AVIGNON UNIVERSITÉ

Rapport final

Groupe 4
Ahmed Ait Oufkir
Abderrahim Bouhriz
Lamyae Khairoun
Oussama Sbaa
Mourad Walid

7 janvier 2021

L3 Informatique
Ingénierie Logicielle
UE Genie Logiciel
UCE Méthode Scrum

Responsable
Nejat Arinik

UFR
SCIENCES
TECHNOLOGIES
SANTÉ

AVIGNON UNIVERSITÉ
CERI

CENTRE
D'**ENSEIGNEMENT**
ET DE **RECHERCHE**
EN **INFORMATIQUE**
ceri.univ-avignon.fr

Sommaire

1 Authors

Walid MOURAD (walid.mourad@alumni.univ-avignon.fr)

Oussama SBAA (oussama.sbaa@alumni.univ-avignon.fr)

Lamiae KHAIROUN (lamiae.khairoun@alumni.univ-avignon.fr)

Abderrahim BOUHRIZ (abderrahim.bouhriz@alumni.univ-avignon.fr)

Ahmed AIT OUFQIR (ahmed.ait-oufqir@alumni.univ-avignon.fr)

2 Abstract

Dans le cadre de l'Unité d'Enseignement "génie Logiciel" en CERI (Centre d'enseignement et de Recherche en Informatique), nous avons réalisé un programme qui était un Parseur d'articles scientifiques. Ce parseur consiste à convertir des fichier `.pdf` en des fichiers `.txt` , puis il les parse et donne comme sortie un fichier `.txt` ou bien `.xml` (selon le choix de l'utilisateur) ce dernier contient les informations principales de chaque articles (le titre, l'abstract, les auteurs, l'introduction, le corps de l'article, la discussion, la conclusion et les références bibliographiques).Pour ce faire nous avons utilisé dans un premier temps le système d'ORC **pdftotext**, car il est le plus fiable en comparaison avec l'autre système d'ORC **pdf2txt**. En effet nous faisions dans un premier temps l'implémentions du code du parseur puis nous passions à l'analyser en effectuant plusieurs tests.

3 Méthode

Le programme réalisé est un parseur d'articles scientifiques.Pour réaliser un parsing l'utilisateur doit donner dans un premier temps un nom d'un répertoire, puis on lui demande de préciser quelles sont les fichiers qui veut parser. Ensuite le programme prend le nom de ce répertoire, après il accède à ses fichiers `.pdf`, il sélectionne les fichiers déjà choisis par l'utilisateur, il les convertit ensuite en des fichiers `.txt`, et comme résultat il nous fournit aussi soit un fichier `.txt` (si l'utilisateur a choisi l'option **`-t`**), soit un fichier `.xml` (si l'utilisateur a choisi l'option **`-x`**), ce fichier résultat contient les informations principaux pour chaque fichier PDF, il les écrit chacune dans une ligne et il les affiche comme suit :

- Le nom du fichier d'origine (dans une ligne)
- Le titre du papier (dans une ligne)
- Le résumé (abstract) de l'auteur (dans une ligne) - L'introduction.
- Le corps (Le développement du papier).
- La conclusion (La conclusion du papier)
- La discussion (La discussion du papier).
- La section auteurs et leur adresse (auteur).
- Les références bibliographiques du papier (biblio).

Le programme alors stocke les fichiers parser a partir des `NOM_FICHIER.pdf` dans un dossier(parsed_files) qui se vide a chaque nouvelle utilisation du programme pour laisser place aux nouveaux fichiers `NOM_FICHIER[.txt/.xml]`. Les fichiers a parser seront choisie grâce a un menu qui s'affiche au début du programme,ce menu affiche les fichiers `.pdf` numérotés existant dans le dossier fournie, l'utilisateur alors choisie une liste de fichiers a parser comme par exemple "1,5,6,11" sinon tape "0" pour tout traiter en une fois, donc le programme contient une fonction qui récupère la liste et convertie les fichiers, puis ces fichiers texte serons ouvert et leurs contenue seras stocker dans une variable **data**, elle seras récupérer par différente fonction chacune parse les sections demander puis les écrit dans le `NOM_FICHIER[.txt/.xml]`.

4 Résultats

Précision = Sections correctes trouvées par le système / Sections trouvées parle système

Fichier PDF	Résultats souples	Résultats corrects
Esmailian2014.pdf	2/3	1/3
Estrada2014.pdf	2/3	2/3
Fazekas2016.pdf	4/7	2/7
Franti2018.pdf	7/8	6/8
Labatut2015.pdf	7/7	4/7
Leskovec2008.pdf	7/7	7/7
Newman2019.pdf	3/3	1/3
Veldt2017.pdf	8/8	7/8
Wu2009.pdf	5/6	2/6
Zhu2018.pdf	5/6	5/6
Précision	**0,86**	**0,63**

Table 1. Résultats de calcule des précisions

5 Conclusion

Malgré les difficultés que nous avons rencontré (contrainte du temps / peu des sources ...), nous avons pu crée un parseur qui convertit les pdf au fichier txt ou XML selon le choix de l'utilisateur avec un pourcentage de performance de **86%** de réussite pour la précision souple, et avec un pourcentage de **63%** de réussite pour la précision stricte sur un corpus de 10 `pdf` mis à notre disposition pour évaluation, ce qui est un bon résultat. Réaliser un parseur avec des expressions régulière (Regex) rend le logiciel plus pratique et performant, c'est la méthode la plus précis qu'on a trouvé,l'essentiel c'est de fournir un logiciel qui marche bien pour faciliter la lecture des articles scientifiques, ce parseur peut s'améliorer après pour qu'il puisse parser plus de pdf et pour que le pourcentage de réussite s'augmente en ajoutant d'autres contraintes aux expressions régulières du logiciel.