Projet de Développement

Cadou Valentin, Diallo Boubacar, Gareche Amirouche, and Ngoma Sita Dorian

cadou.e1800104@etud.univ-ubs.fr,
diallo.e1704214@etud.univ-ubs.fr,
gareche.e1704251@etud.univ-ubs.fr,
ngoma-sita.e182705@eutd.univ-ubs.fr.

Université Bretagne Sud, 56000 Vannes, France

1 Abstract

Les chercheurs de l'IRISA pour avancer dans leurs recherches, doivent lire les articles scientifiques publiés dans le monde. Malheureusement, ils n'ont pas le temps de tout lire et voudraient disposer d'un système qui présente un aperçu de l'article. Notre projet consiste donc à écrire un programme que l'on a choisi de réaliser en langage python. Celui-ci aura pour mission de produire un résumé des sections de l'article PDF dans une version texte bien découpée (comprenant les sections présentes parmi le titre, les auteurs, l'abstract, l'introduction, le corps, la conclusion, les discussions et les références).

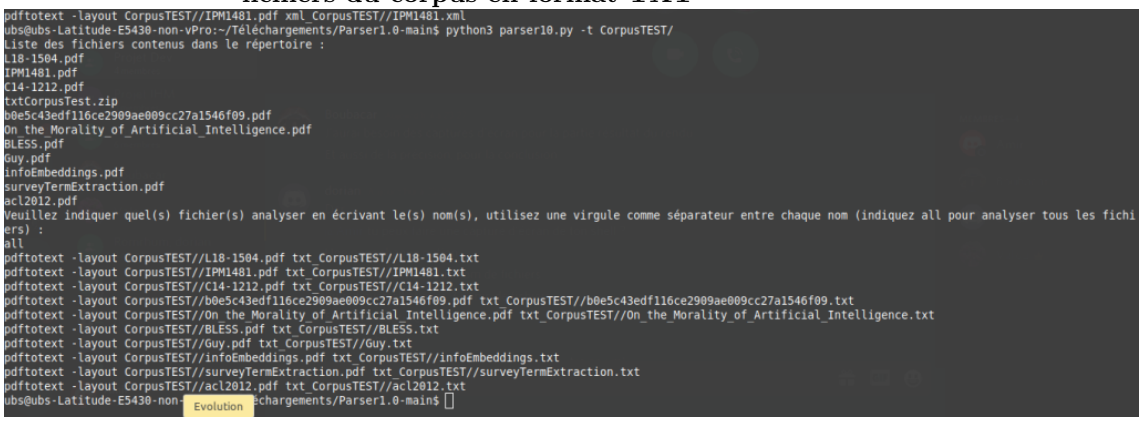
2 Méthode

Le projet a été développé suivant une méthode agile composée de cinq sprints. Tout d'abord, lors du sprint 1, nous avons choisi l'outil pdftotext pour faire la conversion des articles scientifiques en format texte. Le choix du langage python s'est imposé pour son efficacité de traitement des langues. Nous avons placé progressivement les versions intermédiaires et le système final sur Github. Ce dernier prend en entrée un dossier contenant les fichiers PDF et crée un sous-dossier, afin de déposer les sorties en plein texte portant le même nom que les PDF. Nous avons par la suite, ajouté une nouvelle sortie XML. Ainsi, au moment de lancer le parseur, nous avons utilisé un nouvel argument (-t — -x) pour choisir le format de sortie (-t correspondant à la version TXT et -x à la version XML). Suite à cet ajout, un second sous-dossier (qui va contenir les fichiers résumés) est créé pour la version XML dès lors que l'argument -x est renseigné. Ces fichiers doivent être composés du titre, des auteurs avec leur

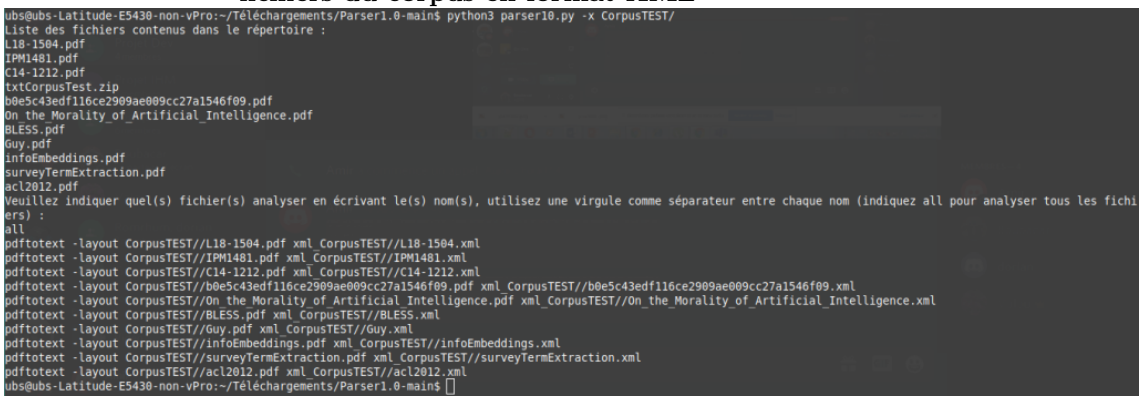
adresse mail et leur affiliation, de l'abstract, de l'introduction, du corps, de la conclusion, des discussions et des références. Pour terminer, nous avons exploité un nouveau corpus composé de 10 fichiers PDF afin de mesurer la précision de notre système sur celui-ci.

3 Résultats

-fichiers du corpus en format TXT



-fichiers du corpus en format XML



-exemples de calcul de la précision

```
ubs@ubs-Latitude-E5430-non-vPro:~/Téléchargements/Parser1.0-main$ python3 parser_precision.py xml_CorpusTEST/L18-1504.xml Resume_corpus_test/L18-1504.xml
L18-1504.xml precision :

titre : 100.0

auteurs : 98.94179894179894

abstract : 100.11547344110853

conclusion : 98.94291754756871
```

```
ubs@ubs-Latitude-E5430-non-vPro:~/Téléchargements/Parser1.0-main$ python3 parser_precision.py xml_CorpusTEST/surveyTermExtraction.xml Resume_corpus_test/surveyTeamExtrac
tion.xml
surveyTermExtraction.xml precision :

titre : 100.0

auteurs : 97.8102189781022

abstract : 99.84984984984985

conclusion : 101.669449081803
```

4 Conclusion

En somme, quatorze semaines et cinq sprints ont été nécessaires pour la réalisation de ce projet. Nous avons utilisé principalement deux ordinateurs équipés du système Linux.

Nous avons rencontré différents problèmes : tout d'abord, la structure des documents (lors du sprint 2) qui était différente d'un document à l'autre. On a donc dû prendre en compte toutes ces différences. Le deuxième problème rencontré a été lors du sprint 5 puisqu'il nous fallait faire évoluer notre parseur afin de convertir un nouveau corpus sans modifier les résultats obtenus avec le premier.

Nous avons également fait face à des bugs tout au long de la réalisation mais que nous avons réussi à corriger lors de nos réunions.

Finalement, nous avons répondu aux attentes des chercheurs de L'IRISA puisque notre parseur est écrit en langage python qui est particulièrement adapté pour ce projet. Ce dernier reposant sur le traitement des langues. De plus, le temps d'exécution est relativement rapide bien que le corpus comporte un nombre restreint de documents. Deux formats de sortie sont possibles (TXT et XML) répondant ainsi à une fonctionnalité très attendue. Pour finir, nous obtenons un bon niveau de précision, plus de 90 pourcents pour chaque document, ce qui montre le bon découpage des fonctions.