Rapport Génie Logiciel Scrum

GUILMAIN Loïc GAUTIER Bastien
DULCAMARA Thomas CIANAMEA Mickaël
LAFAGE Benjamin

January 6, 2020

Abstract

Dans cet article, nous allons vous présenter notre parseur d'articles scientifiques en format texte ou au format XML ainsi que nos résultats.

1 Méthode

Nous convertissons un article scientifique normalisé au format PDF en fichier texte ou XML en fonction de l'option choisit par l'utilisateur.

1.1 Pré-requis

Pour utiliser notre parseur, il est nécessaire d'avoir Python dans sa version 2.7 ainsi que le convertisseur *pdftotext*. Attention, les noms des fichiers PDF à convertir doivent être impérativement normalisés comme suit:

- *auteur_titrepdf_année.pdf*

Il faut bien veiller à ce que le titrepdf ne comporte pas d'espace et ceux-ci doivent être remplacé par des "_".

1.2 Utilisation

Il faudra glisser les fichiers PDF dans un dossier *Papers* situé au même endroit que les fichiers Python permettant le bon fonctionnement du parseur. A ce même endroit ouvrez une console commande puis lancer le programme avec la commande suivante:

- *python main.py*

Celle-ci va lancer le programme et stocker les fichiers résultats dans un dossier nommé *finalDossier*. Par défaut, les fichiers résultant de la conversion sont au format texte.

1.3 Les options

Notre parseur possède deux options, une pour la format texte et une autre pour le format XML. Ce choix se fait lors de l'appel du programme, c'est-à-dire, pour obtenir des fichiers XML il faudra rentrer la commande suivante:

- *python main.py -x*

Et pour les fichiers txt:

- *python main.py -t*

2 Résultats

Nous avons mesuré la précision de notre parseur avec le nouveau corpus de test de 10 nouveaux articles PDF.

Nom	Frontières véritables	trouvées	correctes	incorrectes	non détectées
Boudin	7	7	7	0	0
Gonzalez	8	8	8	0	0
Iria	7	7	7	0	0
Martin	7	7	7	0	0
Mikheev	7	7	7	0	0
Mikolov	7	7	7	0	0
Nasr	7	7	7	0	0
Torres98	7	7	7	0	0
Torres2018	8	8	7	1	0

Le PDF nommé Jing possédait une protection empêchant *pdftotext* de le convertir. Nous avons donc décider de le retirer des résultats.

$Precision = \frac{correctes}{trouvees} = \frac{64}{65} = 0,98$

3 Conclusion

Nous avons réparti l'ensemble des fonctions entre tous les membres du groupe.

- Loïc s'est occupé de mettre en commun toutes les fonctions, il a crée la fonction *converter.py*, il s'est occupé de la gestion du XML et du *main.py*
- Bastien a crée la fonction *abstract.py*, la fonction *getBiblio.py* et la fonction *getConclusion.py*
- Thomas a crée la fonction *getTitre.py*, une fonction pour écrire dans les fichiers finaux et de la fonction *getIntroduction.py*
- Mickaël a aidé sur la fonction *getTitre.py*, sur la gestion du XML, et sur la fonction *getIntroduction.py*
- Benjamin a aidé sur la fonction *converter.py*, a crée la fonction *menu.py*, a rédigé les rapports de chaque sprint ainsi que le rapport final.

Il est possible de retrouver l'ensemble des versions du parseur ainsi que tous nos rapports sur le répertoire Github suivant:

- `https://github.com/SkyNeolagos/Genie_Logiciel_Scrum`

Nous avons choisi d'utiliser *pdftotext* car c'est le convertisseur qui gère le mieux les PDF contenant deux colonnes.
Pour le langage de programmation, nous avons décidé de choisir python car nous avons jugé que celui-ci était le plus simple pour réaliser ce parseur et c'est celui où nous avions des connaissances.
Pour conclure globalement sur notre parseur, ce dernier est très fiable puisqu'il a obtenu une précision de 98%.