AVIGNON UNIVERSITÉ

Parseur d'articles scientifiques en format texte ou xml

TD4-ALT

Mohamed Amine SAADOUN
Ismail FAKIR
Wafae EL MANSOURI
Maryem BOUZIANE

1er janvier 2023

L3 Informatique
Ingénierie logiciel
UE Genie logiciel

Responsables
Juan-Manuel TORRES
MORENO JIMENEZ Luis Gil

UFR
SCIENCES
TECHNOLOGIES
SANTÉ

AVIGNON UNIVERSITÉ
CERI

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

Sommaire

1 Abstract

L'objectif de ce projet est de développer un parseur d'articles scientifiques qui va nous permettre de convertir des fichiers PDF en fichiers txt et xml en identifiant les différentes sections d'un corpus : Nom du fichier, Titre du corpus, Abstract / Résumé du corpus, Introduction du corpus, Corps du texte, Discussions du corpus, Conclusion du corpus et Bibliographie / Références.

Le Python est le langage choisi pour la réalisation de ce projet, et le résultat sera sous forme d'un répertoire (Resultat) contenant l'ensemble des fichiers (Texte ou xml) .

2 Méthode

Pour répondre aux besoins de l'équipe IA, on a suivi les principes de la méthodologie agile Scrum, ainsi on a fusionné entre les fonctionnalités de python et le Shell.

Premièrement, on a créé une fonction trouver_ext qui prend deux arguments(l'extension des fichiers recherchés et le nom de dossier)qui a comme but de chercher tous les fichiers se trouvant dans un dossier .Puis on lance notre script Shell qui nous permet de transformer tous les fichiers PDF se trouvant dans le répertoire en fichiers txt qui seront quant à eux déposés dans le dossier "dossierText" .Durant cette deuxième partie nous avons utilisé pdftotext car il se caractérise par l'existence de plusieurs options qui nous ont facilité le travail.

Deuxièmement, on a développé plusieurs méthodes pour obtenir les informations recherchées, par exemple, pour chercher l'introduction, l'abstract et la conclusion, on a créé une fonction trouverParag qui a comme argument le chemin de fichier et le titre de la section, et qui retourne le paragraphe recherché ou not found si elle n'a pas trouvé le titre passé par argument.

Pour différencier les trois types de paragraphe, on a posé plusieurs conditions d'arrêt, pour le cas d'introduction, l'extraction s'arrête quand on trouve le chiffre 2 et le dernier caractère sur la ligne précédente est un point.

Pour le cas d'une conclusion, l'exécution de la fonction s'arrête quand on trouve le mot REFERENCES ou references, et pour le cas d'abstract, on s'arrête quand on trouve 1 introduction ou I. INTRODUCTION.

Pour trouver les autres informations, on a développé d'autres méthodes, pour l'extraction de la discussion, on a une fonction trouverDisc, qui ressemble à la fonction trouverParag, en utilisant le titre passé par argument, on cherche une section ayant pour titre Discussion, et on s'arrête une fois le mot References, Conclusion est trouvé.

Pour trouver les références, la fonction trouverRef, fonctionne comme la méthode précédente, elle commence quand elle trouve le mot Reference et elle s'arrête à la fin du fichier.

Pour trouver le corps de fichier, on a créé une méthode nommée trouverCorps qui prend comme argument le chemin de fichier, tout d'abord on vérifie si l'introduction existe, si oui, on commence par le chiffre 2 , et on s'arrête une fois le mot References , Conclusion, Discussion est trouvé ,si non ,on commence par le chiffre 1 ,en vérifiant toujours que la ligne précédente est fini par un point.

Pour trouver le titre et les auteurs, on a procédé comme suit : On récupère les deux premières lignes d'un fichier pour le titre, puis on continue l'extraction jusqu'à avoir trouver le

mot abstract, et c'est la section des auteurs.

Finalement, on a la méthode main consacré à l'exécution de programme, dans laquelle, on prend le deuxième argument qui correspond au choix de format -t(parser_file_to_txt) pour texte et -x (parser_file_to_xml) pour xml ,et le troisième argument qui est le chemin de répertoire d'exécution ,puis on appelle le script Shell .

3 Resultat

Pour calculer la précision, l'équipe s'est divisée entre ceux qui ont établi le corpus de référence des frontières de toutes les sections et ceux qui se sont occupés de la demande du département de contrôle de qualité, à savoir l'évaluation de la qualité du système en calculant la précision sur un corpus de 10 fichiers PDF. La précision a été calculée en comptant le nombre de sections correctement trouvées par le système, divisé par le nombre total de sections trouvées. (Précision = Sections correctes trouvées par le système / Sections trouvées par le système). Les résultats du calcul de précision pour chaque PDF du corpus de test sont présentés dans le tableau ci-dessous.

3.1 Numérotation des fichiers

Numero	Fichiers
1	A Benders Decomposition Approach toCorrelation Clustering.pdf
2	A memetic algorithm for community detection in signed networks.pdf
3	An Improved Branch and Cut Code for the Maximum Balanced Subgra.pdf
4	Cabrera RESUMES 2019.pdf
5	Conversational Networks for Automatic Online Moderation.pdf
6	Dynamical Models Explaining Social Balance and Evolution of Cooperation.pdf
7	Exact Clustering via Integer Programming and Maximum Satisfiability.pdf
8	LDA resume.pdf
9	Partitioning large signed two mode networks Problems and prospects.pdf
10	Polibits 42 02.pdf

Table 1. Tableau des fichiers.

3.2 calcul de precision

Fichiers	1	2	3	4	5	6	7	8	9	10
Frontières véritables	7	7	7	7	7	7	7	7	7	7
Frontières trouvées	7	7	7	7	7	7	7	7	7	7
Frontières correctes	7	6	6	4	3	6	5	6	6	6
Frontières incorrectes	0	1	1	3	4	1	2	1	1	1
Frontières non détectées	0	0	0	0	0	0	0	0	0	0

Table 2. Calcul de precision.

La précision moyenne de tous les fichiers PDF est calculée en divisant la somme des précisions individuelles (1+0.86+0.86+0.57+0.43+0.86+0.71+0.86+0.86+0.86) par 10. La précision moyenne est de 0.787.

4 Conclusion

En guise de conclusion, l'adaptation de la méthodologie Scrum, nous a apporté la flexibilité nécessaire pour la réalisation de parseur et nous a permis de nous adapter en fonction du besoin de l'équipe IA. Cela nous a également permis de renforcer l'esprit d'équipe, car nous travaillons toujours à plusieurs sur une tâche.

Restons agile dans un monde qui bouge !