

2023



My First ChatBot

BOUSSOURA Rayane
ZHANG Ludovic
Groupe F



Introduction

Le projet que nous avons réalisé porte sur l'analyse de texte et vise à développer un système basé sur la méthode de TF-IDF pour répondre à des questions en se basant sur les fréquences des mots dans un corpus de discours présidentiels français. Cela nous permet d'avoir un aperçu des concepts fondamentaux utilisés dans le traitement de texte et sert de base pour la création de chatbots.

Notre objectif principal est de concevoir un système qui, à partir d'un ensemble de discours présidentiels, peut répondre de manière intelligente à des questions en utilisant la fréquence des mots dans le corpus, nécessitant plusieurs étapes tels le filtrage et le traitement du texte, l'analyse de nombres de caractères,...Le travail demandé est divisé en trois parties.

Dans la première partie, des fonctions de base sont développées pour comprendre le contenu des fichiers fournis.

- Dans la première partie, des fonctions de base sont développées pour comprendre le contenu des fichiers fournis.
- La deuxième partie se concentre sur le calcul de la matrice de similarité et la génération automatique de réponses en utilisant la méthode TF-IDF. Cela implique la recherche des mots de la question dans le corpus, le calcul du vecteur TF-IDF pour les termes de la question, le calcul de la similarité, et enfin la génération de la réponse.
- La troisième partie nous conseil certaines améliorations et des fonctionnalités bonus, telles que le nettoyage avancé du texte, le traitement des verbes conjugués, l'enrichissement du corpus, et la généralisation de l'application pour couvrir divers thèmes.

En résumé, ce projet offre une opportunité d'explorer et de mettre en pratique des concepts clés du traitement de texte, tout en permettant de développer un système interactif capable de répondre à des questions en se basant sur l'analyse de la fréquence des mots dans un corpus spécifique.

Programme et Fonctionnalités

Le programme est fait de sorte à ce que l'utilisateur n'ait aucun problème d'utilisation ou de compréhension du code et puisse utiliser un menu facilement.

```
print("          Bienvenue dans le tchatbot: Partie 1")
print("Veuillez respecter les règles suivantes pour une meilleure expérience.")
print("-Taper 1: Afficher la liste des mots avec le score tf-idf le moins élevé")
print("-Taper 2: Afficher la liste de mot avec le score tf-idf le plus élevé")
print("-Taper 3: Indiquer le(s) mot(s) le(s) plus répété(s) par le président Chirac")
print("-Taper 4: Indiquer le(s) nom(s) du (des) président(s) qui a (ont) parlé de la « Nation » et celui qui l'a répété le plus de fois")
print("-Taper 5: Indiquer le premier président à parler du climat et/ou de l'écologie")
print("-Taper 6: Afficher quel(s) est(sont) le(s) mot(s) que tous les présidents ont évoqué(s)")
print("Le chargement peut-être un peu long merci de prendre cela en considération")
valeurs_numerique = int(input("Saisissez des chiffres entre 1 à 6 pour accéder au diverse fonctionnalité du tchatbot: "))
```

Si le tiers souhaite comprendre le fonctionnement du Chatbot il lui est aussi possible de consulter le code ainsi que les commentaires.

```
# Lecture du discours original
# Écriture du discours nettoyé dans le fichier Cleaned
```

Le programme possède plusieurs fonctionnalités simples d'utilisation tels que:

- L'affichage des listes de mots avec le score TF-IDF (que nous allons expliquer incessamment sous peu) le plus élevé.
- L'affichage des listes de mots avec le score TF-IDF le plus faible.
- Indiquer quel est le mot le plus souvent répété par le président Jacques Chirac.
- Chercher la liste des présidents qui ont parlé de la « Nation » et celui qui en a le plus parlé
- Trouver quelle est le premier président qui a parlé du climat et/ou de l'écologie.
- Afficher le ou les mots que les présidents ont le plus évoqué tout au long de leurs discours.

A cela se rajoute la sélection du document le plus pertinent par rapport à une question donnée

Fonctionnalités et TF-IDF

TF-IDF permet de représenter de manière numérique la pertinence d'un mot dans un document par rapport à l'ensemble du corpus, facilitant ainsi l'analyse et le traitement de texte.

TF (Term Frequency) :

Cela mesure la fréquence d'un terme dans un document spécifique. Le score TF élevé si un terme apparaît fréquemment dans le document. On calcule alors en comptant le nombre d'occurrences de chaque mot dans un texte.

La fonction à écrire : Prend une chaîne de caractères en paramètre et retourne un dictionnaire associant à chaque mot le nombre de fois qu'il apparaît.

