

Agents conversationnels

Master IA2

Célia da Costa Pereira
Université côte d’Azur
celia.perera@unice.fr

Année universitaire 2021/2022

Résumé

Dans cette séance, vous allez partir d’un code open source existant qui programme les fonctionnalités de base du chatbot Eliza vu en cours.

1 Introduction

Nous allons utiliser une réalisation du *chatbot* Eliza en Python, <https://github.com/wadetb/eliza>, mise à disposition par son développeur avec une Licence MIT.

2 Consignes

1. Le *chatbot*, qui “parle” anglais, peut être lancé avec la commande

```
python3 eliza.py
```


Essayez d’avoir une petite conversation avec lui.
2. Tout le comportement du *chatbot* est contrôlé par un script qui se trouve dans le fichier `doctor.txt`, qui doit se trouver dans le même répertoire que le programme `eliza.py`. Eliza lit ce que l’utilisateur tape, le traite et formule une réponse.
Le traitement comprend les étapes suivantes.
 - (a) Tout d’abord, la phrase est décomposée en mots, séparés par des espaces. Tout le traitement ultérieur porte sur l’ensemble de ces mots, et non sur leurs caractères individuels qui les composent.
 - (b) Un ensemble de pré-substitutions a lieu.
 - (c) Eliza prend tous les mots de la phrase et fait une liste de tous les mots-clés qu’il trouve. Il classe cette liste de mots-clés par ordre de poids décroissant. Il traite ces mots-clés jusqu’à ce qu’il produise une sortie.
 - (d) Pour le mot-clé donné, une liste de patrons de décomposition est recherchée. Le premier qui correspond est sélectionné. Si aucune correspondance n’est trouvée, le mot-clé suivant est sélectionné à la place.
 - (e) Pour le patron de décomposition correspondant, une transformation est sélectionnée. Il peut y avoir plusieurs transformations, mais une seule est utilisée pour une phrase donnée. Si une phrase suivante sélectionne le même patron de décomposition, la transformation suivante dans la séquence est utilisée, jusqu’à ce qu’elles aient toutes été utilisées, auquel cas Eliza recommence avec la première transformation.
 - (f) Un ensemble de post-substitutions a lieu.

(g) Enfin, la phrase résultante est affichée en sortie.

Le script donne les listes des règles de pré- et post-substitution, les listes de mots clés, les patrons de décomposition et les transformations. En outre, il existe une fonction de recherche de synonymes, qui est expliquée ci-dessous.

Chaque ligne de script est précédée d'une étiquette qui indique à quelle liste elle appartient. Voici une explication des étiquettes.

- **initial** : ce qu'Eliza dit au démarrage.
- **final** : ce qu'Eliza dit pour clore une séance.
- **quit** : si l'utilisateur tape ça, Eliza s'arrête ; il peut y en avoir plusieurs.
- **pre** : une règle de pré-substitution ; si le premier mot apparaît dans la phrase, il est remplacé par le reste des mots.
- **post** : une règle de post-substitution ; si le premier mot apparaît dans la phrase, il est remplacé par le reste des mots.
- **key** : un mot-clé, suivi par son poids (optionnel) ; les mots-clés ayant un poids plus important sont sélectionnés de préférence à ceux qui ont un poids moindre ; si aucun poids n'est indiqué, il est considéré comme égal à 1.
- **decomp** : Un patron de décomposition ; le caractère * représente une séquence quelconque de mots.
- **reasm** : une transformation, aussi appelée "patron de réassemblage" ; une séquence de mots qui correspond à * dans le patron de décomposition peut être utilisée dans le réassemblage ; par exemple, (2) insère les mots correspondant au deuxième * dans le patron de décomposition.
- **synon** : une liste de synonymes ; dans un patron de décomposition, par exemple, @be peut correspondre à chacun des mots "be, am, is, are, was" s'il y a la ligne "**synon: be am is are was**" dans le script. Cette correspondance @be compte aussi comme un * aux fins de la numérotation des séquences à utiliser par la transformation.

En outre, si un \$ apparaît en première position dans un patron de décomposition, alors la sortie est formée normalement, mais elle est sauvegardée et Eliza passe au mot-clé suivant. Si aucun mot-clé ne correspond et il y a des phrases sauvegardées, une d'elles est choisie au hasard, utilisée, puis écartée. S'il n'y a pas de phrases sauvegardées et aucun mot-clé ne correspond, le mot-clé **xnone** est utilisé.

3 Questions

1. Ajoutez d'autres règles au script pour rendre le *chatbot* encore plus convaincant.
2. Pensez à un scénario de conversation par exemple, voyage, cuisine, musique, et ajoutez des règles au script pour soutenir une conversation dans ce scénario.
3. "What is your name?" ne se comporte pas comme "What is your name ?" Modifiez le code python pour éviter ce problème.
4. Modifier le code pour qu'Eliza se rende compte si l'utilisateur répète les mêmes phrases et gère correctement cette situation, par exemple qu'Eliza dise "Stop repeating yourself.". Utilisez la distance de Levenshtein (edit distance) pour comparer deux phrases (ex. what is your name ? what's your name ? your name ? se ressemblent).
5. Remplacer les mots clés dans (Clé, Patron, Règles de transformation) par des regex.
6. Modifier le script pour exploiter les regex.
7. Pensez à rédiger un document décrivant la démarche que vous avez suivie, votre apport personnel et vos éventuelles remarques. Cette documentation sera intégrée dans votre rapport final.