18/12/2023

Une image contenant Police, logo, symbole, Graphique

Description générée automatiquement

*Projet réalisé par*

Alexandre LE GALLIARD

Alexandre DAUVEL

Mathis GOSSELIN

*Projet encadré par*

Ibtihel BEN LTAIFA

Soufia BENNA

**“My first chatBot”**

*Rapport de projet L1 B*

# **I - Introduction :** ………………………………………………………………………………

# **II - Les fonctionnalités de base :** ……………………………………………………….

## Extraction des noms des présidents à partir des noms de fichiers : …………………

## Association des prénoms à chaque président : ……………………………………………

## Affichage de la liste des noms des présidents (Gestion des Doublons) : …………..

## Conversion des textes en minuscules et stockage dans un nouveau dossier : ……

1. Traitement de la ponctuation dans les fichiers du répertoire « Cleaned » : …………

# **III - Les fonctionnalités proposées à développer :** .......................................

1. Affichage des mots moins Importants dans le corpus : ………………………………..
2. Affichage du mot avec le score TF-IDF le plus élevé : …………………………...……..
3. Mots les plus répétés par le président Chirac (hors mots non importants) :
4. Président(s) parlant de la « Nation » et nombre de répétitions le plus élevé :
5. Président(s) parlant du climat et/ou de l'écologie : ………………………....................
6. Tokenisation de la Question : ……………………………………………………………………
7. Recherche des Mots de la Question dans le Corpus : …………………………………..
8. Calcul du vecteur TF-IDF pour les termes de la question : …………………………….
9. Calcul de la similarité : …………………………………………………………………………..
10. Calcul du document le plus pertinent : ……………………………………………………..

**IV – Difficultés rencontrées :** …………………………………………………………… **VI - Conclusion :** …………………………………………………………………………….

**I - Introduction :**

## Le projet que nous proposons **« My first chatBot »**, se concentre sur l'analyse de texte, offrant une opportunité d'explorer des concepts fondamentaux du traitement de texte. L'objectif ultime est de fournir une compréhension approfondie d'une méthode utilisée dans le développement de chatbots et d'intelligences artificielles génératives, tels que le modèle ChatGPT. Contrairement à la manipulation directe de réseaux de neurones, notre attention se porte sur une approche basée sur le nombre d'occurrences des mots. Cette méthode nous permettra de créer un système capable de générer des réponses intelligentes en répondant à des questions à partir d'un corpus de textes. Dans ce contexte, notre démarche met l'accent sur la fréquence des mots dans le corpus comme clé pour élaborer des réponses pertinentes.

**II - Les fonctionnalités de base :**

1. Extraction des noms des présidents à partir des noms de fichiers :

## 

Cette fonction **recup\_nom**, traite une liste de noms de fichiers représentant les nominations des discours présidentiels, en extrayant et nettoyant les noms des présidents. Les résultats sont stockés dans une liste **presidents\_name**, simplifiant ainsi la récupération des noms pour une analyse ultérieure des discours.

Résultat :



1. Association des prénoms à chaque président :



La fonction **name\_association** associe un prénom à chaque nom de président dans la liste **presidents\_name.** En parcourant la liste, elle effectue des associations spécifiques en fonction du nom du président, telles que "Jacques" pour "Chirac", "François" pour "Hollande", etc. La liste modifiée est ensuite renvoyée, simplifiant ainsi l'association des prénoms aux noms des présidents pour une utilisation contextuelle ultérieure.

Résultat :

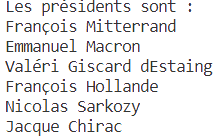


1. Affichage de la liste des noms des présidents (Gestion des Doublons) :



La fonction **presidents\_name\_display** affiche les noms des présidents de la liste **presidents\_name.** Après une déclaration introductive, elle parcourt un ensemble (pour éliminer les doublons) des noms et les affiche individuellement. En résumé, cette fonction simplifie la visualisation claire et concise des noms des présidents extraits et traités.

