à une ou plusieurs réponses, et le bot se

¹ NLU : Natural Language Understanding (Compréhension/Traitement du langage naturel)

Dialogflow – Alexis Guyot

contente d'afficher ces réponses pré-enregistrées lorsqu'il en reconnaît un. Le machine learning est une méthode beaucoup plus récente qui consiste à faire évoluer l'intelligence artificielle d'un programme au fur et à mesure de son utilisation en le faisant apprendre de ses expériences passées. Ainsi, on commence au départ l'apprentissage en montrant au programme les bons comportements face à problème, pour lui donner des bases. Après cela, le programme va essayer de déterminer un schéma dans les bons comportements montrés par le développeur et essayer de le suivre. Pour chaque décision prise par le programme, le développeur peut ensuite donner son avis en précisant si la réponse était pertinente, bonne mais pas complète ou hors sujet. De cette manière, la machine apprend de ses erreurs et de ses expériences. Cette méthode devient particulièrement efficace après un grand nombre d'itérations. Dialogflow utilise ces deux méthodes. Au départ, il demande au développeur de lui montrer comment reconnaître une intention et ses entités sur quelques exemples. A partir de cela, il va commencer le processus de machine learning. Afin de minimiser son taux d'erreur, il va également essayer de reconnaître des mots souvent utilisés dans les exemples donnés par le développeur, jusqu'à ce que son algorithme de machine learning soit assez performant pour ne plus avoir à baser une grosse partie de son jugement sur le matching. La plateforme en ligne de Dialogflow permet d'effectuer l'étape dite « d'entraînement » durant laquelle le développeur donne son avis sur les réponses de l'intelligence artificielle.

A partir de là, il sera possible de se servir de ces informations pour aller chercher les informations attendues dans une base de données ou de connaissances. Pour ce projet, il commencera à chercher dans la base Simbad gérée par le CDS dans un premier temps, puisque trouver une information dans celle-ci est très rapide à travers une requête TAP. S'il ne trouve pas l'information voulue, il cherchera dans VizieR, un processus beaucoup plus long (et donc pas prioritaire). Il faut savoir que Dialogflow propose un système de « fulfillment » (accomplissement en anglais) exécutant cette recherche à travers un script Javascript. Cependant, j'ai pris la décision de gérer moi-même cette partie avec un programme Javascript, afin de centraliser les fonctionnalités du chatbot. Dialogflow n'est alors utilisé que pour comprendre la requête de l'utilisateur.

2. Sauvegarde et utilisation du contexte

L'un des plus gros intérêts de Dialogflow par rapport à d'autres moteurs de traitement du langage naturel (comme Rasa ou Wit.ai par exemple) est qu'il propose de gérer un système de contexte. Le contexte est tout simplement une mémoire accordée au programme, dans laquelle il va pouvoir stocker les valeurs des entités identifiées lors des requêtes précédentes pendant un certain temps (en minutes ou en nombre de requêtes, par défaut 20 min ou 5 requêtes). Ce point est extrêmement important lorsque l'on parle de langage naturel. Reprenons l'exemple de la partie précédente : « Quelle est la température effective de Sirius ? ». Après une telle demande, un astronome pourrait avoir envie d'obtenir une autre donnée sur Sirius. En langage naturel, il aurait tout simplement dit « Et ses coordonnées ? ». En tant qu'êtres humains, on comprend facilement que l'utilisateur parle toujours de Sirius. Mais sans contexte, la machine ne peut pas le savoir.

Dialogflow – Alexis Guyot

Dialogflow, avec son système de contexte et de mémoire, le peut et permet ainsi de rendre plus naturelle une conversation avec l'agent conversationnel.

3. Suites logiques d'intentions (stories)

Avec Dialogflow, il est également possible de définir des suites logiques d'intentions. Cette fonctionnalité peut s'avérer utile dans les cas où le chatbot a forcément besoin d'une information précise pour pouvoir fonctionner et que l'utilisateur ne la renseigne pas lors de son premier message. Par exemple, si le bot perd le contexte parce que le temps est écoulé et que l'utilisateur, ne s'en rendant pas compte, demande « Et son type spectral ? », alors celui-ci lui posera une question du type « De quel objet parlez-vous ? » plutôt que de remonter une erreur. Dans ce cas, il sera également capable de savoir que l'intention suivant un manque de contexte est de combler ce manque, et que l'entité donnée par l'utilisateur devra être couplée avec celle qu'il a en mémoire pour répondre à l'intention de départ. D'autres idées d'implémentation peuvent utiliser cette fonctionnalité. On peut par exemple imaginer que le chatbot suggère une autre intention en rapport avec celle demandée juste avant par l'utilisateur pour continuer une sorte de dialogue plutôt que de juste répondre et se taire ensuite.

