1 Création d'un index inversé et moteur de recherche booléen et vectoriel

Dans cette partie nous détaillerons notre approche sur les collections CACM et CS276 afin de répondre aux questions posées.

1.1 Collection CACM

1. Nous avons choisi d'utiliser la librairie NLTK pour traiter la tokenisation de la collection. Toutes les valeurs fournies sont après avoir retiré les common_words du corpus. Sur la collection CACM, nous avons donc obtenu un nombre de tokens de 92305.

2. La taille du vocabulaire obtenu avec ces tokens est de 10478 mots.

3. Le nombre de tokens sur la moitié de la collection est de 29916. La taille du vocabulaire associ est de 5780 mots. Nous pouvons donc calculer la valeur de k avec la formule suivante :
$$b = \log(\frac{\text{vocabulaire\_total}}{\text{demi\_vocabulaire}}) \div \log(\frac{\text{tokens\_total}}{\text{demi\_tokens}}) \tag{1}$$
Ce qui nous donne une valeur pour b de 0.53. Nous pouvons ensuite calculer la valeur de k avec la formule suivante:
$$k = \frac{\text{vocabulaire\_total}}{\text{tokens\_total}^b} \tag{2}$$
Ce qui nous donne une valeur de k de 25.05.

4. Maintenant que nous avons des valeurs pour k et b, nous pouvons fournir une estimation de la taille du vocabulaire pour un million de tokens avec la loi de Heaps:
$$M = k * T^b \tag{3}$$
Ce qui nous donne une estimation de l'ordre de 36865 mots de vocabulaire pour une collection ayant un million de tokens.

5. Voici les graphes de fréquence et de rang pour la collection CACM. Pour la fréquence, nous avons limit aux 200 premiers tokens afin de pouvoir mieux visualiser l'évolution:

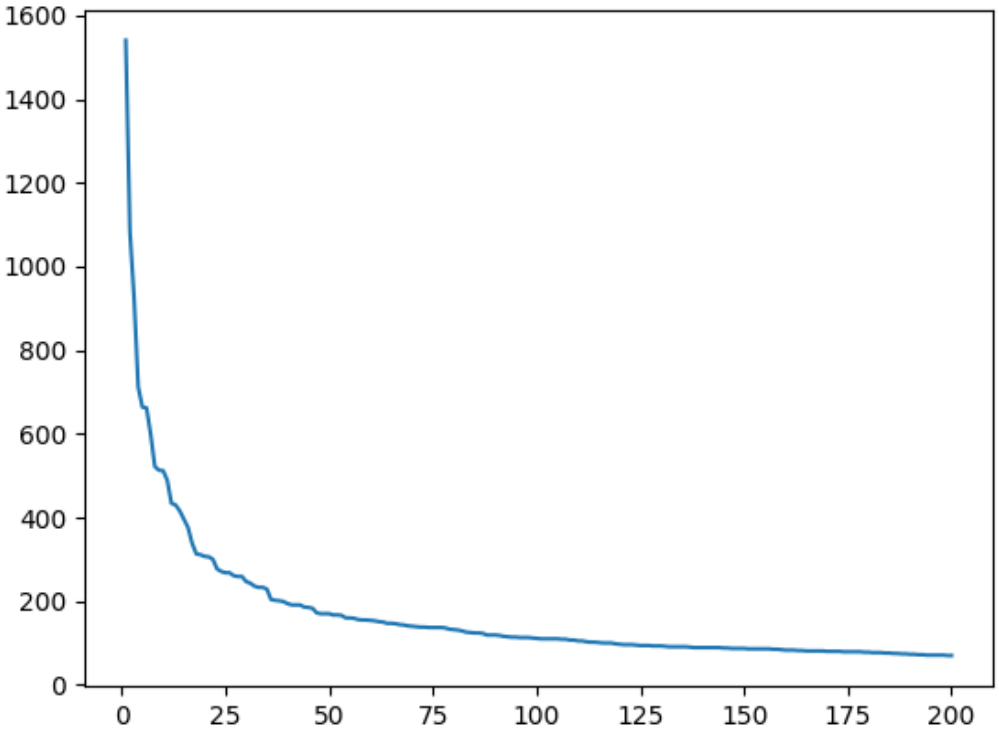

Figure 1: Fréquence par rapport au rang

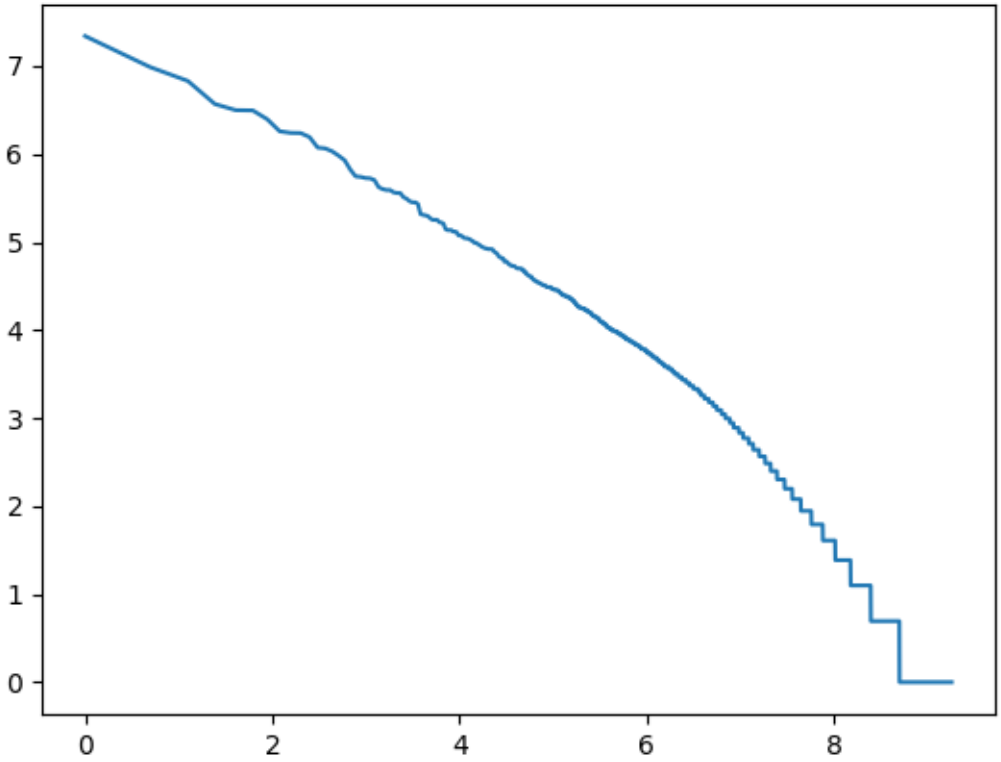

Figure 2: log(freq) par rapport au log(rank)

1.2 Collection CS276

1. Pour cette collection, aucune liste de common_words n'est fournie avec. Nous avons donc utilisé celle de NLTK pour l'anglais. Ce traitement nous donne un nombre de tokens de 18 918 184.

2. La taille du vocabulaire associé est de 335 167 mots.

3. En utilisant l'équation que nous avons utilisé pour la collection CACM, nous trouvons les valeurs de 0.71 pour b et 2.29 pour k.

4. Le nombre de tokens sur la moitié de la collection est de 8 905 232. La taille du vocabulaire associé est de 196 338 mots. La taille du vocabulaire estimé pour une collection avec un milion de tokens en suivant la loi de Heaps est de 41590 mots.

5. Voici les graphes de fréquence et de rang pour la collection CS276. Comme la collection contient beaucoup de tokens, nous avons représenté ces graphes uniquement pour les 200 premiers tokens :

