Université Bretagne Sud - Vannes

Projet et conduite de projet

Analyseur d'articles scientifiques

CHAUVEL Lauren
KAHASHA Guy-Alexandre
RANDRIAMIHAJA Lalatiana

Professeur : Rémy KESSLER
Scrum Master : KAHASHA Guy-Alexandre

Janvier 2024 — Mai 2024

1 Introduction

Ce rapport a été rédigé pour rendre compte du parser réalisé pour le cours de "Projet et conduite de projet". Il présente les différentes méthodes d'extraction des informations et les résultats obtenus sur le corpus de document "corpus TEST" fourni sur la page Moodle du cours.

2 Choix de la bibliothèque

Nous avons tout d'abord consacré la première semaine du projet à évaluer les différentes bibliothèques (pymupdf , pdfplumber,pdftotext,pdf2txt) sous différents langages (Python,C,Java). Nous avons donc fini par opter pour la librairie pymupdf en python car elle disposait de plusieurs avantages tels que :

- Obtention de la taille des caractères de chaque ligne
- Obtention de la police de caractère de chaque ligne
- Obtention du bloc ainsi que des dimensions de chaque ligne
- Obtention du texte ligne par ligne

NB : Avec pymupdf le document est divisé en plusieurs rectangles géométriques appelés 'bbox' selon la concentration de texte qui se trouve dans le document et est donc subdivisé en plusieurs bloc de texte qui sont contenus par une bbox.

3 Première approche

Nous avons tout d'abord tenté une approche assez naïve et simpliste qui consistait à faire une simple reconnaissance de mots clés , basée sur le résultat obtenu après l'extraction du titre là où nous contentions de prendre la ligne de plus grande police de caractère. Ensuite nous cherchions les mots clés de même taille et de même police. Nous nous sommes heurtés par la suite à divers problèmes à la vue des premiers résultats du comparateur.

Boudin-Torres	69.56%
Das_Martins	61.97%
Gonzalez	60.97%
Iria	51.02%
Jing	2.17%
mikolov	67.02%
Torres-moreno	NA
Torres	47.72%
kessler94715	61.03%
kessler Metics	67.89%
mikheev	67.89%
Total	50.66%

Nous avons donc pu relever les erreurs de notre première approche non concluante

- Le titre a une taille de police et une police souvent particulières donc difficulté dans la détection des autres mots clés
- Manque de précision suite à une délimitation faites à la rencontre d'un mot , qui pouvait ne pas être celui souhaité
- Difficulté dans l'extraction des adresses mails, des noms d'auteurs et des affiliations pour lesquels nous délimitions une zone de recherche (entre l'abstract et le titre) , donc si celle-ci est mal délimitée la recherche ne donne pas de résultat satisfaisant, l'absence d'adresse mail nous empêchait d'avoir les noms d'auteur (qui de plus n'est pas une source des plus fiables) et les affiliations qui étaient extraites mais non délimitées, donc impossible de les faire correspondre avec le bon auteur.
- Un temps d'extraction trop long pour des performances médiocres

Nous nous sommes alors repris et avons revu l'entièreté de l'implémentation du code.

4 Remodélisation du parser

A. Extraction des pages

Jusque là nous traitions le fichier à la volée par plusieurs parcours successifs pour chacune des données à extraire, alors dans des soucis d'optimisation nous avons cette fois-ci stocké l'ensemble des pages dans une liste puis nous récupérons les données jugées utiles et les regroupons toutes dans un objet de type dictionnaire de liste auxquelles nous avons accès via des clés précises :

— TEXT : Les lignes de textes sont stockées
— FONT : La police de chaque ligne de texte
— SIZE : Les tailles de police de chaque ligne
— NUMPAGE : Les numéros de page de chaque ligne
— BBOX : La bbox de de chaque bloc de texte pour chacune des lignes
— bbox : La bbox de chaque ligne de texte
— PROP-FONT : La proportion d'apparition de chaque police de caractère dans le fichier
— PROP-SIZE : La proportion d'apparition de chaque taille de police dans le fichier

NB :
— Toutes les listes sont de même taille
— Les éléments de même indices dans les différentes listes sont associés

B. Extraction du titre

Le titre est obtenu en parcourant parallèlement les tailles et le texte , nous retenons la ligne de plus grande taille de police et nous nous assurons qu'il ne contienne pas le motif 'arXiv', grâce à notre structure de dictionnaire de liste de même taille nous parcourons également les numéros de page et une fois le titre identifié nous retenons son numéro de page et sa position dans le tableau

C. Extraction des références

Plutôt que de se baser sur les propriétés du titre, nous nous basons sur celles de la référence, elle est donc extraite en premier lieux, nous cherchons le motif 'reference' dans toutes les pages avec une une priorité pour celui qui sera écrit en majuscule (si oui valeur stockée dans un booléen sur vrai), dans le cas où il ne l'est pas nous regardons le motif le plus atypique selon sa police et sa taille, s'il est de la même taille que la majorité du texte, information obtenue grâce à notre liste de proportions d'apparitions des différentes tailles de police, nous l'indiquerons dans un booléen qui sera placé sur vrai sinon faux et nous considérerons qu'il se distingue par sa police. Nous stockons toutes ses informations (taille, position,police,taille). Les références seront donc toutes les lignes du tableau se trouvant après sa position.