IDF (Inverse Document Frequency):

Cela mesure l'importance d'un terme dans l'ensemble du corpus de documents. Le score IDF plus faible pour les termes fréquents dans de nombreux documents, plus élevé pour les termes rares. Puis, Calculer en prenant le logarithme décimal de l'inverse de la proportion de documents dans le corpus qui contiennent ce mot.

La fonction à écrire prend le répertoire où se trouvent les fichiers du corpus en paramètre et retourne un dictionnaire associant à chaque mot son score IDF.

```
def tf(texte): #Cette fonction prend en
    textel = texte
    dico = {}
    i = 0
    while i < len(textel):
        if textel[i] == " ":
            if textel[0:i] in dico:
                dico[textel[0:i]] += 1
            else:
                dico[textel[0:i]] = 1
            textel = textel[i+1:]
            i = 0
        else:
            i += 1
    return dico
```

```
def IDF(repertoire): # Cette fonction prend en entrée le chemin
    dic_idf = {}
    frequence_doc = {} # Dictionnaire pour stocker la fréquence
    lst = list_of_files(repertoire, ".txt")
    texte = ""
    lst_termes = recupere_les_termes(repertoire) # Liste des
    for el in lst_termes: # Initialisation de la fréquence
        frequence_doc[el] = 0
    for i in range(len(lst)): # Parcours de chaque fichier
        with open(repertoire + "/" + str(lst[i]), 'r') as f1:
            f_text = f1.readlines()
            texte += f_text + " " # Concaténation du texte du fichier
        doc = tf(texte) # Calcul des fréquences des termes
        for el in doc.keys():
            if el in frequence_doc.keys():
                frequence_doc[el] += 1 # Mise à jour de la fréquence
        for key, values in frequence_doc.items(): # Calcul du
            dic_idf[key] = math.log10((len(lst) / values)+1)
    return dic_idf
```

```
def tf_idf(directory):
    idf_dico = IDF(directory) # Calcul des valeurs IDF pour chaque
    tfidf_matrix = []
    lst_aux = [] # Liste auxiliaire pour stocker les valeurs TF-IDF
    lst = list_of_files(directory, ".txt") # Liste des noms de fichiers
    all_words = recupere_les_termes(directory)
    for mot in all_words:
        for j in range(len(lst)):
            with open(directory + "/" + str(lst[j]), 'r') as f1:
                f_text = f1.readlines()
                doc = tf(f_text) # Calcul des fréquences de chaque terme
                if mot in doc.keys():
                    lst_aux.append(doc[mot] * idf_dico[mot]) # Calcul
            else:
                lst_aux.append(0) # Si le terme n'est pas présent
    tfidf_matrix.append(lst_aux)
    lst_aux = [] # Réinitialisation de la liste auxiliaire
    return tfidf_matrix
```

Les différentes fonctions et algorithmes



Liste de fichiers par extension : L'algorithme récupère la liste des fichiers dans un répertoire avec une extension spécifiée.

Extraction du nom du président : L'algorithme extrait le nom du président à partir du nom du fichier.

Suppression des numéros, Nettoyage des caractères spéciaux et mise en minuscules, Suppression des doublons dans une liste Correspond au traitement du texte

Calcul des fréquences des termes (TF) : L'algorithme génère un dictionnaire des fréquences de chaque mot dans un texte.

Récupération des termes uniques : L'algorithme extrait une liste de tous les termes uniques dans les fichiers texte d'un répertoire.

Calcul de l'IDF (Inverse Document Frequency) : L'algorithme calcule les valeurs d'IDF pour chaque terme dans les fichiers d'un répertoire.

Calcul de la matrice TF-IDF : L'algorithme génère une matrice TF-IDF pour les documents dans un répertoire.

Tokenisation : L'algorithme divise un texte en tokens tout en éliminant certains caractères.

Intersection de listes : L'algorithme trouve les termes communs entre deux listes.

Calcul du vecteur TF-IDF : L'algorithme génère le vecteur TF-IDF pour un texte donné et un répertoire spécifié.

Similarité cosinus : L'algorithme mesure la similarité cosinus entre deux vecteurs.

Sélection du document le plus pertinent : L'algorithme identifie le document le plus pertinent par rapport à une question donnée.

(Identification du terme le plus significatif : L'algorithme trouve le terme avec la valeur TF-IDF la plus élevée.)

Recherche d'une occurrence spécifique : L'algorithme identifie une occurrence spécifique dans un texte.

Gestion des données et Structure des choix

Les structures de données choisies sont appropriées pour les tâches spécifiques du projet, telles que le stockage de noms de fichiers, la gestion de fréquences de termes, le calcul de valeurs IDF, etc. Ces choix facilitent la manipulation et le traitement des données nécessaires à l'analyse des discours présidentiels.