Résultat :

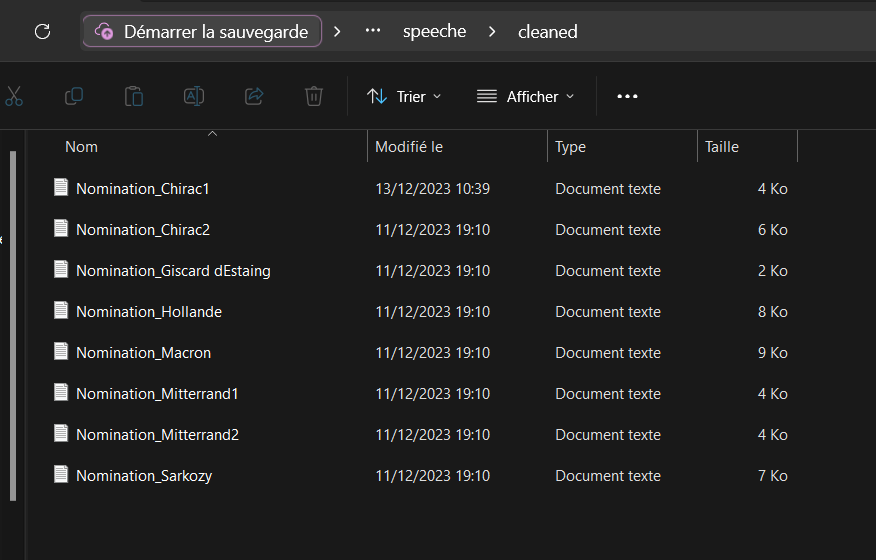


1. Conversion des textes en minuscules et stockage dans un nouveau dossier :



## La fonction **cleaning\_files** parcourt chaque fichier dans un dossier spécifié, convertit les majuscules en minuscules et supprime la ponctuation du contenu de chaque fichier, puis réécrit le contenu modifié dans le même fichier. Cette fonction facilite le nettoyage des fichiers en vue d'une analyse ultérieure.

Résultat :



1. Traitement de la ponctuation dans les fichiers du répertoire « Cleaned » :

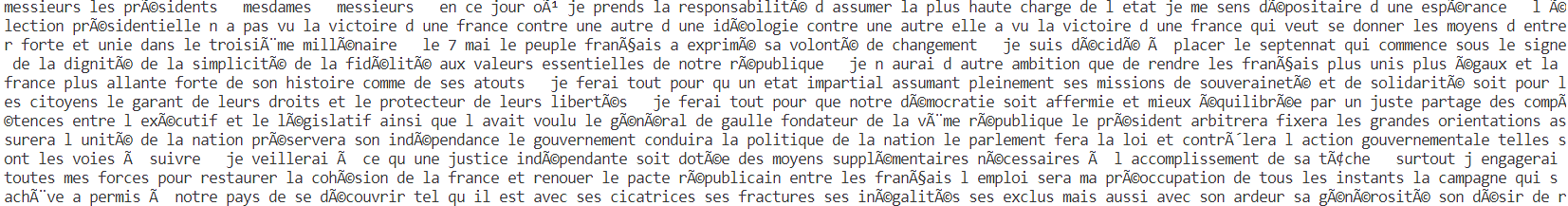




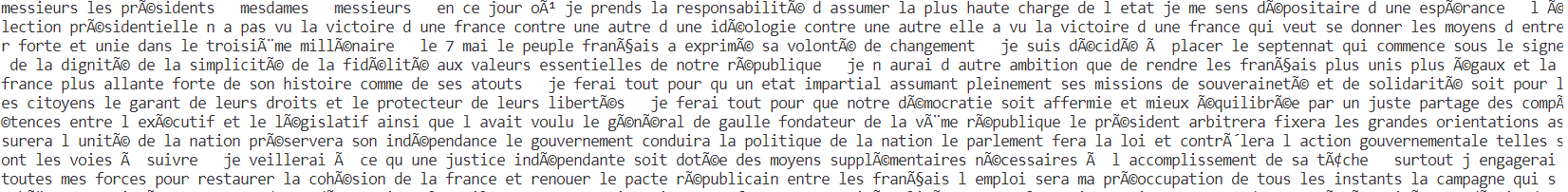
La fonction **suppression\_ponctuation** élimine la ponctuation d'une chaîne en remplaçant certains caractères par des espaces ou en les supprimant.

