Projet SEO

Promotion: 2018
Majeure: MTI

Contributeurs au projet: Romain Jouffret (jouffr_a)
Valentin Barat (barat_v)

I. Objectifs du projet

Sujet 1 :

L'objectif du projet est de réaliser un programme permettant, via une chaîne de traitement, d'analyser différentes pages passées en paramètres une à une afin de détecter le présences ou l'absence de similitudes entre celles ci. Un autre paramètre correspondant à la taille des données que l'on souhaite comparer est fourni en entrée. Plus ce paramètre est bas, plus grand sera le nombre de similitudes trouvées. Il convient de bien déterminer ce nombre en fonction du résultat désiré.

Le résultat sera fourni sous forme de matrices de taux de duplication. Ces matrices représentent le taux de similitudes (ou duplications) entre les pages associées à la colonne et à la ligne correspondantes.

II. Technologies utilisées*

Python3:

Language stable et puissant, mais aussi proposant de nombreuse bibliothèques permettant de développer rapidement, Python était le choix évident pour ce projet.

Scrapy:

Un framework Python dédié au crawl et à l'extraction de données en ligne. Il est utilisé pour récupérer les pages demandées.

Numpy:

Un second Framework Python. Numpy permet de manipuler de grands ensembles de nombres. Numpy est réputé pour sa stabilité et sa performance.

Pandas:

Également un framework pour Python, Pandas permet de manipuler de grands volumes de données structurées très simplement. Pour ce faire, Pandas est basé sur le framework précédent: Numpy. Pandas nous permet de trouver les similitudes entre les pages.

Plotly:

Plotly est une bibliothèque graphique pour Python. Elle permet de dessiner de manière simple et rapide des graphes à l'aide de données passées en entrée. Bien sur, Plotly nous permet d'afficher les résultats obtenus.

** D'autre outils mineurs sont aussi présents mais pas cités dans le document volontairement par soucis de clarté. On fera notamment attention aussi pour l'installation de xhtml2pdf qui se doit d'être faites en beta car sinon il n'y a pas de support de Python 3.*

III. Explications et Algorithmes

Shingles:

Un shingles représente la taille de données que nous allons utiliser pour calculer les similitudes entre deux pages. Plus cette taille est faible, plus grand sera le nombre de similitudes trouvées. Cependant, il faut adapter cette valeur en fonction du type de données à analyser ainsi que du type résultat désiré.

En effet, dans notre cas, si l'on met une taille de 2 caractères, le nombre de pages contenant 2 caractères identiques sera très élevé, mais pas forcément pertinent. Au contraire, un nombre trop élevé ne révélerai pas forcément deux pages proches l'une de l'autre.

Indice et distance de Jaccard:

L'indice et la distance de Jaccard représentent respectivement les similitudes et les différences entre deux éléments composés de données binaires.

Nous utilisons cet algorithme afin de pouvoir déterminer, à l'aide d'un pourcentage, le taux de similitudes (ou duplications) entre deux pages. Le résultat sera donc rentré dans la matrice de comparaison finale.

Matrice de comparaison:

Nos résultats sont exprimés à l'aide de matrices de comparaison. Les matrices de comparaisons ont pour but de pouvoir analyser des résultats de comparaison d'ensemble un à un de manière simple et rapide.

Une matrice de comparaison affiche le résultat de la comparaison en l'élément 1 et l'élément 2 dans la case $M_{1,2}$. Un simple calcul linéaire nous permet d'obtenir (dans notre cas), les pages les plus proches des autres de manière globale.

Fonctionnement du programme:

Ligne de commande : $ python3 main.py *thewebsite.here.com #ofShingles*

Dans un premier temps l'utilisateur passe en argument le site qu'il veut analyser puis la taille des shingles qu'il veut créer pour analyser les items que le logiciel aura extrait. Le script python lance alors une série de tests avant de lancer une araignée créée à l'aide de scrapy pour parcourir le site et en extraire le contenu de sa/ses page(s).

Ensuite on peut voir que le script parcourt les items stockés dans un fichiers texte en dur pour une utilisation extérieure si besoin. Puis les différents documents sont générés à l'aide de différents outils et puis ouvert automatiquement pour l'utilisateur.

