Projet Wikipedia Matrix — rapport d'avancement

Jean-Yves Degos

20 décembre 2020

La classe `BenchTest.java` génère dans le répertoire `output/html/` un fichier `csv` en plus de ceux demandés. Ce fichier, `BenchTestStats.csv`, est alimenté, chaque fois qu'un fichier `csv` est écrit, par, pour chaque tableau :

- le numéro de l'URL + 1 ;
- le numéro du tableau dans la page ;
- le nombre de rangées ;
- le nombre de colonnes ;
- le nombre d'en-têtes (*headers*) ;
- le nombre de cellules.

Dans la mesure où on ne retient pour le moment que les tableaux (balise `table`) de classe `wikitable sortable`, qui dont les balises `tr` (resp. les balises `td`) n'ont pas d'attribut `rowspan` (resp. pas d'attribut `colspan`), on s'attend à ce que pour chaque entrée, le nombre de cellules soit égal au nombre de rangées, multiplié par le nombre de colonnes. C'est n'est pas toujours le cas...

Nous avons effectué quelques statistiques avec `R` sur le fichier `BenchTestStats.csv`), qu'on retrouve dans le fichier `BenchTestStats.R` ajouté au dépôt.

1 Nombre de tableaux extraits

Cela correspond au nombre d'observations du fichier `BenchTestStats.csv`, soit 690.

2 Nombre de colonnes présentes

On obtient la distribution suivante :

Min.	Median	Mean	Max.	Std
0.000	2.000	4.316	35.000	5.546887

3 Nombre de lignes présentes

On obtient la distribution suivante :

Min.	Median	Mean	Max.	Std
2.00	22.00	28.88	464.00	35.75158

4 Nombre de cellules présentes

On obtient la distribution suivante :

Min.	Median	Mean	Max.	Std
4.0	55.0	151.9	5568.0	365.3429

5 Noms de colonnes les plus fréquentes

Nous n'avons pas encore fait le développement nécessaire qui permettrait d'obtenir cette réponse.

6 Qualité et faiblesse de l'extraction

L'extracteur prend en charge uniquement des tableaux "normaux", c'est-à-dire sans colonnes ni rangées fusionnées. Il faudrait donc améliorer le code actuel pour que ces tableaux puissent être pris en compte. Définissons, pour chaque tableau extrait, une variable `tabTest` égale à la différence du produit du nombre de rangées multiplié par le nombre de colonnes, et du nombre de cellules. On obtient alors les résultats suivants :

	tabTest==0	tabTest > 0	tabTest < 0
Occurrences	40,14493 %	9,275362 %	50,57971 %

Cela montre que l'extraction fonctionne fidèlement dans seulement 40 % des cas environ, ce qui demanderait à être amélioré.

7 Synthèse générale

Je trouve ce projet à la fois intéressant et utile. Toutefois, j'ai démarré avec un certain handicap en Java, et des contraintes personnelles le weekend des 4-5-6 décembre m'ont empêché d'y consacrer tout le temps que j'aurai voulu.

J'en ai retiré quelques savoir-faire intéressants : une meilleure connaissance de Java, l'utilisation de `jsoup`, la lecture/écriture dans des fichiers csv, mise en œuvre des tests unitaires, etc.

Au moment où je fins d'écrire ces lignes (dimanche 20 décembre, 17h30), le code est encore assez embryonnaire, et il reste pas mal de choses à revoir au niveau de la conception. Par

exemple, on pourrait utiliser un patron *Strategy* pour mettre en œuvre un comportement différent de l'extracteur selon le type de tableau rencontré en document HTML, prendre en compte les tableaux à colonnes ou rangées fusionnées, par exemple. Je pense que je continuerai à travailler sur ce projet après le rendu, car ça m'intéressait d'obtenir une application efficace.

8 Dépôt github

`https://github.com/JYDegos/wikipediamatrix-bench`