

Mon premier ChatBot

Introduction

Dans le cadre du projet de fin de semestre, nous avons eu pour mission de développer un ChatBot. La tâche qui nous a été attribuée était de concevoir dans un premier temps plusieurs fonctionnalités permettant d'analyser différents documents ainsi que d'en tirer les informations nécessaires à l'élaboration de notre ChatBot. Ce qui nous a permis ainsi de développer ce dernier tout en respectant au maximum les consignes demandées. Notre ChatBot se base sur plusieurs discours de présidents. Subséquemment, il est capable d'élaborer des réponses seulement sous-jacentes au corpus document (ex: la nation, le climat...).

Présentation des fonctionnalités de notre programme

Notre programme est capable de:

- Connaître les mots les moins importants de tout le répertoire.
- Connaître les mots avec le plus grand score TF_IDF.
- Connaître le mot le plus prononcé par un président.
- Donner la liste des présidents qui ont prononcé un mot donné et lequel l'a le plus répété.
- Trouver quel président a prononcé un mot donné en premier.
- Trouver hormis les mots non important, les mots que tous les présidents ont évoqués.
- Répondre aux demandes de l'utilisateur grâce au ChatBot

Présentation technique

```
import os
import math
from random import *
from time import sleep
```

Nous avons utilisé les modules `os` et `math` principalement afin de développer les fonctions nécessaires au ChatBot. Les modules `random` et `time` ont été utilisés seulement dans le but d'améliorer l'affichage.

```
2 usages 1 Gerem +1 +
264 def Matrice_TF_IDF(directory): #Fonction qui fait la matrice TF_IDF
265
266     #Initialisation de la list des stop words
267     stop_word_list = stop_word()
268     #Initialisation de toutes les variables utiles comme le score idf du repertoire,
269     idf = IDF(directory, stop_word_list)
270     list_word = list_of_word(directory, stop_word_list)
271     list_files = list_of_files(directory, extension: ".txt")
272     Matrix = []
273
274     #Parcours de la list contenant tout les mots de tout les fichiers
275     for word in list_word:
276
277         #Ajout de la ligne avec le score TF_IDF du mot étudié
278         tf_idf = TF_IDF(word, idf, list_files)
279         Matrix.append(tf_idf)
280     return Matrix
281
```

Le ChatBot s'appuie principalement sur une matrice TF-IDF à partir de laquelle il effectue des calculs. Grâce à celle-ci, notre ChatBot renvoie la réponse la plus pertinente avec le corpus de documents. La matrice n'est pas composée de Stopwords, ils ont été enlevés au préalable.

```

385 # Fonction qui calcule le produit scalaire de deux vecteurs.
386 1 usage new +
387 def dot_product(VectorA, VectorB):
388
389     dot_product = 0
390     for i in range(len(VectorA)):
391         dot_product += VectorA[i] * VectorB[i]
392     return dot_product
393
394 # Fonction qui calcule la norme d'un vecteur.
395 2 usages new +
396 def norm(vecteur):
397
398     summ = 0
399     for val in vecteur:
400         summ += val**2
401     norme = math.sqrt(summ)
402     return norme
403
404 # Fonction qui calcule la similarité cosinus entre une question et un document spécifique.
405 1 usage new +
406 def cosine_similarity(question, num):
407
408     VectorA = TF_IDF_question(question, directory: "./cleaned")
409     VectorB = Vector_B(num)
410     Dot_product = dot_product(VectorA, VectorB)
411     norm1 = norm(VectorA)
412     norm2 = norm(VectorB)
413     if norm1 * norm2 == 0:
414         return 0
415     cosine_similarity = Dot_product / (norm1 * norm2)
416
417     return cosine_similarity
418

```

Le calcul de vecteur similaire est la partie prépondérante de l'élaboration d'une réponse pertinente. Après avoir traité la question de l'utilisateur, notre programme transforme la question en vecteur et effectue le produit scalaire avec chaque ligne de notre matrice qui représente un vecteur associé à un document. Cette partie a été la plus exigeante car il a fallu développer plusieurs fonctions pour traiter la question ainsi que le corpus document.

```

end = False

# Boucle principale du programme
while not end:

    # Saisie de la fonctionnalité choisie par l'utilisateur

    Menu = input(
        "Voulez-vous utiliser le chatbot ou la fonctionnalité de votre choix ? (Entrer (c) pour le chatbot ou (f) pour les fonctionnalités : ")
    Menu = Menu.lower()

    # ChatBot
    if Menu == "c":

        with open("./Chatbot", "r", encoding="utf-8") as f:
            content = f.read()
            print(content)

        stop = False

        # Boucle ChatBot
        while not stop:

            question = input("Veuillez entrer un message: ")

            #Stop pour quitter le ChatBot
            if question == "stop" or question == "Stop":
                stop = True
                break

            answer = answer_with_starters(question, stopword)
            print(time_display(answer, delay: 0.05))

    elif Menu == "f":

```

Le fichier main est composé d'une boucle infinie qui est arrêtable seulement si l'utilisateur le décide. L'utilisateur a la possibilité de naviguer entre le ChatBot et les fonctionnalités. Le main est composé d'un if-elif qui renvoie le numéro de la fonctionnalité demandée par l'utilisateur.

