Polytech Paris-Sud

Traitement Automatique des Langues

Projet Chatbot – NBA Assistant

COULIBALY Mamadou
OUDGHIRI Aboubakr

Promo 2017-2018

5 Mai 2018

Choix de la langue

Au cours de ce projet, nous avons utilisé un chatbot qui communique en anglais. Nous avons fait ce choix car l'anglais est une langue qui est plus parlée que le français. De plus, l'anglais possède une grammaire plus simple.

1. Mode 1

Pour commencer, nous avons constitué un fichier composé d'un backchannel par ligne.

Pour répondre à l'utilisateur nous utilisons un principe assez simple. Après chaque entrée de l'utilisateur, on tire une phrase au hasard dans notre fichier. Pour ne pas avoir deux fois le même backchannel, on sauvegarde le dernier élément que l'on a tiré au sort. Lorsque l'on tire un backchannel au sort on le compare au précédent. Tant qu'ils sont identiques, nous refaisons un tirage au sort.

2. Mode 2

Dans le mode 2 nous avons implémenté deux éléments : Un mode Eliza qui montre de l'empathie envers le locuteur. En effet, la conversation sera centrée sur les questions du locuteur et la réponse du chatbot se fera en fonction de cela.

Si par exemple, le locuteur dit « I am very very happy ».

Le chatbot répondra « Why are you happy ?"

De même au passé ou au futur : « I was very sad » et I will be in the kitchen tomorrow" donneront respectivement "Why were you very sad" et "Why will you be in the kitchen ?"

C'est à dire que le chatbot est capable de détecter le temps qui est employé (passé, présent, futur) et de répondre en conséquence.

Nous avons également voulu aller un peu plus loin en traitant des formes autres que la première personne du singulier (« I »). Ainsi ce mode accepte toutes les formes à toutes les personnes de la langue anglaise (« I », « You », « He / She / It », « We » and « They ».

Pour l'instant nous couvrons les axillaires « être » et « avoir » qui sont très utilisés dans la langue anglaise.

Le mode Eliza est donc assez complet puisqu'il supporte plusieurs temps, plusieurs personnes et les deux auxiliaires.

Afin de mettre en place tout cela, nous avons encore une fois utilisé des fichiers textes. Il s'agit de deux fichiers nommés « input.txt » et « output.txt ». Dans ces fichiers, nous avons écrit les règles de conjugaisons aux différentes personnes et aux différents temps. Grâce à cela notre chatbot est capable de connaitre les règles que nous lui avons fournies. Par exemple, il sait que si le locuteur emploi « I », il devra répondre par « You »

Ex : « I am very good at natural language processing" devient "Why are you very good natural language processing"

Nous avons mis en place ce système de fichier car il apporte un très grand avantage qui est que le chatbot pourra s'améliorer avec le temps de manière lorsque de nouvelles règles de conjugaisons ou de grammaire seront ajoutés.

Si nous voulons désormais que le mode Eliza puisse traiter les entrées comportant de nouveaux verbes comme « like » ou « love », il nous suffit simplement de rajouter quelques lignes.

A supposer que nous trouvions un moyen d'automatiser l'ajout de nouvelles règles sans interventions humaines, notre chatbot pourra en théorie pouvoir reconnaitre tous les verbes de la langue anglaise et aura les capacités d'y répondre.

En plus du mode Eliza, nous avons mis en place un mode appelé « keyword ». Ce dernier ce base sur l'utilisation de mots-clés et propose une question basée sur le thème abordé par le locuteur. Nous avons choisi 5 thèmes qui nous plaisent beaucoup : le sport, la cuisine, l'informatique, les automobiles et motos ainsi que la famille. Pour chacun de ces thèmes nous avons cherchés un ensemble de mots-clés qui ont un lien fort avec le thème.

Par exemple pour la cuisine, il s'agit de d'ingrédients tels que « curry », « pepper » ou encore des fruits comme «grape », « orange » etc.

Pour notre vocabulaire sera grand et composé de mot se rapprochant du thème plus le chatbot reconnaitre le contexte de la phrase et aura des réponses adaptées. Encore une fois, nous pensons qu'il est possible d'augmenter le vocabulaire du chatbot grâce à internet.

L'un des points intéressant que nous avons identifié est lorsque dans une même phrase, il y a des mots qui correspond à plusieurs des thèmes que nous avons choisis. Par exemple dans la phrase « J'aime les voitures orange », il y a à la fois le mot voiture qui se réfère au thème des voitures et le mot « oranges » qui peut se référer au thème de la cuisine. Dans ce cas une étude plus poussé de la phrase est nécessaire et il faut donc analyser la sémantique. Détecter des mots-clés ne suffit pas dans les cas les plus complexes.

