Compte rendu du TP : Moteur de recherche pour Wikipédia Mathématiques

Thomas Koch

Description rapide de la base de données

Nous disposons de **2 tables** dans la base de données initiale :

- ***webpages***

URL	content
URL de la page	*contenu HTML de la page retourné par le serveur au format txt*

- ***responses***

queryURL	respURL
URL que le crawler a demandé	*URL que le crawler a reçu*

1. Première phase : crawling du site

Pour toutes les questions, nous utilisons le code suivant :

```
import sqlite3

conn = sqlite3.connect("data.db")
cursor = conn.cursor()
```

Question : ***Combien il y a de pages indexées ?***

Pour répondre à cette question, on effectue la requête suivante :

```
cursor.execute("SELECT COUNT(DISTINCT URL) FROM webpages")
print(cursor.fetchone()[0])
```

On obtient en retour **6 424 pages indexées**.

La réponse est la même avec la requête suivante :

```
cursor.execute("SELECT COUNT(DISTINCT respURL) FROM responses")
print(cursor.fetchone()[0])
```

Question : ***Combien de pages ont la même URL requêtée et répondue ?***

On répond à cette question à l'aide de la requête suivante :

```
cursor.execute("SELECT COUNT(DISTINCT queryURL) FROM responses WHERE queryURL = respURL")
print(cursor.fetchone()[0])
```

On obtient en retour **3 197 pages qui ont la même URL requêtée et répondue**.

Remarque : Nous avons au total **9 048 queryURL**. Cela s'obtient avec

```
cursor.execute("SELECT COUNT(DISTINCT queryURL) FROM responses")
print(cursor.fetchone()[0])
```

Question : ***Certaines pages comme https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Surface_de_Delaunay ne sont pas indexées, pourquoi ? Pouvez-vous en trouver d'autres ?***

En tapant la requête suivante :

```
page_requete = 'https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Surface
cursor.execute("SELECT COUNT(respURL) FROM responses WHERE queryURL = ?", (page_requete,)
print(cursor.fetchone()[0])
```

On obtient bien en retour **0**. La page en question n'est effectivement pas indexée.

Cela s'explique par le fait que certaines pages ne sont **accessibles que depuis la homepage**. Ces pages sont dites **orphelines**. C'est ce qui explique la différence entre les plus de 7 000 pages annoncées et les 6 424 pages indexées.

Pour en trouver d'autre, il faut se rendre sur la homepage https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/index puis rechercher à la main des pages non indexées.

2. Deuxième phase : précalcul de TF et IDF

Pour cette partie, j'ai modifié le fichier `shared.py` pour faciliter certains calculs.

Ainsi, j'ai appelé directement la fonction `stem()` à la fin de `extractListOfWords` pour avoir une liste de mot déjà *stemée* en sortie.

```
def extractListOfWords(content):
    soup = BeautifulSoup(content, 'html.parser')
    textTags = soup.find_all(text=True)    # find all tags with texts
    # this regex tries to find words that should not contain any of these weird chars.
    regex=re.compile(r"[^ \n'')(\]\[ {}\\]+")
    for t in textTags:
        if t.parent.name not in blacklist: # we don't want to index thoses texts
            for e in regex.findall(t):
                yield stem(e) # appel direct de stem()
```

Deuxièmement, j'ai défini les fonctions suivante pour calculer mes occurences de mot dans un dictionnaire et trier dans une liste les éléments de ce dictionnaire :

```
def wordListToFreqDict(wordlist):
    return Counter(wordlist)

def sortFreqDict(freqdict):
    aux = [(freqdict[key], key) for key in freqdict]
    aux.sort()
    aux.reverse()
    return aux
```

J'avais initialement une autre version de `wordListToFreqDict` mais elle n'était pas optimisée :

```
def wordListToFreqDict(wordlist):
    wordfreq = [wordlist.count(p) for p in wordlist]
    return dict(list(zip(wordlist,wordfreq)))
```

2.1 Sous-phase 1 : calcul de l'index inversé

Dans cette partie j'ai créé la table ***inverted_index*** suivante :

keyword	URL	frequency
mot stemé	*URL dans laquel on trouve ce mot*	*fréquence du mot dans la page*

Le détail de la création se trouve dans le fichier `index.py`.

Un même mot se retrouvant généralement dans plusieurs pages, on trouvera donc dans cette table plusieurs lignes avec le même mot mais pour des URL différentes et donc avec des frequences différentes.

A partir de cette table, j'ai également créé l'index ***inv_ind*** correspondant :

```
conn.execute("DROP INDEX IF EXISTS inv_ind")
conn.commit()
conn.execute("CREATE INDEX inv_ind ON inverted_index(keyword)")
conn.commit()
```

Question : ***Dans quelle page apparaît le plus souvent (en fréquence) le terme "matrice" ?***

En stemant le mot '*matrice*' il devient '*matric*'. Pour répondre à cette question, j'ai donc requêtée la nouvelle table ainsi :

```
for row in cursor.execute("SELECT keyword, URL, MAX(frequency) FROM inverted_index WHERE
    print(row)
```

On obtient en retour :

```
('matric', 'https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_inv
```

Ainsi, le terme matrice apparaît le plus souvent dans https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_involutive avec une **fréquence de 0.069**.