4. Small Talk

Autre fonctionnalité un peu plus superficielle mais appréciable offerte par Dialogflow, la gestion des « Small Talk ». Il s'agit d'une désignation utilisée pour représenter les petits dialogues et petites remarques du quotidien comme les salutations (« Bonjour », « Au Revoir », ...) mais aussi les compliments ou les reproches. Dialogflow propose de définir pour chaque projet une liste de réponse pour un corpus assez étendu de « small talk ». Même si cette fonctionnalité ne nous permet pas de gagner spécialement en efficacité ou en intelligence pour notre projet, elle est un plus non négligeable pour l'expérience utilisateur. En effet, cela permet de renforcer un peu le côté « humain » du programme en le détournant juste légèrement de son objectif et de ses compétences principales. Toute la partie détection de ces « small talks » est entièrement gérée par Dialogflow, et la réponse est choisie aléatoirement parmi celles proposées par le développeur. Sur ce point, son seul travail pour que cette fonctionnalité fonctionne correctement est de lire le corpus de « small talks » préparé par Dialogflow et d'essayer de trouver quelques manières différentes de répondre pour chaque.

5. Exportation et importation

Enfin, Dialogflow propose un moyen extrêmement simple (en un clic) de faire une backup du projet, et un autre moyen tout aussi simple de restaurer un projet à partir d'une backup. Ce point est très important et intéressant car il permet pendant le développement de garder des versions stables du chatbot, et ainsi pouvoir travailler sur son amélioration en toute tranquillité (mais également de proposer une version de production et une version de développement). Cela permettra également

Dialogflow – Alexis Guyot

dans le futur de déplacer le bot sur un compte propre à l'observatoire (puisque pour l'instant il est développé sur mon compte personnel).

III. ASPECT JURIDIQUE

1. Licence et gratuité

Dialogflow existe en deux éditions : une version standard et une version entreprise. La version standard, celle proposée de base par la plateforme, est totalement gratuite et commercialisable (le code source n'étant pas publique mais utilisé par un système de requêtes). Le nombre de requêtes textuelles par jour est illimité mais le service ne supporte pas plus de 180 requêtes par minute. Sur ces points, les deux versions ne sont pas très différentes (la version entreprise propose juste 600 requêtes par minute). La différence se joue essentiellement sur le service client proposé par Google, quasiment inexistant sur la version standard, et sur les requêtes vocales faites au service. En effet, celui-ci permet de gérer une entrée vocale directement, et ce de manière illimitée sur la version entreprise (contre 1000 requêtes par jour pour la version standard). Cependant, cette fonctionnalité peut facilement être contournée par un programme annexe qui transforme depuis le navigateur une entrée vocale en requête textuelle puis l'envoie au moteur Dialogflow.

Il est également important de noter qu'il est possible de demander à Google une augmentation du quota de requêtes par minute pour la version Standard sans avoir à passer par la version Entreprise, si une justification est donnée (la limite est en fait là pour protéger Dialogflow d'abus). Cette demande se fait [ici](#).

2. Politique de confidentialité

L'API Dialogflow NLU ne récupère et stocke uniquement les phrases qui lui sont envoyées (pour le machine learning). Aucune donnée personnelle sur les utilisateurs n'est gardée ou utilisée. Donc du moment que les requêtes des utilisateurs de contiennent pas de données sensibles (du style données médicales), il n'y a pas de problème.

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

ANNEXE 2 – TWIKI

[Retour vers description des logiciels.](#)

Stage de Alexis Guyot - IUT Dijon - [3/04/18 au 8/06/18]

Important : cette page est réservée au suivi du stage, merci de ne pas la modifier

Informations générales pour les stagiaires

Pour toute information concernant ce stage : contacter André, Sébastien, Thomas

- ↳ Stage de Alexis Guyot - IUT Dijon - [3/04/18 au 8/06/18]
 - ↳ Sujet
 - ↳ Stage (Avril - Juin 2018)
 - ↳ Avril
 - ↳ Mai
 - ↳ Juin
 - ↳ Liens
 - ↳ Versions testables
 - ↳ Documentation
 - ↳ Liste de questions que l'on peut imaginer voir traitées par un portail intelligent
 - ↳ Remarques et discussion sur les questions ci-dessus
- ↳ Travail post stage éventuel
 - ↳ Liste des améliorations à envisager
 - ↳ Bugs connus