La fonction **convertir\_majuscule\_en\_minuscules** permet de transformer les caractères majuscules en minuscules.

Résultat après **suppression\_ponctuation** :



Résultat après **convertir\_majuscule\_en\_minuscules** :



## **III - Les fonctionnalités proposées à développer :**

1. Affichage des mots moins Importants dans le corpus :



La fonction **score\_tfidf\_plus\_faible** identifie les mots ayant le score TF-IDF le plus faible dans une matrice donnée en tenant compte des valeurs IDF associées. Elle simplifie cette tâche en parcourant la matrice, calculant les scores TF-IDF et ajoutant les mots avec un score nul à la liste des moins importants.

Résultat :



1. Affichage du mot avec le score TF-IDF le plus élevé :



La fonction **mots\_max** identifie et affiche-le(s) mot(s) ayant le score TF-IDF le plus élevé dans une matrice donnée, en tenant compte des valeurs IDF associées. Elle simplifie cette tâche en parcourant la matrice, calculant les scores, et affichant le résultat.

Résultat :



1. Mots les plus répétés par le président Chirac (hors mots non importants) :



La fonction **mots\_plus\_répétés\_par\_président** identifie et affiche-le(s) mot(s) le(s) plus répété(s) par un président spécifié. Elle parcourt la liste des noms de présidents, trouve les mots les plus répétés pour le président donné, et affiche le résultat. Cette fonction facilite la recherche des mots les plus fréquemment utilisés par un président particulier dans un ensemble de discours.

Résultat pour Macron :

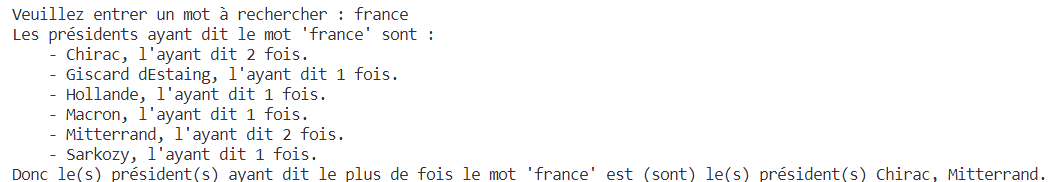


1. Président(s) parlant de la « Nation » et nombre de répétitions le plus élevé :



La fonction **mots\_dit\_par\_présidents** identifie et affiche les présidents qui ont prononcé un mot spécifié, ainsi que le nombre de fois où ce mot a été utilisé. Elle simplifie cette tâche en parcourant la liste des noms de présidents, comptant les occurrences du mot dans les discours (**tf\_list**), et présentant clairement les résultats, y compris le président ayant le plus prononcé le mot.

Résultat pour le mot « france » :



1. Président(s) parlant du climat et/ou de l'écologie :



La fonction **premier\_president\_dire\_theme** permet d'identifier le premier président qui a abordé un thème spécifique dans les discours présidentiels. Elle recherche le thème au sein des discours, affiche le nom du premier président l'ayant abordé, ou indique que le thème n'a pas été évoqué par aucun président.

Résultat pour le mot « climat » :



1. Tokenisation de la Question :



La fonction **tokenisation** prend en entrée un contenu, le convertit en minuscules, supprime la ponctuation, puis le divise en mots individuels en utilisant l'espace comme délimiteur. Elle renvoie une liste de ces mots, constituant ainsi une tokenisation du contenu.

