Projet algorithmique et programmation avancée : Sujet n°2

Création d'un graphe de cooccurrence de mots extrait de tweets sur Netflix

Table des matières

Introduction :

Dans ce projet nous allons extraire des données via un site internet pour avoir en notre possession un corpus de données qui seront plusieurs textes. L'objectif étant de produire un graphe de cooccurrence de ce corpus pour avoir les liens entre les différents mots utilisés dans les textes.

Ici, nous allons prendre des données via le réseau social Twitter. Notre corpus sera constitué d'un nombre définit de Tweet (post) qui parlerons de « Netflix ». Netflix étant une plateforme de streaming de séries, films, documentaires… Une fois que la matrice de cooccurrence sera créée nous pourrions observer des mots revenants plus de fois que d'autres et donc établir une tendance ou alors des groupes de mots fréquemment utilisés et ainsi créer un graphe de cooccurrence.

Notre code est disponible sur Github à l'adresse suivante :

https://github.com/Hugobross/Graph_coocurrence

Extraction des données :

Afin d'avoir les données de Twitter, nous avons utilisé une API qui va nous permettre d'interroger la base de données de Twitter et avoir les différents tweets selon certaines caractéristiques que nous expliciterons.

Pour utiliser cette API sous Python, nous avons utilisé la librairie « Tweepy ». Une fois que nous avons nos identifiants et les tokens d'accès, nous pouvons produire notre requête pour avoir les données que nous souhaitons.

Dans la requête, nous prenons les 50 tweets les plus récents comprenant le mot « netflix », étant dans la langue française et qui ne sont ni des retweets ni des réponses à des tweets (pour éviter d'avoir les mêmes tweets retweetés).

Une fois que les données sont chargées, nous mettons les tweets dans une liste et ce sera notre corpus. Il est important de noter que le corpus est évolutif puisqu'il prend les tweets les plus récents au moment où la requête est effectuée.

Nettoyage des données :

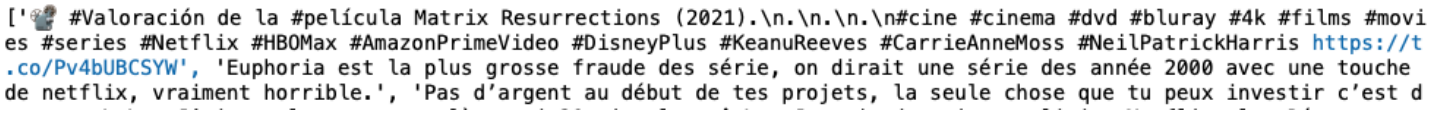
Pour créer le graphe de cooccurrence, nous devons tout d'abord produire une matrice carrée qui aura pour lignes et colonnes tous les mots contenus dans le corpus.

Mais avant de produire cela, nous devons nettoyer les données. Cela va correspondre à enlever tous les caractères spéciaux, mais aussi les « stop-words » pour ne pas être pollué par les mots de liaison ou encore les pronoms personnels. Nous avons également enlevé tous les emojis souvent présents dans les tweets.

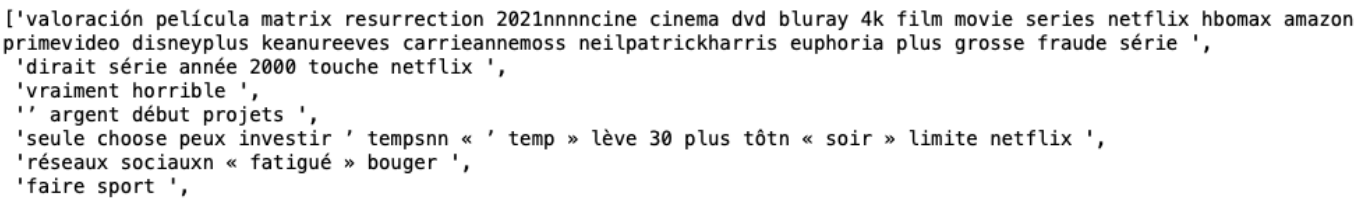
La fonction « cleaning » se charge de l'ensemble de ses opérations.

