Projet de Developpement

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.

Univerité Bretagne Sud, 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 à avoir un système qui les présente un aperçu de l'article. Notre projet consiste donc à faire un programme qu'on a choisit de faire en python qui fait un resumé des sections de l'article PDF dans une version texte bien découpée(le titre, les auteurs, l'introduction, le développement,la conclusion etc.)

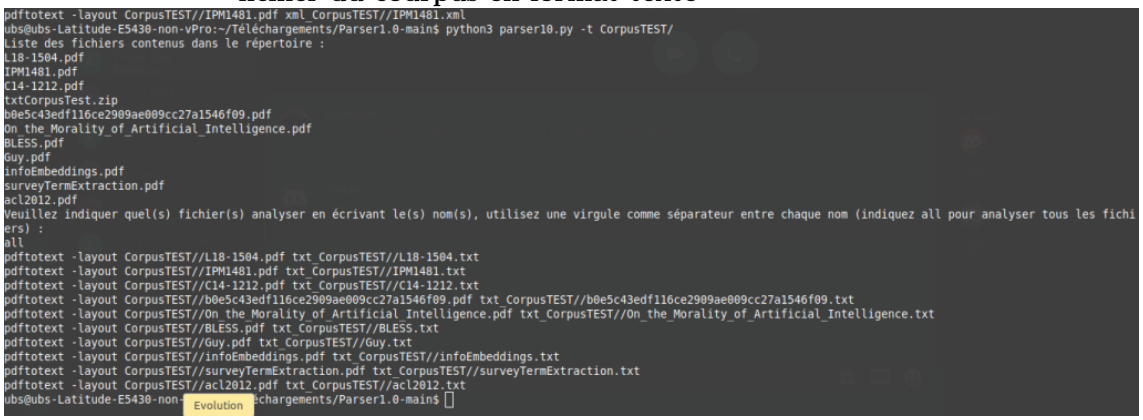
2 Methode

le projet a ete developpé suivant une methode agile composés de cinq sprints . Pour commencer Nous avons choisi l'outil pdftotext pour faire la conversion des aricles scientifiques en format texte et python comme langage de programmation pour faire notre programme parce qu'on le maitrise plutot bien et aussi pour son efficacité en terme de temps d'execution.Nous avons placés progressivement les versions intermédiaires et le systeme final sur Github. Le système prend comme entrée un dossier contenant les fichiers PDF et créer un sous-dossier pour déposer les sorties en plain texte avec le même noms que les PDF.nous avons ensuite rajouter une nouvelle sortie .xml. Ainsi, au moment de lancer le parseur, nous avons ajouter un argument (-t — -x) pour bien choisir le type de sortie : soit des fichiers dans une version texte (-t) , soit au format XML (-x). En sortie nous avons un repertoire composé de fichier resumé des differents pdf en .txt ou .xml ayant les memes noms que les fichiers pdf.Ces fichiers doivent etre composés du Titre,des auteurs et de leur affiliation, de l'abstract, de l'introduction, du corps

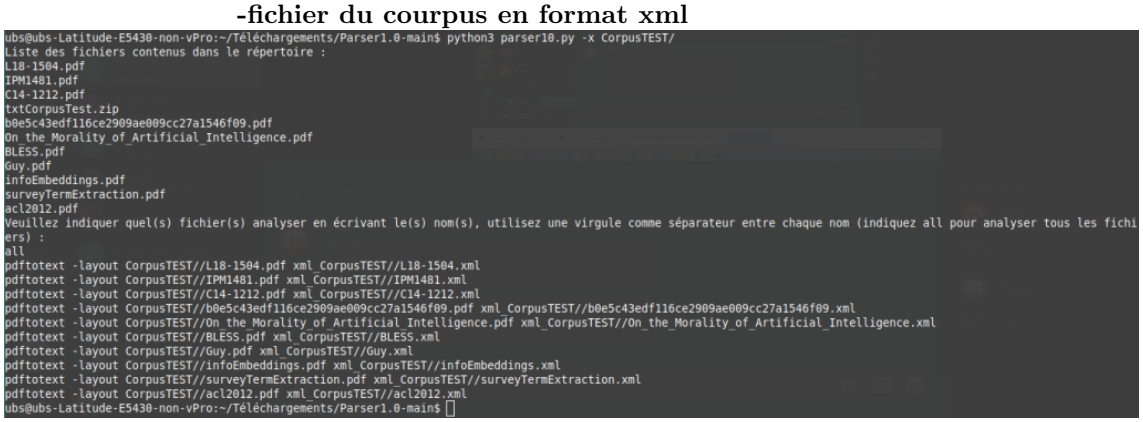
du document, de la conclusion,de la partie discussion et des references. Pour terminer nous avous utilisés un nouveau corpus composés de 10 fichiers pdf pour nous permettre de mesurer la precision de notre systeme.

3 Resultats

-fichier du courpus en format texte



-fichier du courpus en format xml



-exemples de calcul de precision

```
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 semaine et cinq sprints nous ont eté necessaire pour la realisation de ce projet. Nous avons utilisé principalement deux ordinateurs equipés du systeme linux.

Nous avons également rencontrer diffrents problemes, tout d'abord lors du sprint 2 au niveau de la structure des documents qui etait differente d'un document a l'autre on a donc du prendre en compte toutes ses differences.Le deuxieme probleme rencontrer a ete lors du sprint 5 puisqu'il nous fallait faire evoluer notre parseur afin de convertir un nouveau corpus sans modifier les resultats obtenus avec le premier corpus.

Nous avons fait face à des bug tout au long de la realisation mais que nous avons reussi à corriger lors de nos reunions.

Finalemant nous avons repondu aux attentes des chercheurs de L'IRIZA puisque notre parseur est ecrit en langage python et donc particulièrememnt adapté pour se projet qui repose sur le traitement des languees ,de plus le temps d'execution est relativement rapide bien que le corpus comporte un nombre important de document Deux formats de sortie sont possibles (format texte et xml) repondant à une fonctionalite très attendu ,pour finir nous obtenons un bon niveau de précision, plus de 90 pourcent pour chaque document, ce qui montre le bon découpage des fonctions.