Sujet

- Proposition de stage

Figure 22 - En-tête de ma page Twiki

Mai

- 2.
 - Suite du travail sur le formatage des réponses pour certaines mesures :
 - Le type spectral, accompagné d'un sticker dont la couleur s'adapte en fonction de la couleur de l'étoile
 - Les coordonnées, affichées en décimal et en ICRS
 - Le mouvement propre
 - Le redshift/radial velocity
 - Le type d'objet
 - Les flux
 - Implémentation dans le code de deux nouveaux types de requêtes vers SIMBAD : une pour récupérer les flux (magnitudes), et une pour le type de l'objet (la version rédigée et longue), deux types de données non présentes dans la table principale.
 - Mise à jour de la mise en page des previews + correction d'un bug présent jusque là (les données affichées étaient celles de Sirius à chaque fois).
- 3.
 - Etude d'une solution pouvant être mise en place concernant les intentions de type "how many", plusieurs questions se posent à propos de ce point, à voir pendant la prochaine réunion.
 - Reprise totale du système d'historique : Suppression de l'autocomplétion jquery pour pouvoir récupérer les précédentes entrées avec les flèches haut et bas.
 - Modification de l'algorithme derrière la prédiction de texte, le précédent étant trop lent avec un gros corpus de mots.
 - Déploiement d'un serveur Tomcat pour accéder au chatbot depuis l'extérieur. Ceci est maintenant possible à cette adresse : http://130.79.128.184:8084/Chatbot_v1/
 - Entrainement du chatbot.
- 4.
 - Matin : Café + Séminaire (ajout de commentaires dans le code avant)
 - Implémentation dans Dialogflow et dans le code d'une nouvelle intention liée aux catalogues :
 - Obtenir une liste de catalogues en fonction de certains paramètres : nom du catalogue, type d'objet étudié, nom de l'objet étudié, mesure présente + possibilité de préciser comment ordonner les résultats (par date ou par popularité).
 - Messages d'erreurs plus explicites.
 - Tentative de début de conception objet pour rendre le code ouvert à l'extension mais fermé à la modification (principe O de SOLID) --> Il semblerait que la PO ne soit pas très adaptée dans notre cas, à voir.
 - Correction de bugs, notamment avec Vizier (voir partie bugs connus).
 - Entrainement du chatbot.

Figure 21 - Extrait de mon journal de bord

Documentation

- [Lexique.pdf](#): Lexique des termes astronomiques rencontrés durant le stage.
- [Comprendre_Sesame.pdf](#): Synthèse sur l'application Sesame.
- [Machine_learning.pdf](#): Synthèse sur le machine learning dans le domaine du traitement du langage naturel
- [Dialogflow.pdf](#): Présentation du moteur de traitement du langage naturel Dialogflow

Liste de questions que l'on peut imaginer voir traitées par un portail intelligent

1. What is the redshift of 3C273? What is the redshift of the Virgo Cluster?
2. What is the parallax of Barnard's star? What is the distance of Barnard's star? What is the proper motion of Barnard's star?
3. What is the effective temperature of Sirius?
4. What are the galactic coordinates of Geminga?
5. Which galaxy interacts with NGC 4038?
6. Show me an image of the Pleiades in the K band
7. How many QSOs are there at redshift larger than 6? How many QSOs are there at z>6?
8. What is the redshift of galaxies members of the Virgo cluster?
9. Find globular clusters within 3° of M31. Find globular clusters in M31.
10. Query the latest Veron catalogue
11. What is the period of Algol? List of periods of Algol-type stars.
12. What is the effective temperature of T Tau? What is the effective temperature of T Tauri stars?
13. Find supernovae in galaxies brighter than V=12
14. How many planets orbit Kepler 20?
15. List of catalogues measuring surface gravity of giant stars
16. Get color thumbnails of Messier objects
17. List of galactic X-ray supernova remnants

Figure 23 - Utilisations possibles de Twiki

ANNEXE 3 – STRUCTURE D'AUTO COMPLETION/PREDICTION DE TEXTE

Après plusieurs jours de recherche et d'essais pour mettre en place une structure proposant un service d'auto complétion et de prédiction de texte, j'ai finalement décidé d'en proposer une moi-même. Je vous propose ainsi dans cette partie un peu « hors-sujet » une analyse et une critique de mon travail sur cette partie.

Pour commencer, il faut savoir que la structure demandée dans le cadre de notre interface devait proposer ses services mot par mot et non pas pour la phrase en entier. Ainsi, si l'utilisateur était en train d'écrire « What i ... », on ne voulait pas que la structure propose « What is the distance of Sirius » mais juste « is ». La principale raison à cela est le fait que l'astronome devant cet outil allait très probablement à un moment ou à un autre se retrouver à devoir retaper plusieurs fois une même phrase mais en changeant à chaque fois un ou deux mots maximum (par exemple le nom d'une mesure ou d'un objet astronomique). Plutôt que l'obliger à devoir accepter la compléction, sélectionner les mots à changer puis les retaper, il était plus intéressant de juste lui fluidifier sa saisie et lui proposant d'écrire une ou deux lettres puis de sélectionner le mot qu'il souhaitait grâce à des raccourcis. A partir de ce constat-là, il était ainsi impossible de proposer un système d'auto complétion générique comme celui proposé par jQuery. Celui demandé devait beaucoup plus ressembler à l'outil proposé par les téléphones portables lors de la saisie. Comme sur ceux-ci, l'idée d'ajouter un algorithme de prédiction de texte est donc venue naturellement.

Lors de mes premières réflexions sur le sujet, j'ai pensé à l'option des arbres PATRICIA pour l'auto complétion. Il s'agit d'une structure de données que j'avais pu découvrir à l'IUT et plus précisément lors de l'examen du module de « Structure de Données » du troisième semestre. Pour démarrer, j'ai donc contacté M. Guidet, l'enseignant en charge de cette matière, pour lui demander une correction et par la même occasion lui demander s'il avait une idée sur comment implémenter un système de prédiction de texte, mes recherches effectuées sur le sujet jusque-là ne m'ayant rien ramené de très convaincant. A cela, M. Guidet m'avait exposé l'idée d'une structure tournant autour d'un histogramme. Une fonction associée à celui-ci aurait permis de retourner la probabilité que deux mots passés en entrée se suivent dans une phrase. Cette méthode utilisant principalement des tableaux avait l'avantage de proposer un résultat très rapide mais de

vite prendre beaucoup de place en mémoire. En m'inspirant de cette idée, de la structure des résultats à la base VizieR et un peu du système général des bases de données, j'ai proposé une autre méthode.