D. Extraction de l'introduction, des discussions, de la conclusion et de l'abstract

L'extraction des références nous donne des informations cruciales pour le reste de l'analyse, nous nous en servons pour obtenir tous les 'titres' et 'sous-titres' dont feront parties les éléments souhaités, si le motif 'reference' se trouve en majuscule nous partirons sur le principe que tous les autres titres le sont aussi, s'il se distingue par sa taille de police, nous distinguerons tous les autres titres par une reconnaissance de mots clés appuyée cette fois-ci par la taille du motif 'reference' comme minimum, si ce n'est pas le cas nous pratiquons alors une recherche de mots clés appuyée par la police de caractère (nous retenons les mots clés de proportion d'apparition de leur police la plus faible).

Pour chaque mot clé détecté nous limiterons l'extraction entre lui et le prochain titre détecté qui pourra être un autre des mots clés (cas fréquent pour la discussion et la conclusion).

L'abstract est un cas particulier des car il ne respecte pas souvent la même taille de police ou la police des autres titres, donc nous faisons une recherche du motif dans la page du titre, s'il n'est pas trouvé alors nous prendrons le bloc de texte allant du premier titre (l'introduction dans la majorité des cas) jusqu'à la rencontre d'un retour chariot.

E. Extraction des auteurs, affiliations et adresses mail

- Les adresses email : elles sont recherchées dans toute la page du titre (numéro ayant été stockée lors de son extraction), nous recherchons ensuite à l'aide d'expression régulière les adresses mails.
- Les auteurs : ils sont cette fois-ci d'abord extraits à partir des métadonnées (cela étant une des fonctionnalités de pymupdf), mais s'ils ne s'y trouvent pas ils sont alors déduits des adresses mails en les découpant et en travaillant sur leurs préfixes.
- Les affiliations : leur recherche est restreinte quant à elle, à la zone entre l'abstract et du titre par reconnaissance de mots clés tels que 'laboratoire' ou encore 'université', appuyée par des expressions régulières pour clôturer leur extraction une à une.

5 Ecriture du fichier texte et XML

Les différents résultats de nos extractions peuvent être résumés dans un fichier texte ou XML, le fichier texte ne pose pas de souci dans son écriture sachant qu'il est moins fourni que le fichier et XML, le fichier XML lui requiert une syntaxe particulière (langage balisé), nous nous assurons donc de modifier ou supprimer les caractères inappropriés au format XML

6 Réalisation du menu contextuel

Le programme est lancé sur un terminal d'où il est ensuite possible de suivre les instructions qui y sont fournies syntaxe : <python3 Main.py rep1 rep2> De là nous proposons de résumer soit un ou plusieurs fichier de rep1 soit tout le répertoire avec '*'.

7 Résultat

Suite aux modifications apportées, les résultats obtenues ont connu une hausse de 28

Comparateur classique :