Voici ce que nous permet par exemple de faire notre fonction « cleaning » :

Avant nettoyage :



Après nettoyage :



Certes, le texte ne veut plus rien dire mais c'est pour les besoins de notre analyse.

Nombre d'occurrences

Nous pouvons ensuite observer les mots les plus utilisés dans notre texte pour se donner un aperçu des mots nettoyés qui sont les plus liés à notre sujet.

Word	Frequency
netflix	44
'	30
film	8
série	6
ça	5
saison	4

Création de la matrice de cooccurrence

Un fois que nous avons effectué toutes ces étapes nous pouvons construire une matrice carrée avec l'ensemble des mots nettoyés. Pour cela nous avons repris une fonction à l'adresse suivante : https://stackoverflow.com/questions/35562789/how-do-i-calculate-a-word-word-co-occurrence-matrix-with-sklearn

Nous avions réussi à créer une matrice de cooccurrence mais nous n'avions pas les mots en colonne et en ligne mais leur position dans le texte avec un chiffre.

L'avantage de cette fonction est qu'elle nous permet d'avoir les mots à l'intérieur de la matrice et non pas seulement leur position en « integer ». Cela nous permettra d'avoir les mots sur le graphe et non pas une valeur qu'on ne peut pas interpréter. C'est pour cela que nous avons fait le choix d'utiliser cette fonction.

Voici un exemple de matrice de cooccurrence que nous obtenons :

	0	1	11hj	13	16h	19	1997n	2	2006	2022	...	ça	écrans	émotion	être	'	😌	✅	❤	🤡🤡	👨u200dnntout
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	1	0	0	...	1	0	0	0	1	0	0	0	0	0
11hj	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
16h	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
😌	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
✅	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
❤	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
🤡🤡	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0
👨u200dnntout	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0

414 rows × 414 columns

Le résultat de cette matrice met particulièrement en avant 2 problèmes que nous n'avons pas réussi à régler. En effet, même si la majorité des « emojis » sont éliminés par la fonction « cleaning », certains restent dans notre texte. De plus, nous n'avons pas réussi à enlever le caractère suivant « ' ».

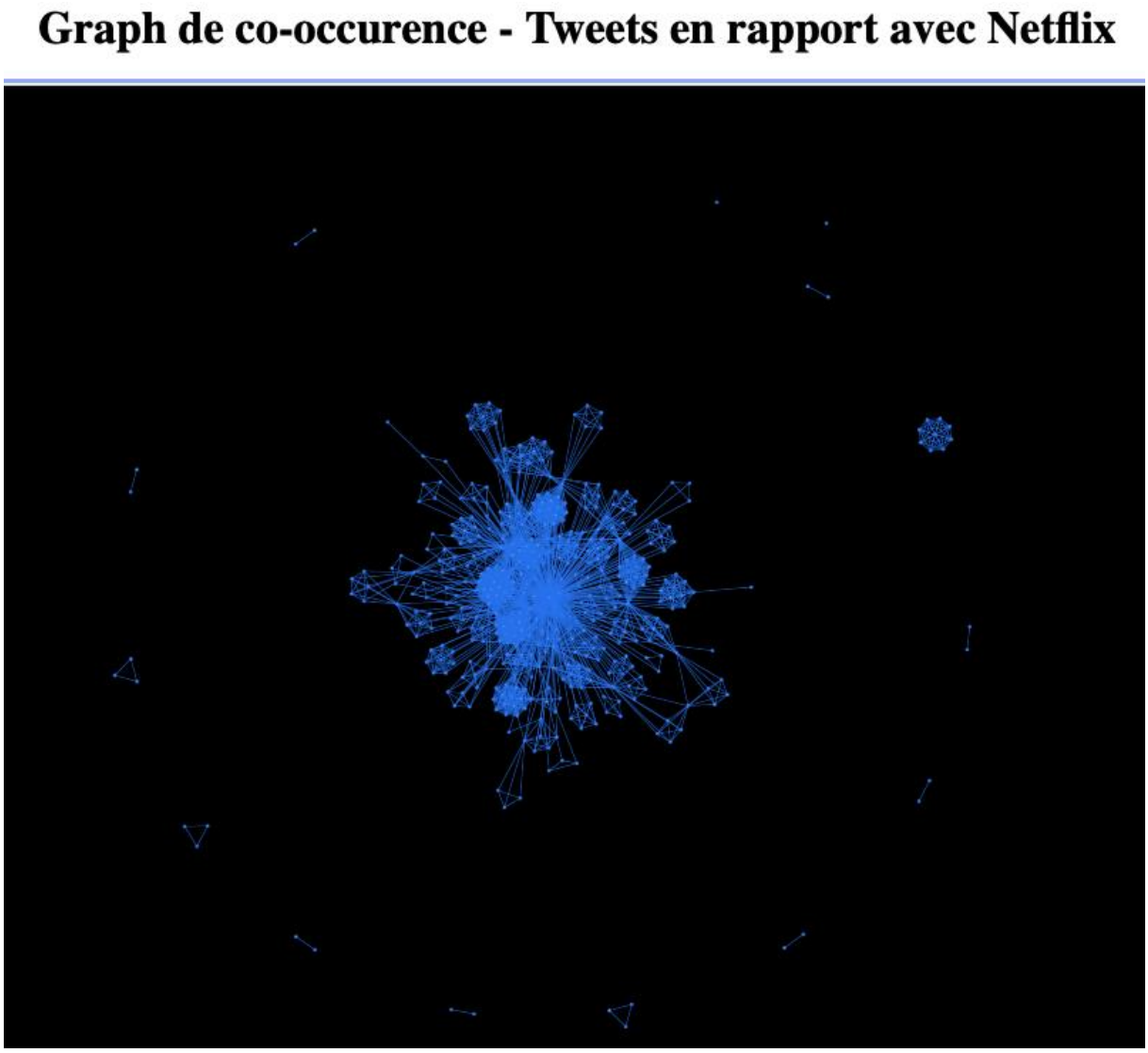
La matrice contient les mots ordonnés, elle met donc en exergue ces 2 problèmes mais la grande majorité de nos mots ont été nettoyés correctement.

