Esteban Thévenon

Accès et recherche d’information

Le fonctionnement et paramètre des fonctions sont indiqué dans le README.md

Ce rapport a pour but de présenter l'ensemble de mes travaux réalisés lors des 6 séances de TP de accès et recherche d’information. Au cours de ces séances, nous avons travaillé sur plusieurs sujets, tels que la loi de Zipf, l'indexation et la recherche d'informations. Il est accompagné d'un fichier README qui décrit le fonctionnement des paramètres de toutes les fonctions.

Ce rapport présente principalement les résultats car tout le code est détaillé dans le ReadME.

# TP1

## Tokenize\_cacm.py

J’ai parcouru l’ensemble des fichiers, récupéré leur contenue en minuscule. Enlevé tous les mots qui n’en étaient pas. Et écrit ses mots dans des fichiers du même nom en .tok avec un mot par ligne.A screenshot of a computer

Description automatically generated with low confidence

## Zipf.py

Le programme à deux parties différentes, une qui récupère le contenue et le traitre. Une autre qui l’inscris dans le fichier résultats.

Pour un nombre de résultats à 10 :

Text, table

Description automatically generated with medium confidence

# TP2

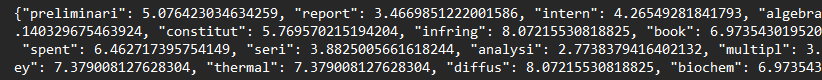
Globalement il y a une fonction par question sauf quand la question demande de revenir sur un code passé. Elles prennent toutes un lien source et un lien de destination minimum.

**Résultat de la fonction Antidico :**

A screenshot of a computer

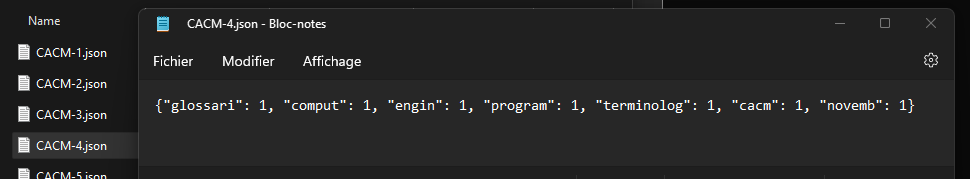
Description automatically generated with medium confidenceIl reste que les mots pertinent tronqué n’étant pas dans l’anti dictionnaire.

**Résultat de la fonction Vocabulary :**

****

Json array avec le mot en clé et son df\_i en valeur.

**Résultat de la fonction DicoWithOccurence:**

****Récupération de la tf pour chaque document sous forme d’un json.

**Résultat de la fonction Ponderation:**

**Text

Description automatically generated**

Calcul de la tf\_idf pour chaque mot de chaque document.

**Résultat de la fonction Norme:**

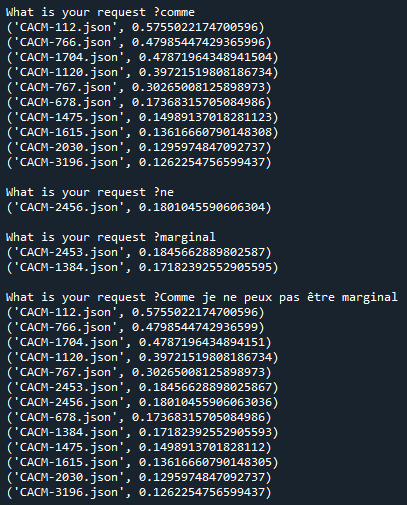
**Text

Description automatically generated**Calcul de la norme pour chaque document.

# TP3

J’ai créé deux fonctions pour charger des données. Le vocabulaire, la norme et l’index inversé.

J’ai avancé étape par étape en suivant l’énoncé.



Voici un exemple de requête pour trois mots différents puis ses mêmes mots dans une seule requête. Ceci est un exemple sans limite de taille. Pour les 5 meilleurs résultats, la réponse sera :