Listes :

On les retrouve dans diverses fonctions telles que `list_of_files`, `supprime_doublon`, `associate_president_first_name`, `recupere_les_termes`, et d'autres encore. Ces listes sont employées pour organiser des ensembles ordonnés d'éléments. Elles se révèlent bien adaptées à la représentation de séquences d'informations, comme les noms de fichiers ou les termes uniques, par exemple.

Dictionnaires :

Ils sont mis en œuvre dans des fonctions telles que `tf` et `IDF` pour enregistrer des paires clé-valeur. Les dictionnaires s'avèrent être le choix idoine pour décrire les associations entre les termes et leurs fréquences, ou encore entre les termes et leurs valeurs d'IDF.

Ensembles (Sets) :

Bien qu'ils ne soient pas explicitement utilisés dans le code fourni, l'utilisation potentielle d'ensembles (Sets) pourrait s'avérer pertinente pour représenter des ensembles de termes uniques sans duplication.

Listes auxiliaires :

Ces dernières sont mises à contribution dans des fonctions comme `lst_aux` dans `tf_IDF`, servant à stocker provisoirement les valeurs TF-IDF pour chaque document. Les listes se révèlent être des structures efficaces pour gérer des séquences dynamiques d'informations.

```
d = {"n  
d["nom"  
re'  
d["age"  
  
d["spor  
ai']
```

```
ain.py  
  
result = [(  
    range(1,  
    range(x,  
    range(y,  
    if x**2
```

Difficulté et problèmes rencontrés

Partie I

Dans un premier temps, la difficulté était la compréhension de certaines consignes mais avec la continuité de l'énoncé cela c'est clarifié.

La fonction IDF s'avérer compliquer lorsque que nous testions notre programme. Il fallait prendre le mot, chercher la fréquence d'apparition pour chaque fichier. Après de nombreuse vérification le problème se situais simplement dans la mauvaise organisation des listes/dictionnaires.

Le sujet à changer quelque fois, lorsque la formule avec $\log() + 1$ a était modifier et le « 1 » disparu l'entièreté de TF-IDF ne fonctionnais plus.

$$IDF(mot) = \log_{10} \left(\frac{\text{Nombre total de documents dans le corpus}}{\text{Nombre de documents du corpus contenant le mot}} \right)$$

Enfin, notre dernier problème de la partie 1 était le comptage de mots, au lieu de trouver 1681 mots nous trouvions 1692. Après quelques modifications dans les algorithmes de traitement du texte (notre boucle s'arrêtait trop tôt oubliant certains espaces) .

Partie II

L'algorithme nous permettant de trouver le pertinent fichier ne trouvais pas le bon fichier. En effet, le problème était surement dû à une mauvaise compréhension de la consigne, mais nous pensons que le problème est plus lié avec la matrice ou le vecteur tf-idf qui renvoie un résultat que la fonction n'arrive pas à traiter.

Le résultat censé être obtenue était climat mais la fonction nous renvoyais autre chose.

Résultat

- Interface de base

```
Bienvenu dans le tchatbot
Veuillez respecter les règles suivantes pour une meilleurs expérience.
-Taper 1: Afficher la liste des mots avec le score tf-idf le moins eleve
-Taper 2: Afficher la liste de mot avec le score tf-idf le plus eleve
-Taper 3: Indiquer le(s) mot(s) le(s) plus répété(s) par le président Chirac
-Taper 4: Indiquer le(s) nom(s) du (des) président(s) qui a (ont) parlé de la « Nation » et celui qui l'a répété le plus de fois
-Taper 5: Indiquer le premier président à parler du climat et/ou de l'écologie
-Taper 6: Afficher quel(s) est(sont) le(s) mot(s) que tous les présidents ont évoqués
-Taper 7: Si vous voulez posez une question au robot
-Taper 8: Pour fermer la fenetre
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: |
```

- Entré sécurisé en cas de fausse manipulation

```
Bienvenu dans le tchatbot
Veuillez respecter les règles suivantes pour une meilleurs expèrience.
-Taper 1: Afficher la liste des mots avec le score tf-idf le moins eleve
-Taper 2: Afficher la liste de mot avec le score tf-idf le plus eleve
-Taper 3: Indiquer le(s) mot(s) le(s) plus répété(s) par le président Chirac
-Taper 4: Indiquer le(s) nom(s) du (des) président(s) qui a (ont) parlé de la « Nation » et celui qu
-Taper 5: Indiquer le premier président à parler du climat et/ou de l'écologie
-Taper 6: Afficher quel(s) est(sont) le(s) mot(s) que tous les présidents ont évoqués
-Taper 7: Si vous voulez posez une question au robot
-Taper 8: Pour fermer la fenetre
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 65595
Cette fonctionnalité n'existe pas !
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: |
```

