



DIALOGFLOW

LE MACHINE LEARNING AU SERVICE DU TRAITEMENT DU
LANGAGE NATUREL

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 Speacktoit, 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 Speacktoit 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

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 ne contiennent pas de données sensibles (du style données médicales), il n'y a pas de problème.