Création du graphe de cooccurrence

Une fois que nous avons obtenu notre matrice, nous pouvons réaliser notre graphe de cooccurrence. Pour cela, nous avons utilisé les librairies « networkx » ainsi que « pyvis.network ».

Plus précisément nous avons utilisé la classe « Network » qui est disponible dans la librairie « pyvis.network » afin de permettre une visualisation du graphe en html. En effet, cette classe « Network » est au centre de cette bibliothèque. Toutes les fonctionnalités de visualisation doivent être implémentés à partir d'une instance de cette classe. Différents paramètres nous permettent de régler à notre convenance la visualisation de notre graphe.

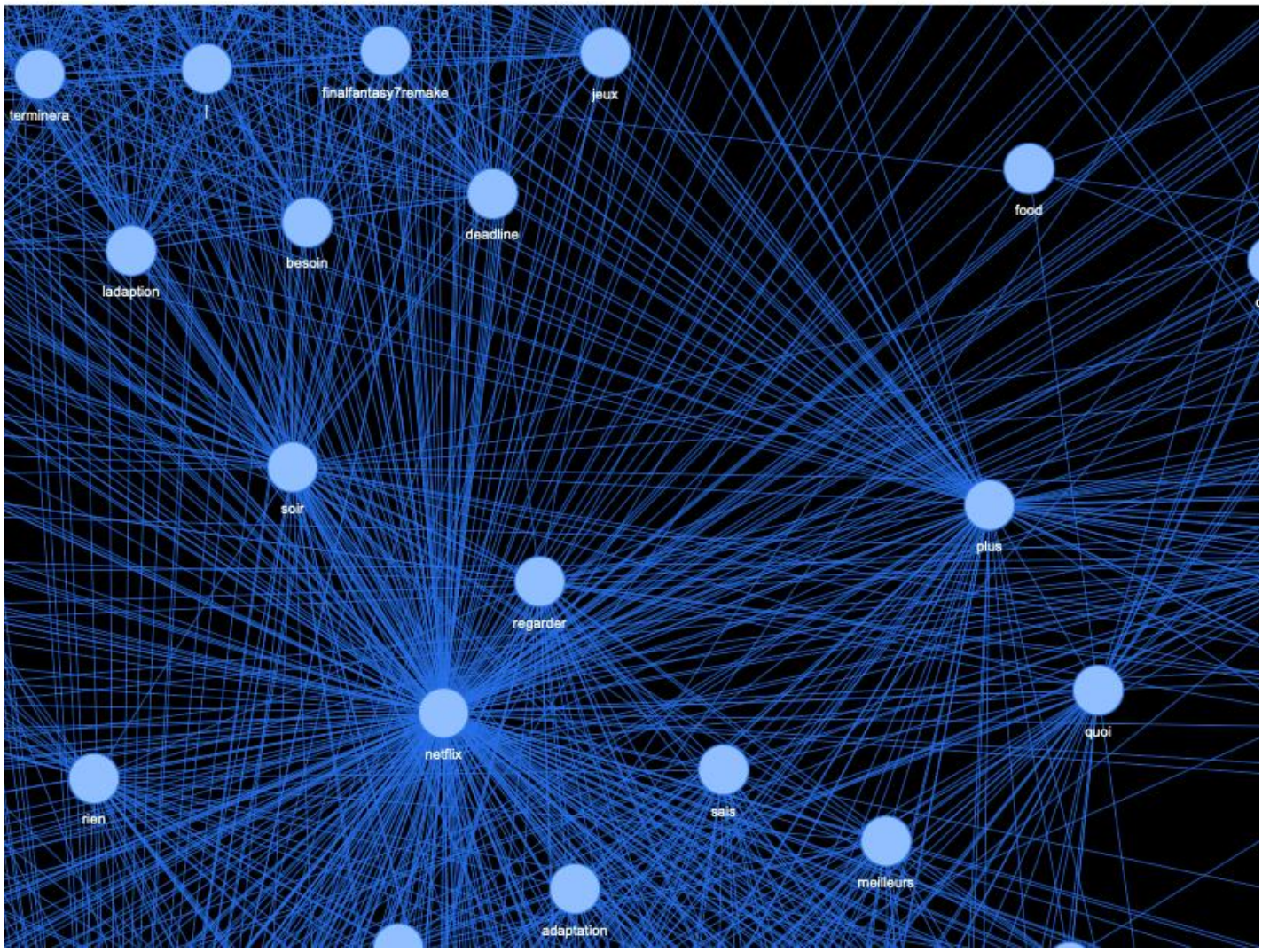
Nous pouvons ensuite avoir un accès à cette visualisation en html. Celle-ci est téléchargé dans votre répertoire de fichiers lors de l'exécution du code. Voici, ci-dessous le rendu obtenu :



La visualisation de ce graphe nous permet d'étudier l'organisation des mots dans les tweets faisant références à Netflix. Différents nœuds se sont formés.

En zoomant, nous avons accès aux mots correspondants aux différents points :

Graph de co-occurence - Tweets en rapport avec Netflix



Conclusion