- Différentes commandes

```
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 2
Le mot avec le score tf-idf le plus eleve est : ['de']
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 3
Le mot le plus répété par Chirac est: de
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 4
Le(s) nom(s) du (des) président(s) qui a (ont) parlé de la « Nation est : ['Chirac', 'Hollande', 'Macron', 'Mitterrand']»
Et celui qui la répété le plus de fois est : Chirac
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 5
Le premier président à parler du climat et/ou de l'écologie est : Macron
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 6
les mots dits « non importants », quel(s) est(sont) le(s) mot(s) que tous les présidents ont évoqués sont : ['messieurs', 'les', 'mesdames', 'en', 'ce', 'je', 'la']
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: |
```

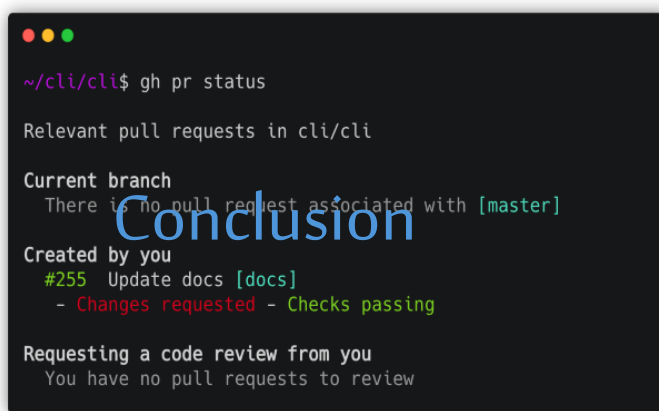
```
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 3
Le mot le plus répété par Chirac est: de
```

```
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 1
Saisissez la taille de la liste souhaité: 5
La liste des mots de taille 5 avec le score tf-idf les moins élevés sont: ['officiellement', 'fonctions', 'traversé', 'relevé', 'parlé']
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: |
```

```
Avez-vous une question ? Peux-tu me dire comment la 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
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot:
```

```
Saisissez des chiffre entre 1 à 8 pour accéder au diverse fonctionnalité du tchatbot: 8

Process finished with exit code 0
|
```

```
~/cli/cli$ gh pr status

Relevant pull requests in cli/cli

Current branch
  There is no pull request associated with [master]

Created by you
  #255 Update docs [docs]
    - Changes requested - Checks passing

Requesting a code review from you
  You have no pull requests to review
```

Ce projet nous a apporté beaucoup de choses sur divers plans aussi bien technique que sur d'autres aspects essentiels de la réalisation de logiciel.

Sur le plan Technique :

Un des points que nous avons appris est le traitement de texte avancé. La mise en œuvre de la méthode TF-IDF a permis de mieux comprendre les techniques complexes du traitement de texte, tels que le calcul des fréquences ou la mesure de similarité.

La manipulation de fichiers en Python est un point essentiel que nous avons pu traiter et améliorer. Le projet a nécessité une manipulation efficace des fichiers en utilisant des modules tels que "os".

Le calcul matriciel qui est une chose presque nouvelle pour nous, nous a permis d'apprendre à créer et la manipuler des matrices pour représenter les données textuelles ont contribué à renforcer la compréhension des calculs matriciels et de la transposée.

Sur le plan de gestion du travail :

La répartition des tâches au sein de l'équipe a été essentielle pour la réussite du projet. La communication régulière nous a permis de résoudre rapidement certains problèmes.

L'obligation d'utiliser Git a souligné l'importance de la gestion de versions, malgré notre difficulté à l'utiliser au départ. Cela a facilité la collaboration, le suivi des modifications nous faisant économiser du temps de discussions et la résolution d'incompréhension.

La gestion du temps a joué un rôle essentiel, surtout avec la division du projet en trois parties. Établir un plan détaillé a été important pour respecter les délais et tirer avantages des ressources à notre disposition. Apprendre à reconnaître les tâches prioritaires et à les traiter a été une compétence majeure que nous avons amélioré durant notre projet.

Possible point d'amélioration:

-Une possible meilleurs gestion du temps aurait pu nous permettre la réalisation de la partie III.

-L'utilisation plus fréquente de GitHub, en raison de certains problèmes techniques, nous avons privilégié l'échange de codes par messages.

-L'ajout de certaine fonctionnalité bonus pour améliorer l'expérience de l'utilisateur.