Utilisation :

- Oublie des arguments :

```
$ python3 main.py
usage: main.py [-h] root shingles
main.py: error: the following arguments are required: root, shingles
```

- Help :

```
$ python3 main.py -h
usage: main.py [-h] root shingles

positional arguments:
root        Website you want to scrap
shingles    Integer that will determine shingles size

optional arguments:
-h, --help  show this help message and exit
```

IV. Résultats et observations

Scrapping des pages:

On peut voir que lorsque la configuration est faites pour un site en particulier on obtient des résultats beaucoup plus cohérents avec ceux fait sur du texte HTML pur. Le temps de scrapping depends aussi beaucoup des sites mais cela n'est en rien corrélé à nos calculs.

Shingles et splitting:

Selon la taille des k-gram on constate des variations importantes des taux de similitudes. Il faudrai plus d'études sur les objets ("items") pour déterminer en fonction des différents textes la taille la plus appropriée pour le découpage. Pour notre projet conformément aux directives données la taille des shingles est définie par l'utilisateur.

Distance de Jaccard:

En prenant la bonne taille de k-gram on peut observer les différents résultats qui se démarque après analyse des objets collectés. Différents cas rencontrés et la comparaison des résultats avec les objets récupérés, cela représente les résultats obtenus avec le script réalisé et peuvent diverger selon les pages webs récupérées et la justesse de notre implémentation :

- 0-5 %: Cela peut correspondre à certains mots de liaison et balises que l'objet contient mais rien de significatif. On peut même dire que très peu de mots voir aucun sont en commun entre les deux textes.

- 5-19%: Pour la majorité des textes qui sont sur le même thème mais pas liés, on constatera ce résultats

- 20-49%: Ces résultats correspondent à du plagiat, non pas dans le but de copier une oeuvre mais des textes similaires dans le but et la façon d'être écrit.

- 50-79%: Le seul constat fait pour des résultats ayant ce pourcentage sont des textes très similaires juste changeant des informations sur des détails et des noms.

- 80-100%: Résultats identiques provenant des même pages ou de page "miroirs", les diverses balises pouvant changer les pourcentage évitant ainsi le contenu dupliqué ou *duplicate conten*t.