Dans le chatbot que nous avons conçu, le thème choisi est celui du premier mot-clé que nous détectons. Dans l'exemple ci-dessus, notre chatbot aurait identifié le thème des voitures ce qui est tout de même correcte. Mais cela ne sera pas toujours le cas.

Une fois que le mot clé a été identifié, nous tirons aléatoirement une phrase se rapportant au thème en question. Le principe est le même que pour les backchannels, il ne faut pas tirer au sort deux fois la même réponse sinon le locuteur sentira pleinement qu'il est fait à un robot.

Etant donné que nous sommes conscients que pour l'instant le chatbot possède des connaissances qui sont limitées que ce soit avec le mode Eliza ou le mode Keyword, nous avons prévu une solution en cas de non reconnaissance d'une phrase.

Nous avons tout simplement intégré le mode 1 avec les backchannels afin de donner des « feedback » et montrer que nous écoutons toujours ce que nous dit le locuteur avec attention.

En résumé, le mode 2 essaye de détecter en premier une forme de conjugaison avec le mode Eliza. S'il ne détecte rien alors il essaye de détecter un mot clé avec le mode Keyword. S'il ne trouve toujours rien alors ce sont les backchannels qui sont appelées.

3. Mode 3

Nous avons nommé ce mode « NBA Assistant ». Il s'agit d'un outil permettant de mieux connaitre le monde de la NBA. La NBA est une ligue de Basketball américaine.

Nous avons fait le choix de traiter ce domaine-là car c'est un domaine connu internationalement et qui nous plait beaucoup. « NBA Assistant » est destinée à plusieurs profils d'utilisateurs. Il peut s'agir de passionnés de basket, de personnes souhaitant découvrir le monde de la NBA ou même d'acteur professionnels tels que de coach souhaitant analyser les performances de leurs joueurs.

« NBA Assistant » est capable de répondre à des questions apportant des statistiques à propos des joueurs, des équipes ou des matchs. La NBA est l'un des sports qui génère le plus de statistiques au mondes à l'issu d'un match opposant deux équipes. La problématique est donc que lorsque l'on recherche des informations sur une statistique particulière, il faut rechercher cela sur des sites particuliers qui sont connus par les grands fans mais pas par les novices. Grâce à NBA Assistant, l'obtention d'information devient beaucoup plus facile et plus ludique. Cela devient très pratique puisque toutes les données sont situées à un seul et même endroit. Tout cela rend les recherches très faciles.

3..1 Récupération d'informations

La première phase par laquelle nous avons commencé est la récupération des données. C'est une phase primordiale car elle sans les données des statistiques, notre chatbot devient inutilisable et nous ne pouvons rien faire. Pour récupérer des données et constitué notre propre base de données nous parcourons plusieurs sites ouverts au public qui répertorie des statistiques telles que (https ://www.basketball-reference.com/, https ://basketball.realgm.com). Etc . etc. Pour cela, nous utilisons une librairie appelée BeautifulSoup. Elle nous permet parser les balises html et d'extraire les informations importantes. C'est en quelques sortes du « data mining ». Ainsi nous pouvons lancer des « requêtes » selon les désirs de nos utilisateurs. Toutes les informations sur la NBA ne sont pas de vraies bases de données mais plutôt des fichiers au format .csv sous forme de tableaux. Le fait de disposer de bases de données SQL plutôt que de fichiers .csv nous aurait permis de manipuler les données plus facilement grâce à des requêtes SQL.

3..2 Reconnaissance de la phrase du locuteur

Avant de commencer l'implémentation du « NBA Assistant » nous avons hésité entre deux méthodes distinctes. Une méthode basée sur des mots-clés et une méthode basée sur un ensemble de phrase modulable qui peuvent être reconnu et ajoutée au fur et à mesure.

Nous avons pesé le pour et le contre. La méthode basée sur les mots-clés ne permet pas vraiment de comprendre la phrase du locuteur. Le locuteur pourrait donc placer une phrase qui n'a pas de signification mais contenant des mots-clés. Cette phrase serait alors traitée de la même manière qu'une phrase qui a du sens et qui contient les mêmes mots-clés.

L'avantage de la méthode des mots-clés est qu'elle est assez facile à implémenter.

Cependant nous avons opté pour une autre solution qui nous a semblée meilleure que les mots-clés.

Le fonctionnement est le suivant : afin de reconnaitre la phrase du locuteur nous avons utilisé un fichier appelé « accepted_sentences.txt » qui liste toutes les phrases qui sont reconnues par « NBA Assistant ».