Résultat pour la question « Peux-tu me dire comment trouver une solution pour réduire le climat ? » :



1. Recherche des Mots de la Question dans le Corpus :



La fonction **presence\_mot** évalue la présence de chaque mot dans une liste de contenu par rapport à un dictionnaire d'inverse document fréquence (idf). Elle crée un dictionnaire où chaque mot du contenu est associé à une valeur booléenne indiquant sa présence dans le dictionnaire idf. Le résultat est affiché sous la forme d'un dictionnaire.

Résultat pour la même question :



1. Calcul du vecteur TF-IDF pour les termes de la question :



La fonction **tf\_idf** crée une matrice TF-IDF en utilisant les fréquences de termes (TF) d'une question spécifique et les inverses des fréquences documentaires (IDF) de tous les mots dans un dossier de documents. Chaque ligne de la matrice représente un document du corpus, chaque colonne correspond à un terme du dictionnaire IDF. Les valeurs des cellules sont calculées en multipliant la fréquence du terme dans le document par son IDF, et la matrice résultante est renvoyée.

Résultat :



1. Calcul de la similarité :







La fonction **scalaire** prend deux vecteurs *A* et *B* en entrée et retourne leur produit scalaire, qui est la somme des produits des éléments correspondants des deux vecteurs.

La fonction **norme** prend un vecteur *A* en entrée et retourne la norme Euclidienne du vecteur, qui est la racine carrée de la somme des carrés de ses composantes.

La fonction **similarite** prend deux matrices *A* et *B* en entrée, calcule la similarité entre chaque paire de vecteurs correspondants, puis retourne l'indice du fichier ayant la plus grande similarité avec la question posée. Si la similarité maximale est nulle, elle indique qu'aucun document ne correspond à la question.

Résultat :



1. Calcul de la similarité :



La fonction **plus\_pertinent** permet de calculer, d’après une liste de valeurs de similarité selon la fonction **similarite** en paramètre, le document du corpus le plus similaire à la question posée.

Résultat :



**IV - Difficultés rencontrées :**

Pendant le projet, nous avons rencontré plusieurs problèmes techniques en raison du manque d'aide, des consignes parfois peu claires, et des changements dans les consignes. Trouver des informations pour surmonter ces problèmes a été difficile en raison du manque de ressources d'aide spécifiques. Les consignes qui n'étaient pas très claires ont provoqué différentes interprétations, soulignant l'importance de discuter régulièrement avec les enseignants pour comprendre ce qui était attendu. Les contradictions dans les consignes ont créé de la confusion, mais nous avons réussi à obtenir des instructions claires en communiquant activement avec les enseignants. Nous avons dû ajuster fréquemment le projet en raison de modifications des consignes, ce qui a nécessité une adaptation rapide. Les problèmes techniques avec PyCharm ont demandé des recherches approfondies et, parfois, l'utilisation d'autres outils de développement. Respecter les délais a nécessité une planification minutieuse des tâches et, dans certains cas, des demandes de délai supplémentaire discutées avec les enseignants. Nous avons réussi à relever ces défis grâce à une approche proactive pour résoudre les problèmes, une communication ouverte et une gestion attentive du temps. En outre, nous avons dû utiliser GitHub sur un seul ordinateur, celui d'Alexandre Dauvel, en raison de soucis techniques. Nous avons réussi à surmonter ces problèmes en résolvant activement les difficultés, en communiquant ouvertement, en gérant le temps avec attention, et en adoptant des solutions alternatives, comme l'utilisation de GitHub sur un seul PC.

**V - Conclusion :**

En fin de compte, malgré de nombreux problèmes avec ce projet, que ce soit la communication, l’aide qui nous a été apporté ou le manque de temps, ce projet comporte tout de même des avantages. Il nous a permit d’acquérir des compétences pratiques dans le domaine du traitement de texte et de renforcer des qualités essentielles comme la résolution de telles problèmes. Ces enseignements seront peut-être précieux et utiles dans la poursuite de projets futurs et dans le développement de compétences liées à l’intelligence artificielle et aux Chatbots.