Le système est en fait un objet gravitant autour d'une structure JSON, un format de structuration des données très bien géré par le langage Javascript. Celle-ci est constituée de trois branches principales : une première représentant un dictionnaire (sans doublon) et contenant tous les mots gérés par la structure, une deuxième contenant un ensemble de scores représentant le nombre de fois qu'un mot est entré dans le dictionnaire, et une troisième contenant plusieurs listes de suivants. Le dictionnaire est rangé par ordre alphabétique, ce qui permet une recherche très rapide par dichotomie. Dans l'objet, plusieurs fonctions accompagnent la structure JSON pour tout ce qui concerne l'ajout, la recherche ou le tri. Chaque mot du dictionnaire est lié à son score et à ses suivants par sa position dans celui-ci. Ainsi, le score et les suivants du mot à l'indice 42 dans le dictionnaire se trouveront aux indices 42 de leur branche respective. Les suivants sont en réalité les indices auxquels se trouvent les mots d'après dans le dictionnaire. Chaque message envoyé par l'utilisateur est découpé en mots qui sont ensuite entrés dans la structure. Si le mot étudié est déjà dans le dictionnaire, son score est augmenté, sinon il est ajouté.

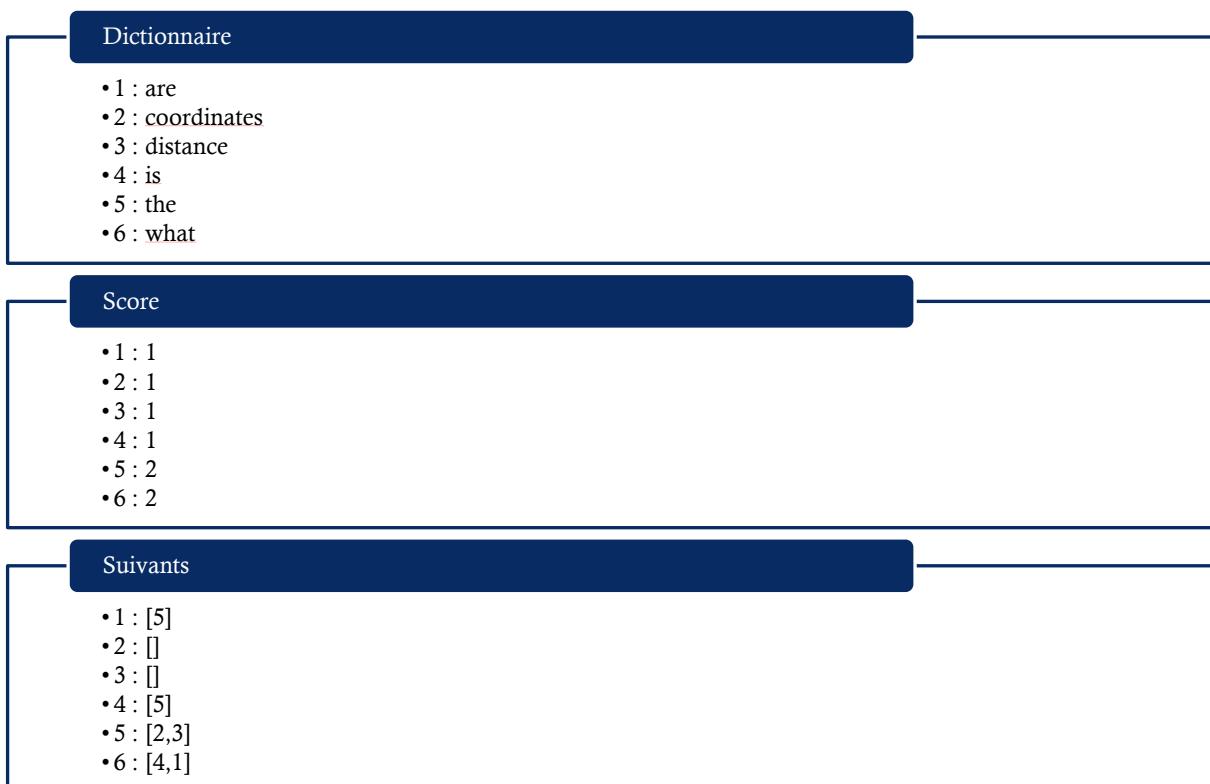


Figure 24 - Version simplifiée de la structure

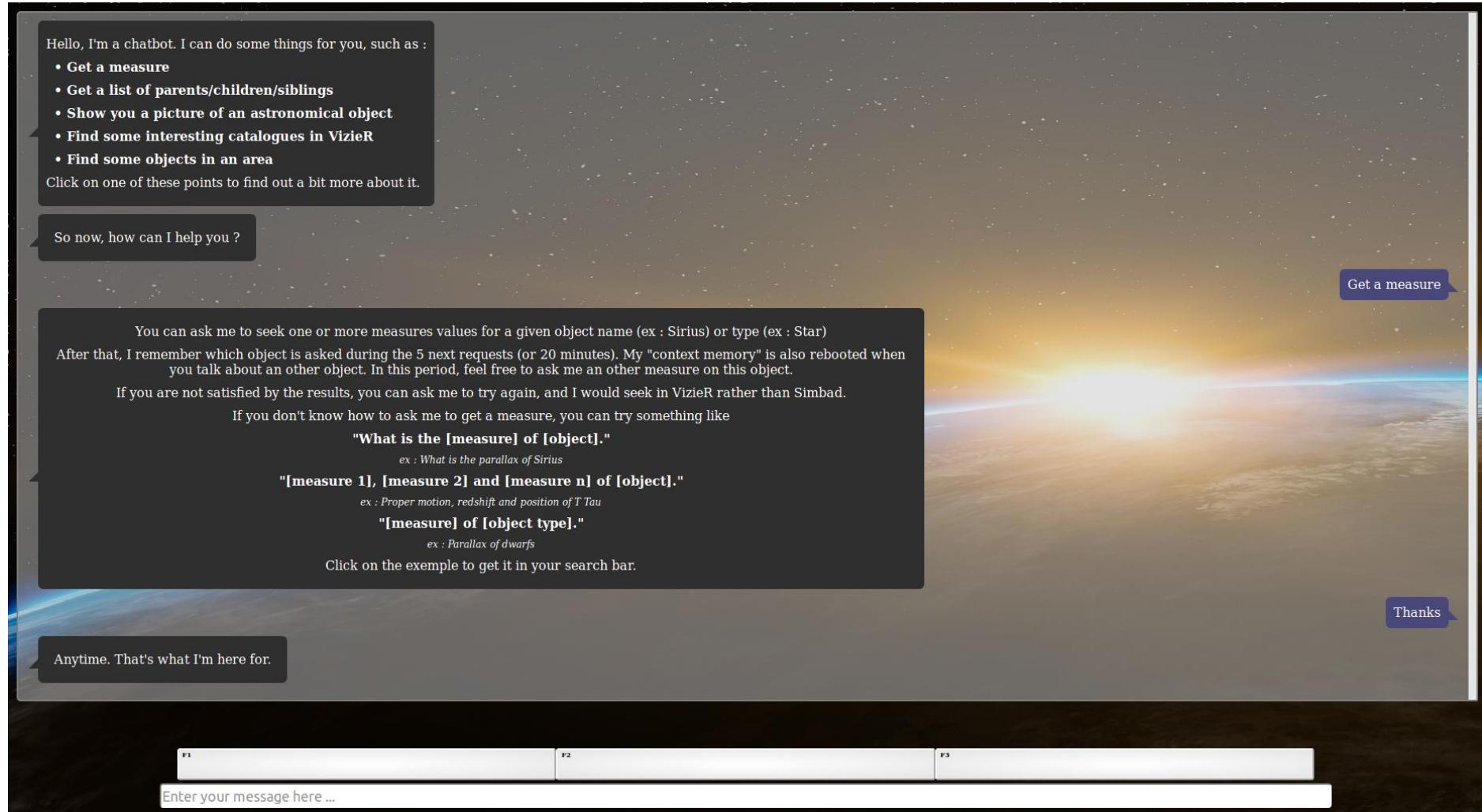
Ainsi, les mots très souvent utilisés comme « the » ou « of » possèdent un plus gros score et sont susceptibles d'être proposés en premier lorsque l'utilisateur commence à entrer un mot commençant par les lettres 't' ou 'o'. Cela permet également de proposer un système qui s'adapte petit à petit aux habitudes d'utilisation de chacun, la structure étant stockée à chaque mise à jour dans le *local storage*⁵ du navigateur. Concernant les limites, il faut savoir que la zone de stockage citée précédemment propose un espace de 10 MB. Après avoir fait quelques calculs comme la longueur moyenne d'un mot dans le dictionnaire, etc., et en ne prenant en compte que la moitié de cette capacité de stockage, on arrive tout de même à un peu plus de 100 000 entrées possibles. Cela est plus que nécessaire, la langue anglaise complète en possédant environ [200 000](#). La vraie question concerne surtout la rapidité à laquelle les résultats peuvent être trouvés dans la

⁵ Petite zone de mémoire proposée pour un site web par le navigateur.