Nous avons essayé de rendre les phrases le plus modulable possible. En effet nous savons bien que nous ne parviendrons jamais à pouvoir lister toutes les phrases auxquelles les locuteurs pourraient penser. Pour assurer cette modularité dans les phrases nous avons placé des « tokens » qui peuvent prendre plusieurs valeurs.

Par exemple dans la phrase «How many _statistics.txt _player.txt average ? »

« _statistics.txt » et « _players.txt » sont des tokens. C'est-à-dire que si l'on utilise n'importe lequel des mots situés dans ces fichiers la phrase reste correcte. Cette modularité nous permet de gérer beaucoup de manière assez efficace. Avec une meilleure connaissance de l'anglais, ou bien en collaboration avec des linguistes, nous pourrions mieux structurer nos tokens et généraliser encore plus les phrases.

Pour voir si une phrase est acceptée ou non, nous la parcourons et petit à petit tout en essayant de trouver si la forme de la phrase se situe bien dans « accepted_sentences.txt »

Pour l'instant, nous avons pensé à une dizaine de phrase génériques acceptées. Avec toutes les combinaisons possibles, cela représente en réalité des centaines de possibilités.

En effet, la phrase «How many _statistics.txt _player.txt average ? » peut donner :

[2]

“How many points LeBron James average ?”

“How many rebounds LeBron James average ?”

“How many points Stephen Curry average ? “

“How many X Y average ? “

Pour l’amélioration du NBA Assistant, nous avons pensé a un système qui permettrait de l’améliorer de plus en plus rapidement en fonction du nombre d’utilisateur. En effet, à chaque fois qu’un locuteur entrera une phrase que nous ne reconnaissons, nous la sauvegardons dans un fichier « newPropositions.txt ». Ce fichier grandira donc avec le nombre d’utilisateurs. Et dans ce fichier, pour chaque phrase nous analyserons si elle est correcte. Si elle n’est pas correcte elle sera supprimée. Si elle est correcte on essayera de la rendre générique afin de multiplier les combinaisons possibles. L’Ideal serait de trouver un moyen d’automatiser la phase de vérification d’une phrase pour gagner beaucoup de temps.

En plus d’augmenter le nombre de possibilités, les tokens nous permettent de cibler les informations essentielles que l’on recherche dans la phrase et donc de mieux la comprendre.

3..3 Récupération des cibles de la phrase

Lorsque l’on analyse une phrase, notre objectif principal est de la reconnaitre. Une fois qu’elle est reconnue, il faut que nous trouvions la valeur des éléments clés.

Par exemple dans la phrase « How many points LeBron James average ? » les deux informations essentielles à capturer sont « points » et « LeBron James ». Cette phase qui nous permet de savoir ce que veut le locuteur. La récupération est simplifiée par le choix de conception basée sur les tokens.

3..4 Recherches des informations dans nos fichiers

Maintenant que nous savons ce que veut le locuteur il faut que nous récupérions le résultat de sa requête dans nos bases de données. Pour cela nous avons implémenter des fonctions de manipulations et de tri.

Par exemple : « Who scores the most points ? »

Nous avons capturé les cibles « most » et « points ». Par conséquent dans la base de données des statistiques des joueurs, nous faisons un tri croissant des points et récupéront le dernier élément de la liste.

Si nous avions « less » à la place de « most » nous aurions récupéré le premier élément de la liste.

3..5 Réponse du NBA Assistant

Maintenant que nous avons récupérer les informations qu'il nous faut, il ne nous reste plus qu'a bien formuler la réponse. Nous utilisons donc le même principe que pour les phrases en entrées. Dans un fichier nommé « answers.txt ». Nous avons des phrases réponses qui sont modulables.

Si l'on reprend l'exemple « How many _statistics.txt _player.txt average ? »

La réponse modulable associée est « _player average +value _statistics »

Avec un exemple concret on a :

« How many points LeBron James average ? » -> « LeBron James average 28 points »

Conclusion

En conclusion avec le mode 3 « NBA Assistant » nous avons voulu mettre place les fondations pour une solution de chatbot qui est évolutive. C'est-à-dire qu'elle s'améliore avec le temps et si possible sans l'intervention humaine. Pouvoir comprendre la sémantique est une tâche difficile que nous avons essayé de rendre un peu plus facile avec le système de token.

Ce qui est intéressant avec les chatbot c'est que l'on peut combiner plusieurs domaines. Avec toutes les données on pourrait imaginer associé tout cela avec du Machine Learning afin de construire des modèles et pouvoir mieux comprendre les données et même pour pouvoir prédire l'issue des matchs