

## **Projet indexation**

### **Introduction**

Le présent travail s'inscrit dans le cadre de la mise en pratique des connaissances lors du cours de Techniques informatiques & web. Il ambitionne de parfaire ces connaissances au regard des contraintes réelles de l'indexation et des moteurs de recherche.

Dans cette perspective, ce document se propose de relater la genèse de mise en place d'une application d'indexation.

### **Un moteur de recherche, c'est quoi ?**

Un moteur de recherche est un logiciel qui aide les utilisateurs à localiser des informations sur des dossiers données. Il comporte trois étapes de base qui sont l'exploration des dossiers, le nettoyage de données et l'indexation.

### **Structure d'un moteur de recherche**

L'étape d'exploration est l'endroit où le programme explore les dossiers indiqués récursivement selon une méthode récursive définie en collectant des données telles que les fichiers TXT, PDF et HTML. L'étape de nettoyage de données permet de nettoyer le flux de données qu'on a pu obtenir lors de la première étape, en appliquant des techniques comme la tokenisation, lemmatisation, suppression des STOPWORDS et le stemming. Enfin, l'étape d'indexation est l'endroit où les données collectées sont ensuite stockées dans une structure de données. Les données collectées sont ensuite classées par pertinence en ce sens que plus le classement est élevé, plus la réponse est précise.

### **Aspect fonctionnel**

Nous allons construire notre moteur de recherche avec le Framework web de python Django. Ainsi, dans ce projet, nous avons utilisé les technologies suivantes :

- Python Django
- Pandas et Numpy
- NLTK
- BeautifulSoup
- Wordcloud

## Exploration

Avant de nous lancer dans la création d'un moteur de recherche, nous avons d'abord besoin de données non structurées ou structurées en texte intégral pour effectuer une recherche. Nous allons chercher les fichiers TXT, PDF, HTML dans dossier exploré de manière récursive.

## Nettoyage de données

Nous allons appliquer une tokenisation très simple, en divisant simplement le texte en espaces blancs. Ensuite, nous allons appliquer quelques filtres sur chacun des jetons : nous allons mettre en minuscules chaque token, supprimer toute ponctuation, supprimer les Stopwords, et appliquer par la fin le stemming et la lemmatisation.

## Indexation


Nous allons stocker cela dans une structure de données. Considérez-la comme l'index à la fin d'un livre qui contient une liste alphabétique de mots et de concepts pertinents, et sur quel numéro de page un lecteur peut les trouver et le nombre d'occurrences.

## Recherche

Maintenant que nous avons tous les Tokens indexés, la recherche d'une requête devient une question d'analyse du texte de la requête. Cela rendra nos requêtes très précises, en particulier pour les longues chaînes de requête (plus notre requête contient de jetons, moins il y aura de chances qu'il y ait un document contenant tous ces tokens)

## Rendu final

Notre projet est maintenant terminé, On écrit notre requête dans la barre de recherche et on doit obtenir des résultats comme celui-ci :



**parse.html**

Dans la foulée du meilleur temps signé par Lewis Hamilton lors de la première séance, Max Verstappen a immédiatement répondu au Britannique en étant à son tour le plus rapide lors des essais libres 2 du Grand Prix d'Abou Dhabi, ce vendredi 18 novembre.....

[Voir le document](#)