Fonctionnalité 1: Mots les moins important de corpus de document

```
Voulez-vous utiliser le chatbot ou la fonctionnalité de votre choix ? (Entrer (c) pour le chatbot ou (f) pour les fonctionnalités : f
Saisir la fonctionnalité de votre choix (numéros de la fonctionnalité (Ex : 1) ) : 1
Voulez-vous savoir quels sont les mots les moins importants dans tous les discours : (y) or (n) y
Voici le(s) mot(s) le(s) moins important dans tous les discours :
messieurs , mesdames , france , peuple , histoire , faire

Voulez-vous voir d'autres fonctionnalités ? (y) or (n) |
```

Fonctionnalité 2: Le(s) mot(s) avec le TF_IDF le plus élevé

```
Saisir la fonctionnalité de votre choix (numéros de la fonctionnalité (Ex : 1) ) : 2
Voulez-vous savoir quel(s) est/sont le(s) mot(s) avec le plus grand score TF IDF ? : (y) or (n) y
Le(s) mot(s) qui a/ont le score TF-IDF le plus élevé est/sont :
pense
```

Fonctionnalité 3: Le mot le plus prononcé par un président

```
Voulez-vous utiliser le chatbot ou la fonctionnalité de votre choix ? (Entrer (c) pour le chatbot ou (f) pour les fonctionnalités : f
Saisir la fonctionnalité de votre choix (numéros de la fonctionnalité (Ex : 1) ) : 3
Souhaitez-vous voir le mot le plus prononcé par le président de votre choix ? (y) or (n) : y
Quel président voulez-vous choisir ? : Chirac
Chirac a prononcé le plus de fois le mot : français
```

Fonctionnalité 4: Quel président a dit un certain mot et lequel d'entre eux l'a le plus répété

```
Saisir la fonctionnalité de votre choix (numéros de la fonctionnalité (Ex : 1) ) : 4
Voulez-vous savoir quel est/sont le(s) nom(s) du (des) président(s) qui a (ont) parlé du mot de votre choix ainsi que celui qui l'a répété le plus de fois ? (y) or (n) : y
Quel mot voulez-vous chercher ? : Nation
Chirac a prononcé le mot : nation
Hollande a prononcé le mot : nation
Macron a prononcé le mot : nation
Voulez-vous savoir quel(s) président(s) a/ont répété le plus de fois le mot : nation ? (y) or (n) : y
C'est Chirac qui a répété le plus de fois ce mot
```

Fonctionnalité 5: Qui un prononcé un certain mot en premier ?

```
Saisir la fonctionnalité de votre choix (numéros de la fonctionnalité (Ex : 1) ) : 5
Voulez-vous chercher un mot afin de trouver quel président l'a énoncé en premier ? (y) or (n) :y
Entrez le mot souhaité : climat
C'est Macron qui a énoncé le mot climat en premier !
```

ChatBot:

```

      /  | | | | | /  \ | | |
     /-----\ | | | | / ^ \ \-----\
    / | | | | | / / \ \ | | |
   /-----\ | | | | /-----\ \ | |
  \-----\ | | | | / \ / \ \ \ | |
   \-----\ | | | | / \ / \ \ \ | |

'-----'
| _ \ / _ \ | | | | | | | | | | | | | | |
| |_) | | | | | | | | | | | | | |
| _ < | | | | | |
| |_) | | | | | | | |
|-----/ \-----/ | |

Veuillez entrer un message: Peux-tu me dire comment une nation peut-elle prendre soin du climat ?
Oui, bien sûr! et je songe bien sûr à François Hollande, faisant oeuvre de précurseur avec l'accord de Paris sur le climat et protégeant les Français dans un monde frappé par l
```


Bilan

L'élaboration de ce projet a été très enrichissant. Nous avons pu d'un côté apprendre un peu plus sur comment fonctionnent les ChatBot, ce qui est un premier pas dans le machine learning, un domaine qui nous intéresse fortement. Ainsi que de l'autre mettre en application des notions de mathématiques appris plutôt. Il n'y a pas eu de problème au niveau de l'organisation du travail en raison de l'importante charge de celle-ci. Chacun a pu contribuer à sa manière, le résultat que nous avons produit est en adéquation avec nos attentes, en espérant que celle-ci le soit aussi avec la notation :). En bref, malgré de nombreux problèmes rencontrés nous avons pu rebondir et trouver des solutions.