**Classification de personnages dans la trilogie originale Star Wars :**

**Introduction :**

La saga Star Wars est constituée de 3 trilogies, qui font intervenir de nombreux personnages, que l'on peut distinguer selon leur espèce, leur genre, ou encore leur appartenance à un groupe. Pour chacun des trois épisodes de la trilogie originale, nous avons pour données les dialogues ainsi que des caractéristiques concernant les personnages présents.

**Problématique :**

L'objectif de ce travail est de mettre en évidence les principaux indicateurs (âge, genre, champs lexicaux utilisés) qui influencent le nombre d'intervention et le temps de parole des personnages. On pourra en déduire si la répartition du temps de parole est équilibrée ou pas. Enfin, on essayera de répondre à la question suivante : d'après l'analyse des données de la première trilogie, aurait-on pu prédire que le personnage principal de la dernière trilogie était une femme ?

**Pré-traitement des données :**

La première étape de ce projet sera de prendre en main les données (brutes) des scripts de la saga, et d'opérer les premiers traitements permettant de les transformer en données exploitables automatiquement.

Ici notre base de données n'est pas un dataframe, il s'agit d'un texte composé de noms des personnages, de dialogues et de didascalies. On s'intéresse uniquement aux personnages et à leurs dialogues, donc on va d'abord faire le tri afin d'obtenir des données structurées faciles à exploiter.

Nous avons remarqué que les discours des personnages sont décalés de 25 espaces, suivis d'un espace, c'est pour cela qu'on a défini 3 groupes dans la fonction re.compiler. On a ensuite appliqué la fonction pattern.finditer à notre texte de base pour demander le groupe 2 en sortie, ce qui nous a permis d'obtenir la partie texte qui nous intéresse notamment les noms des personnages suivis de leurs discours. Les librairies pandas et numpy sont très pratiques pour ces manipulations.

Pour cela, on a créé un 'pattern' pour chercher toutes les chaines de caractères qui y correspondent. Par une loupe nous avons créé une liste de toutes les chaines de caractères qui correspondent à notre pattern. Dans un premier temps, nous avons supprimé mannuellement les 11 premières lignes qui nous n'intéressent pas car elles correspondent à des didascalies. Nous avons remarqué que les prénoms des personnages ont une structure spéciale, ils sont précédés par 13 ou 14 espaces, nous avons donc créé un 'pattern' pour les récupérer. Nous avons ensuite regroupé les prénoms des personnages dans une liste. Tous les éléments qui ne sont pas dans cette liste sont des répliques des personnages, ce qui nous a permis de récupérer les répliques des personnages.

Pour la concaténation des paroles, nous avons utilisé la fonction join() qui permet la concaténation des chaines de caractères dans une liste. Ainsi on obtient un tableau composé des noms de personnages et leurs dialogues correspondant, il s’agit d’un tableau de 2 colonnes et 1002 lignes. Afin d’automatiser cette démarche et pouvoir l’appliquer aux autres bases de données nous avons créé une fonction qui fait tout le travail car il est très pénible, voire impossible, de supprimer toutes les lignes inutiles manuellement.