Dans ce projet, nous avons utilisé les tweets récents parlant de la plateforme Netflix pour ensuite observer s'il y avait une tendance. D'après notre graphe de cooccurrence, nous voyons bien qu'il y a plusieurs groupes de mots qui se sont créés. Si nous allons plus près nous voyons que la plupart des groupes de mots sont la description d'une série ou d'un film en particulier. Par exemple, nous avons un groupe de mot parlant du film « don't look up », un film actuellement très tendance sur la plateforme.

Nous n'avons pas pu aller plus loin mais dans le même processus nous aurions pu regarder les mots caractérisant le plus une série en fonction des différents groupes de mot et du poids de chaque mot.

On aurait également aimé extraire des « colocat » de notre texte.

Répartition du travail :

Nous nous sommes réparti le travail de manière à ce que chacun cherche de son côté et lorsqu'une personne trouvait une solution, nous nous mettions à 2 pour résoudre les différentes difficultés.

Par exemple : pour la création du graphe avec la matrice, nous avons tous les 2 cherchés de notre côté comment faire. Hugo ayant trouvé une solution mais ayant du mal à la concrétisé, Louison a commencé à l'aider et nous avons trouvé la solution à 2.

Nous nous sommes donc tous les deux occupés de toutes les parties de manière assez homogène.