2.2 Sous-phase 2 : précalcul de l'inverse document frequency

Dans cette partie j'ai créé la table ***invert_document_frequency*** suivante :

keyword	idf
mot stemé	*idf du mot*

Le détail de la création se trouve dans le fichier `index.py`.

Comme précédemment, j'ai créé l'index ***inv_doc_freq*** correspondant :

```
conn.execute("DROP INDEX IF EXISTS inv_doc_freq")
conn.commit()
conn.execute("CREATE INDEX inv_doc_freq ON invert_document_frequency(keyword)")
conn.commit()
```

Question : ***Quel est l'IDF de "matrice" ?***

En stemant le mot '*matrice*' il devient '*matric*'. Pour répondre à cette question, j'ai donc requêtée la nouvelle table ainsi :

```
for row in cursor.execute("SELECT * FROM invert_document_frequency WHERE keyword LIKE('ma
    print(row)
```

On obtient en retour :

```
('matric', 2.0670651460787868)
```

Ainsi, **le terme matrice a un IDF de 2.067**.

3. Calcul du PageRank

Dans cette partie, j'ai simplement suivi l'algorithme donné dans l'énoncé. Le code complet se trouve dans `pagerank.py`.

A partir des scores calculés pour chaque page, j'ai créé la table ***page_rank*** :

URL	rank_score
URL	*score de la page*

Question : ***Quelles sont les 20 pages qui ont le meilleur PageRank ? Comment l'expliquez-vous ?***

Pour répondre à cette question, j'ai requêtée la table ***page_rank*** ainsi :

```
i = 1
for row in cursor.execute("SELECT * FROM page_rank ORDER BY rank_score DESC LIMIT(20)"):
    print("Rang n°", i, " : " , row)
    i += 1
```

On obtient en retour :

```
Rang n° 1  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Fonctio
Rang n° 2  :  ('https://wiki.jachiet.com/skin/jquery-ui/jquery-ui.min.css', 0.01967156224
Rang n° 3  :  ('https://wiki.jachiet.com/skin/jquery-ui/jquery-ui.theme.min.css', 0.01967
Rang n° 4  :  ('https://wiki.jachiet.com/skin/taskbar.css', 0.019664733244636343)
Rang n° 5  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/index',
Rang n° 6  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/-/s/css_m
Rang n° 7  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/-/s/css_m
Rang n° 8  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/-/s/css_m
Rang n° 9  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/-/s/css_m
Rang n° 10  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/-/s/css_
Rang n° 11  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/-/s/css_
Rang n° 12  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/-/s/css_
Rang n° 13  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/-/s/css_
Rang n° 14  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Math%C
Rang n° 15  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Math%C
Rang n° 16  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Nombre
Rang n° 17  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Nombre
Rang n° 18  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/G%C3%A
Rang n° 19  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Polyn%
Rang n° 20  :  ('https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Americ
```

On constate donc que nous obtenons beaucoup de pages `.css`. Cela est finalement normal puisqu'elles permettent d'avoir une uniformité de design entre plusieurs pages différentes.

Remarque : Il serait éventuellement intéressant d'exclure les pages `.css` lors du crawling pour ne noter que les url avec du contenu qui intéresse les utilisateurs du moteur de recherche.

4. Requêtage

Le code de cette partie se trouve dans `query.py`.

4.1 Avec simplement tf-idf

On implémente le tri en utilisant uniquement la métrique tf-idf.

Question : ***Quelles sont les dix premières pages pour "comment multiplier des matrices" ?***

Le résultat de l'algorithme donne :

```
Saisissez votre requête :comment multiplier des matrices

Résultat de votre requête avec le mode tf-idf .

Rang n° 1  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_i
Rang n° 2  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_n
Rang n° 3  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_d
Rang n° 4  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_d
Rang n° 5  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_%
Rang n° 6  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_o
Rang n° 7  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_u
Rang n° 8  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrices_
Rang n° 9  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrices_
Rang n° 10  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_
```

4.2 Avec tf-idf et PageRank

On implémente le tri en utilisant tf-idf multiplié par le PageRank de la page.

Question : ***Quelles sont les dix premières pages pour “comment multiplier des matrices” ?***

Le résultat de l’algorithme donne :

```
Saisissez votre requête :comment multiplier des matrices

Résultat de votre requête avec le mode tf-idf * pagerank .

Rang n° 1  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_(
Rang n° 2  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_i
Rang n° 3  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_o
Rang n° 4  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_d
Rang n° 5  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_d
Rang n° 6  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Groupe_g%
Rang n° 7  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_i
Rang n° 8  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_n
Rang n° 9  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Groupe_de
Rang n° 10  :  https://wiki.jachiet.com/wikipedia_fr_mathematics_nopic_2020-04/A/Matrice_
```