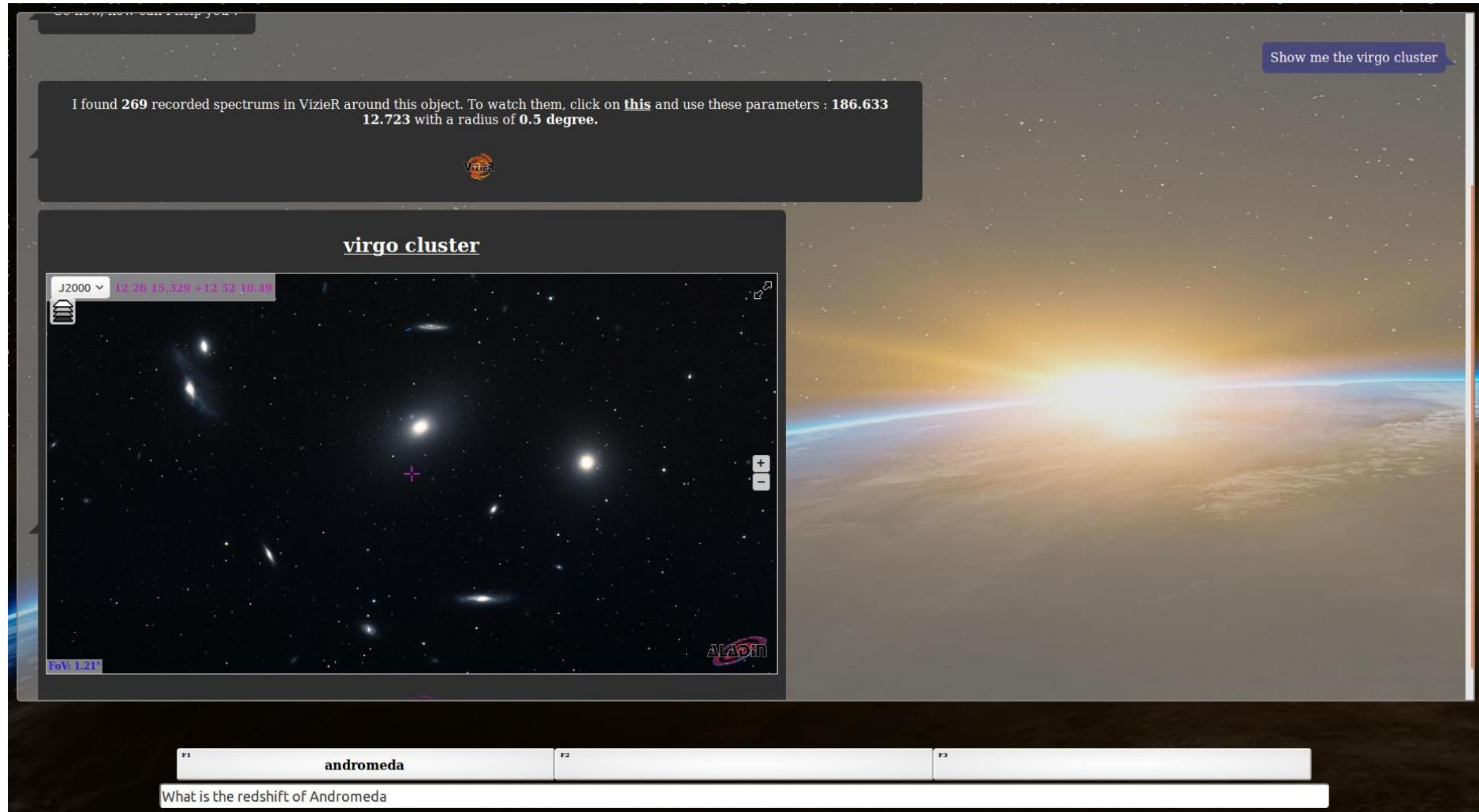
structure. Sur ce point aucun problème n'a été détecté au cours de mon stage. Après utilisation très régulière chaque jour pendant plusieurs semaines, je n'ai réussi à atteindre que 200 mots environ et le temps d'exécution était instantané. Dans un souci de tester les limites du programme, j'ai ajouté des phrases aléatoires jusqu'à ce que le dictionnaire soit composé d'un millier de mots. Avec une telle composition, toujours aucun temps d'attente à l'échelle humaine n'était détecté. Globalement, on peut donc considérer qu'il n'y aura pas de problème concernant la rapidité de récupération d'une prédition ou d'une compléction. En effet, il faut garder à l'esprit que quelqu'un qui utilise un tel outil n'a pas à vocation d'écrire un roman à chaque question au **chatbot** et va même concrètement par habitude utiliser quasiment tout le temps le même vocabulaire et les mêmes tournures de phrases, à une ou deux variantes près.

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

ANNEXE 4 – CAPTURES D'ECRAN DE L'APPLICATION



INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL



INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

What is the redshift of Andromeda

I looked for some data about **Andromeda** linked to this measure (redshift) in Simbad. Here is what I found :
Radial Velocity / Redshift : v(km/s) -300 [4] C [2012AJ....144....4M](#)



I want to see it please

I found **1859** recorded spectrums in VizieR around this object. To watch them, click on **this** and use these parameters : **10.684 41.268** with a radius of **0.5 degree**.