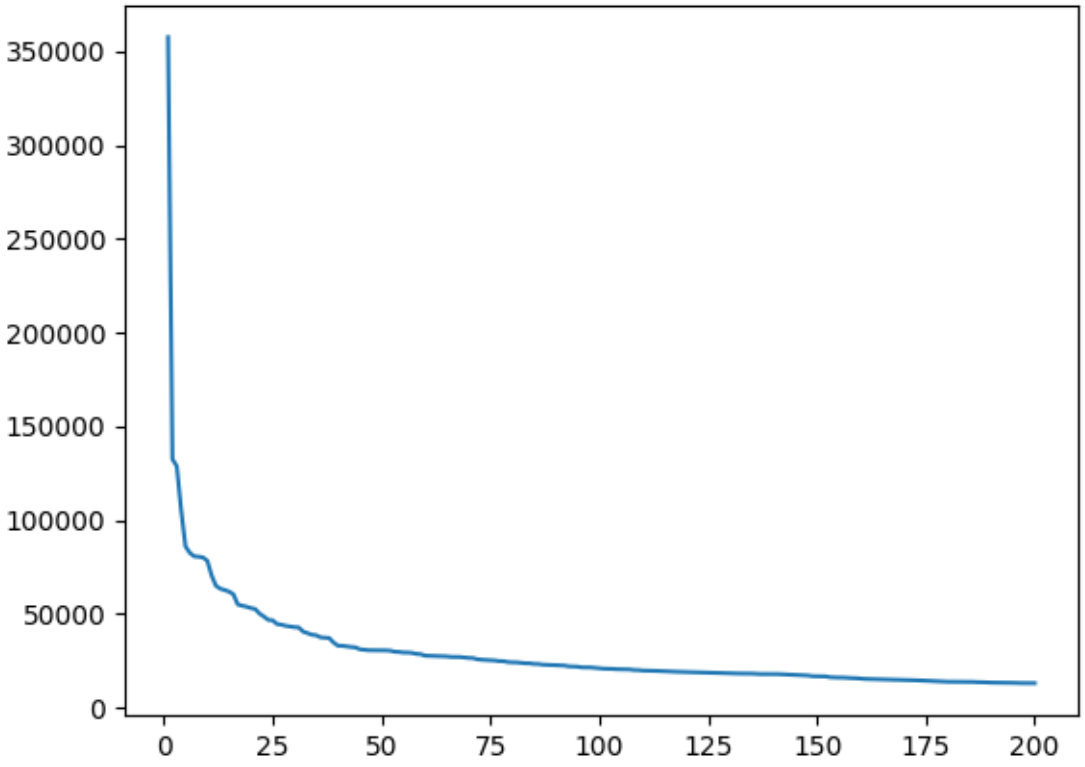

Figure 3: La fréquence par rapport au rang

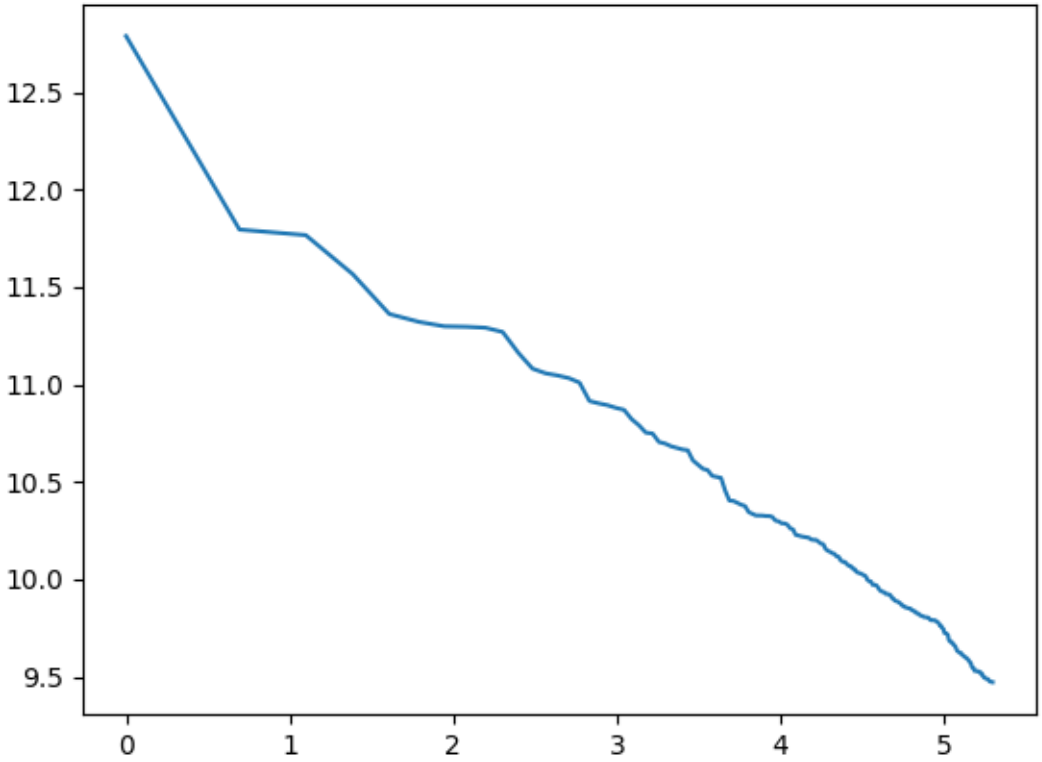

Figure 4: log(freq) par rapport au log(rank)

Indexation et évaluation pour la collection CACM

1.3 Indexes et recherches simples pour les deux collections

Pour la collection CACM, comme la collection est petite et que ce n'est pas très coûteux de génerer l'index, nous n'avons pas implémenté de fonction capable de la sauvegarder dans un fichier. Pour la collection CS276, celle-ci étant beaucoup plus volumineuse, l'index ainsi que les documents parsés sont sauvegardés dans des fichiers.

Afin de pouvoir observer les performances de chaque index, nous avons cré plusieurs fichiers pour faciliter ceci. Leur utilisation est détailée dans le fichier readme du projet github.

1.4 Mesures de performance pour la collection CACAM

1. Les mesures de performance sont accessibles en exécutant les recherches de test grace aux fichiers préparés.

2. Les résultats détailés pour l'evaluation des performances sont accessibles et facilement modifiables (modification du nombre de documents retournés en sortie du fichier prepare_cacm.py. La F-mesure de notre systeme est de 0.12. La E-mesure de notre systeme est de 0.82. La précision moyenne est de 0.3.