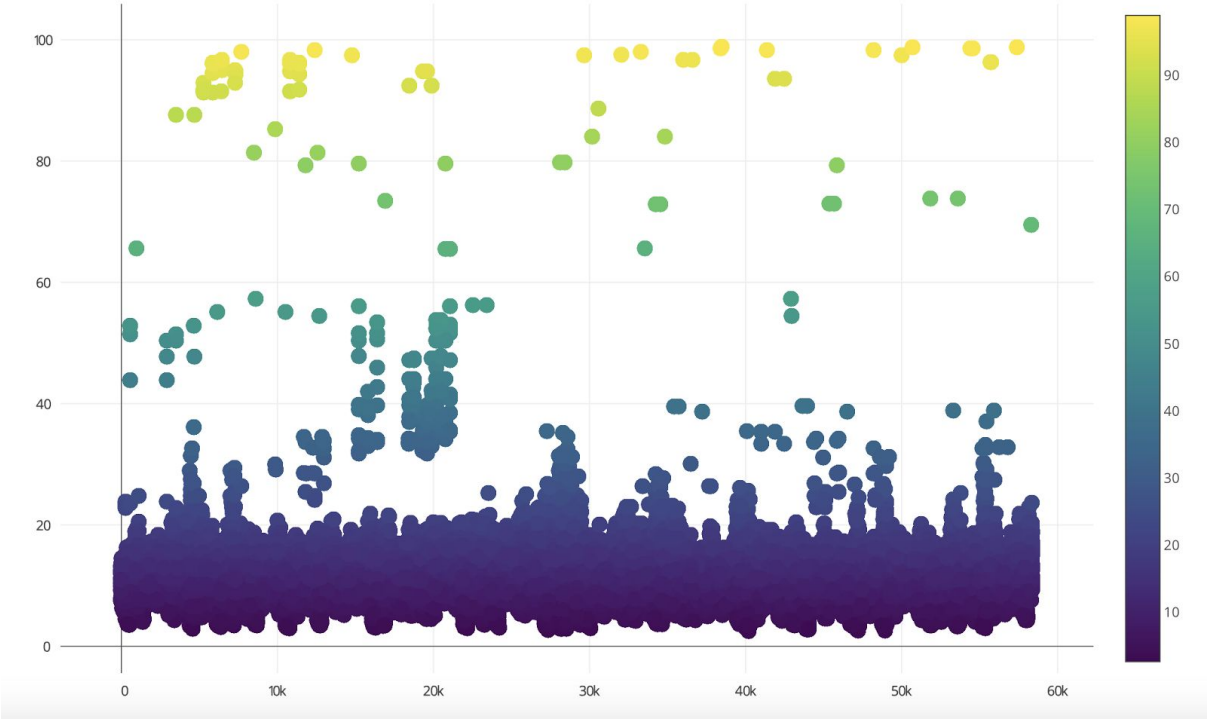

Figure 1: Résultats des distance de Jaccard sur le site craigslist.org/search/egr colorises en fonction des pourcentages (~55,000 comparaison avec possibilité de doublons)

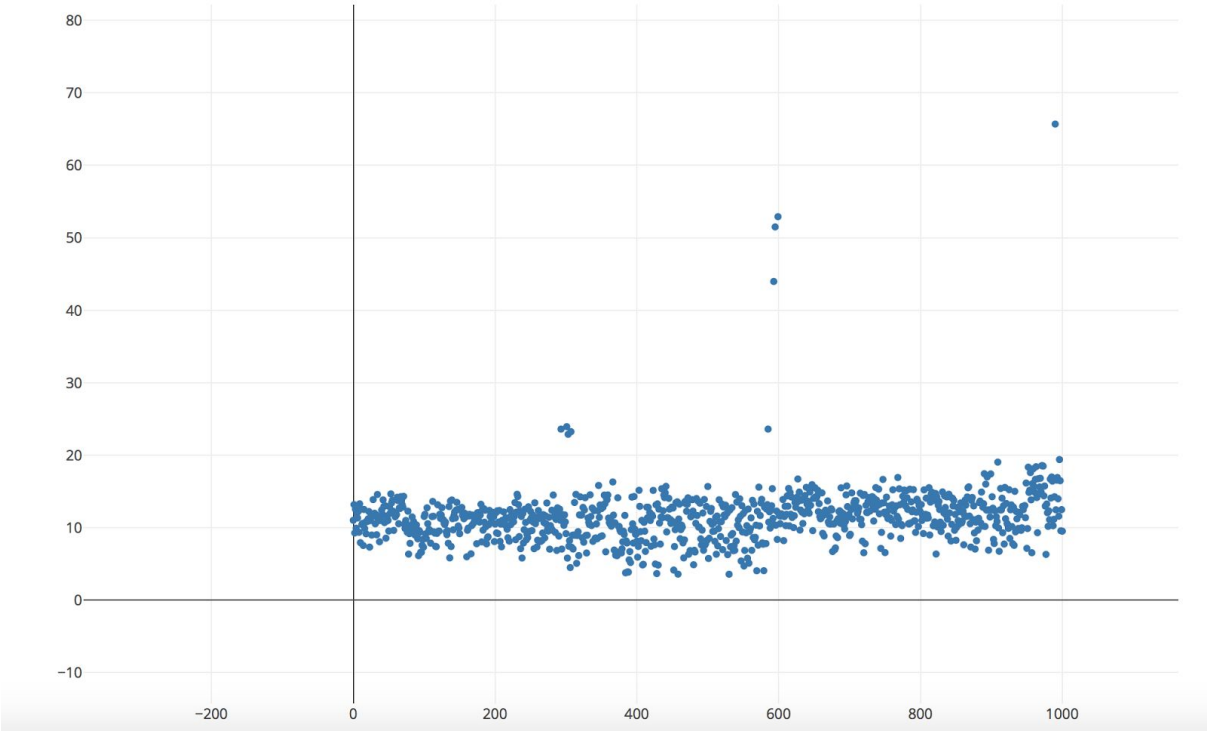

Figure 2: Résultats des distance de Jaccard sur le site craigslist.org/search/egr en poucentage limités aux 1000 premières comparaisons