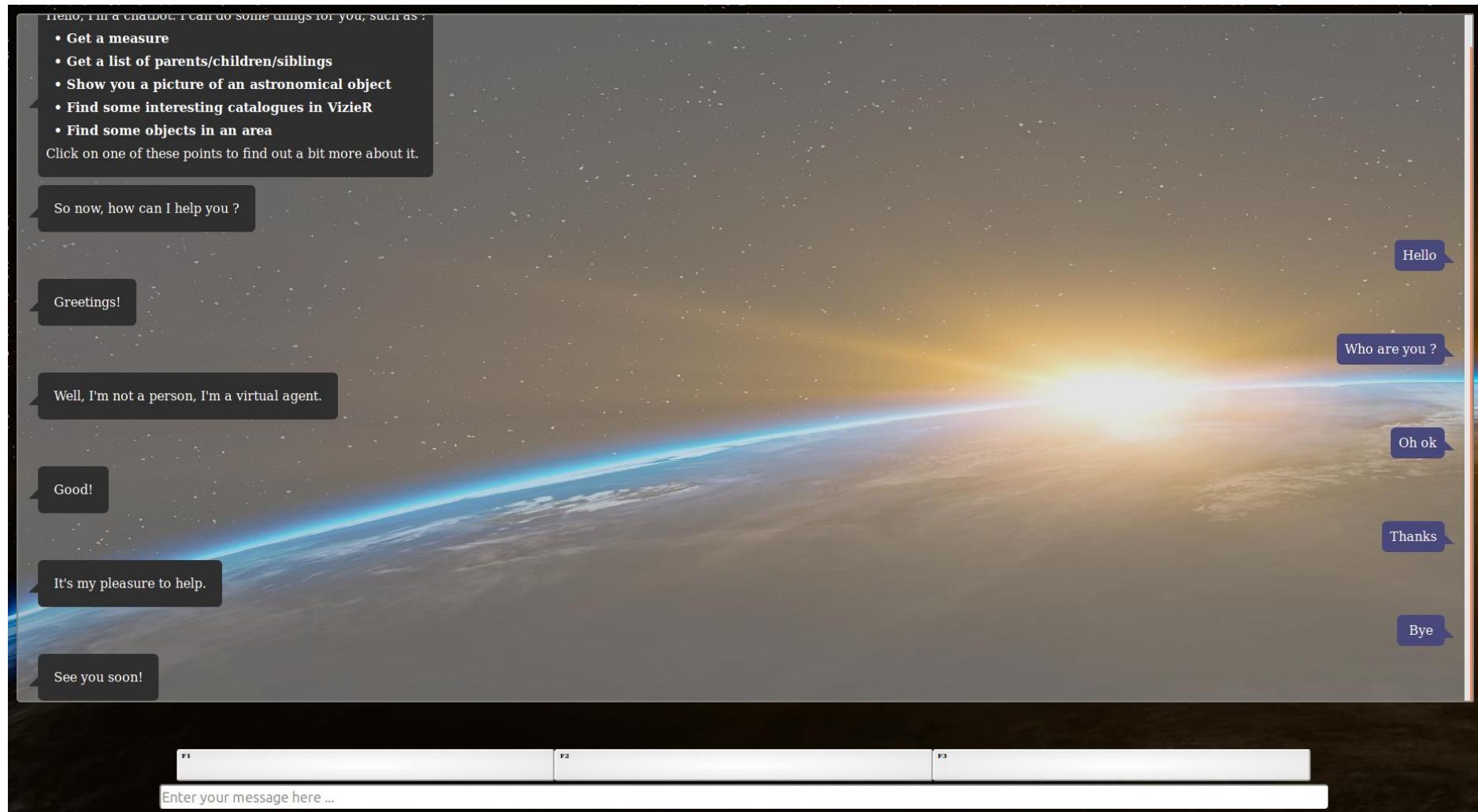
Andromeda

J2000 00 42 44.330 +41 16 7.50



Enter your message here ...

INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL



INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL

The screenshot displays a user interface for querying astronomical data. At the top, a dark blue header bar contains the title "INTERROGATION DE SERVICES DE DONNEES ASTRONOMIQUES EN LANGAGE NATUREL". Below this, a large window shows a star map of the Virgo Cluster. The cluster is highlighted with red squares, and several galaxies are visible as bright points of light. The interface includes a coordinate selector (J2000) with coordinates 18 26 32.100 +12 43 21.98, a zoom control, and a FoV indicator of 1.05°. The Aladin logo is present in the bottom right corner of the map window.

Find some galaxies 0.3 degrees around the virgo cluster

virgo cluster

J2000 18 26 32.100 +12 43 21.98

FoV: 1.05°

Aladin

Among all these objects, here are the objects with the type "galaxy".

Get the complete list (43 results)

LEDA 40571
LEDA 40823

Enter your message here ...

LEDA 40571

Galaxy

Main identifier : LEDA 40571

Coordinates (decimal) : 186.431 12.674

Click for more information ...