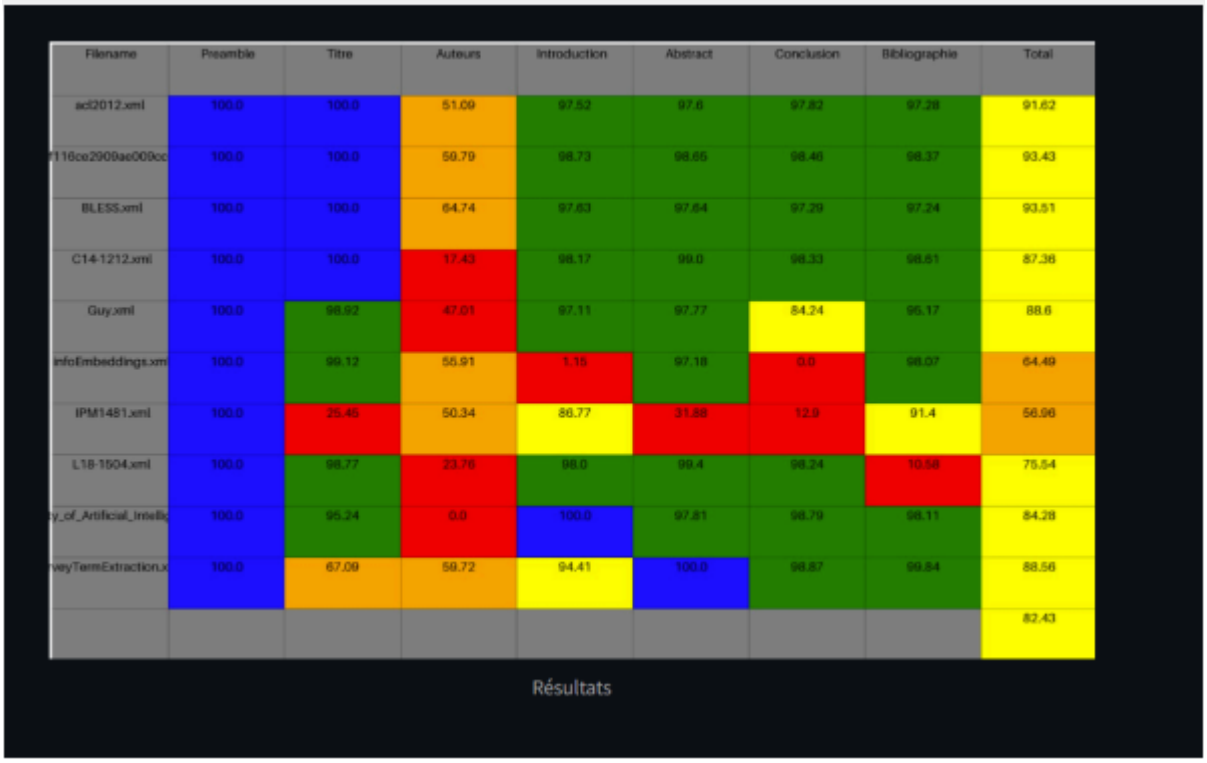
Parser V2 :

acl2012.xml	84.4%
b0e5c43edf116ce2909ae009cc27a1546f09.xml	92.39%
BLESS.xml	73.31%
C14-1212.xml	84.09%
Guy.xml	84.94%
infoEmbeddings.xml	52.25%
IPM1481.xml	66.75%
L18-1504.xml	71.62%
On_the_Morality_of_Artificial_Intelligence.xml	82.66%
surveyTermExtraction.xml	91.24%
Total	78,419%

Parser V1 (retravaillé dans un premier temps) :

acl2012.xml	75.72%
b0e5c43edf116ce2909ae009cc27a1546f09.xml	81.44%
BLESS.xml	84.25%
C14-1212.xml	82.92%
Guy.xml	71.37%
infoEmbeddings.xml	70.2%
IPM1481.xml	52.22%
L18-1504.xml	79.33%
On_the_Morality_of_Artificial_Intelligence.xml	53.02%
surveyTermExtraction.xml	55.75%
Total	70,62%

Comparateur 2023-2024 :



Filename	Preamble	Titre	Auteurs	Introduction	Abstract	Conclusion	Bibliographie	Total
acl2012.xml	100.0	100.0	51.09	97.52	97.6	97.82	97.28	91.62
f116ce2909ac009cc	100.0	100.0	59.79	98.73	98.65	98.46	98.37	93.43
BLESS.xml	100.0	100.0	64.74	97.63	97.64	97.29	97.24	93.51
C14-1212.xml	100.0	100.0	17.43	98.17	99.0	98.33	98.61	87.36
Guy.xml	100.0	98.92	47.01	97.11	97.77	84.24	95.17	88.6
nfoEmbeddings.xm	100.0	99.12	56.91	1.15	97.18	0.0	98.07	64.49
IPM1481.xml	100.0	25.45	50.34	86.77	31.88	12.9	91.4	56.96
L18-1504.xml	100.0	98.77	23.76	98.0	99.4	98.24	10.58	75.54
y_of_Artificial_Intelli	100.0	95.24	0.0	100.0	97.81	98.79	98.11	84.28
veyTermExtraction.x	100.0	67.09	59.72	94.41	100.0	98.87	99.84	88.56
								82.43

Résultats

En plus d'un gain de précision dans l'extraction nous avons grâce à nos parcours parallèles et non successifs des temps d'exécutions moins longs.

8 Conclusion

Pour le développement de ce parser, nous avons procédé en 2 temps. Il y a tout d'abord eu une première version utilisant les mots clés comme "Abtract" ou Introduction" par exemple qui nous permettaient de récupérer les sections recherchées. Or, nous nous sommes rendu compte que cette méthode nous posait des problèmes de précision et nous freinait dans l'augmentation du pourcentage de réussite. Ainsi, nous avons décidé de procéder d'une autre manière en analysant le fichier pour mettre toutes les informations nécessaires dans dictionnaire. Dans celui-ci, nous délimitons les différents champs recherchés grâce à une reconnaissance des titres du document même (par la taille de la police notamment ou la police elle-même selon leur proportion d'apparition). Pour finir, les points faibles de notre parser sont l'extraction des auteurs et des affiliations.