Aladin

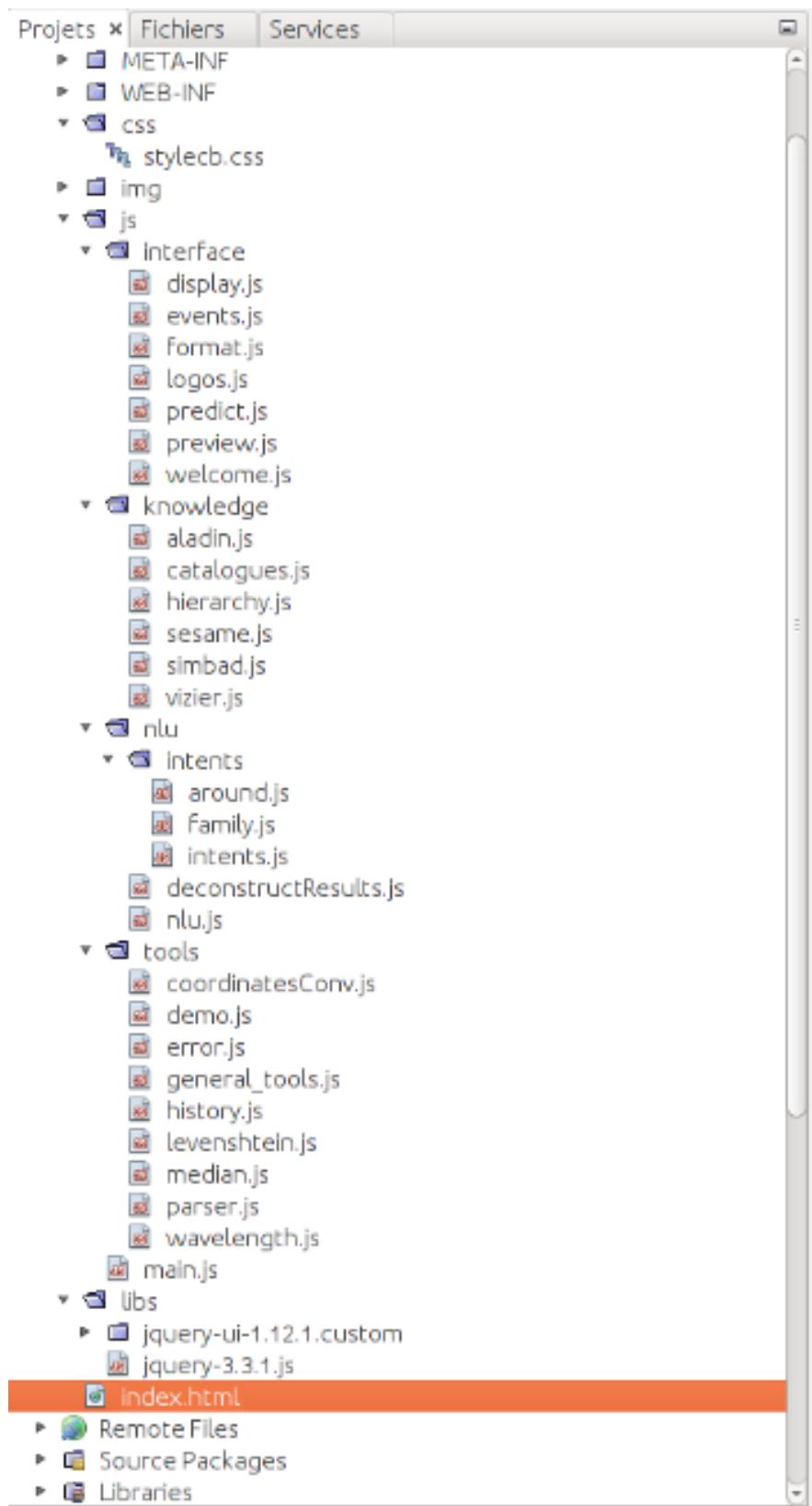
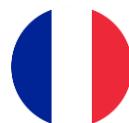


Figure 25 - Arborescence du projet

RESUME ET MOTS CLES



Afin de conclure les deux années de DUT, une période de stage est aménagée durant les 12 dernières semaines afin d'offrir aux étudiants une première expérience professionnelle dans l'informatique. Dans ce cadre-là, j'ai effectué du 3 avril au 22 juin 2018 un stage à l'observatoire astronomique de Strasbourg et plus précisément dans le service du CDS, le Centre des Données astronomiques de Strasbourg. Ma mission était de développer une nouvelle application permettant aux astronomes d'accéder à quelques outils du service en utilisant le langage naturel, c'est-à-dire en faisant des phrases comme si on posait la question à une vraie personne, plutôt que de passer par des formulaires parfois longs et peu intuitifs.

Mots-clés : Langage naturel, développement web, Javascript, agent conversationnel, Dialogflow, bases de données, astronomie.



To conclude on the two years of the DUT training, an internship period was planned during the last 12 weeks to allow students to have a first professional experience in IT. In this context, I made from April 3rd to the June 22nd, 2018 an internship in the astronomical observatory of Strasbourg, more precisely in the “Centre de Données astronomiques de Strasbourg” (CDS) service. My mission was to develop a new application allowing astronomers to reach service's tools using natural language, with sentences as if they were talking with a true person, rather than using long and not very intuitive forms.

Key words : Javascript, chatbot, Dialogflow, NLU, natural language, web development, databases, astronomy