Parseur d'articles scientifiques au format texte

Aribard Antoine
Université Bretagne Sud
aribard.e2103334@etud.univ-ubs.fr

Brzustowski Matthias
Université Bretagne Sud
brzustowski.e1802762@etud.univ-ubs.fr

Cart-Grandjean Louis
Université Bretagne Sud
cart-grandjean.e1803972@etud.univ-ubs.fr

Casanova Arthur
Université Bretagne Sud
casanova.e1901754@etud.univ-ubs.fr

15 mai 2022

Abstract

Dans le cadre du projet de développment à réaliser ce semestre, notre équipe avait pour objectif la réalistion d'un parseur d'articles scientifiques au format texte. Cet analyseur a pour but de permettre aux scientifiques du laboratoire de l'IRISA un accès simple et concret aux informations dont ils ont besoin.

Ce projet a été entièrement réalisé en suivant la méthode agile SCRUM. Le travail a été répartie en plusieurs "Sprints", et l'équipe a été en mesure d'adapter son travail en fonction des nouvelles demandes au cours de l'ensemble du semestre.

Ce rapport de projet va permettre d'expliquer le fonctionnement général de l'analyseur, ainsi que les résultats obtenus lors de tests de précision.

1 Méthode (explication du système)

Le parseur a été entièrement réalisé à l'aide du langage de programmation Python. Il est exécutable en ligne de commande depuis un terminal GNU/Linux. Lors de l'exécution, le programme doit prendre plusieurs paramètres : le chemin d'accès du dossier contenant les

articles à parser , le choix du format texte en sortie (.txt ou .xml), ainsi que les noms des différents articles à analyser.

1.1 Exécution du programme

Pour exécuter le parseur, plusieurs étapes sont à réaliser :

1. Ouvrir un terminal, puis se placer dans le répertoire contenant le programme python avec la commande suivante : 'cd'
2. Entrer la commande suivante : 'chmod a+x script.py'. Celle-ci indique que le programme est un fichier exécutable, et que tous les utilisateurs ont les droits d'exécution.
3. Entrer la commande 'python3 script.py <chemin d'accès> <-t>/<-x> <n° articles>.

   L'argument <chemin d'accès> est l'adresse à laquelle se trouve le répertoire à parser.

   L'argument <-t>/<-x> permet de spécifier le format des fichiers en sortie : txt ou xml. Il est possible d'obtenir les deux formats en entrant les arguments <-t> puis <-x>.

   L'argument <n° articles> permet de spécifier quels sont les articles à sortir au format demandé. Il se compose des numéros des articles séparés par une virgule et sans espaces. Si on souhaite traiter tous les fichiers, cet argument doit être vide ou contenir le texte 'all'.

   Si l'on ne souhaite qu'un seul format de sortie (.pdf ou .xml), et que l'on veut parser seulement certains fichiers (et donc que l'on utilise l'argument <n°articles>), il faudra alors saisir deux fois l'argument <-t> ou <-x>.

   Le programme efface puis remplace dans ce répertoire le sous-répertoire nommé 'result' s'il existe déjà, ou le crée sinon. Il va ensuite créer les fichiers txt/xml dans ce sous-dossier.

1.2 Fonctionnement du programme

Le programme analyse le répertoire entré en paramètre, contenant les articles au format PDF à traiter, puis, pour chaque article (demandé en paramètre), génère ce même article, au format TXT ou XML (précisé en ligne de commande). Pour chaque article à parsé, le programme réalise une recherche et un traitement de chaque section que peut contenir l'article, à savoir :

- Preambule (nom du fichier d'origine)
- Titre de l'article
- Auteurs (Nom, affiliation et adresse mail)
- Résumé de l'article
- Introduction

- Développement de l'article
- Conclusion de l'article
- Discussion de l'article
- Références bibilographiques de l'article

Pour conserver le contenu ainsi qu'une structure de texte la plus précise selon l'originale,le programme analyse les paragraphes contenues dans le texte, les tabulations, ainsi que les mot-clés (tel que 'Abstract' par exemple) pour savoir à quelle section correspond le segment traité.

2 Résultats

Une fois le programme implémenté et fonctionnel , une évaluation était nécessaire pour connaitre la précision du parseur, la structure ainsi que le contenu de l'article devant être respéctés au maximum.

Pour ce faire, le package Python 'Sequence Matcher' a été mis à disposition, dans le but de comparer des paires de séquences d'entrée. De plus, 10 articles scientifiques ont été fournis dans le but de calculer une moyenne de précision du parseur sur ces 10 articles. La formule permettant de calculer la précision du parseur sur un article est la suivante :

$$PrecisionTotal = \Sigma(PrecisionParSections)$$

3 Conclusion

Ce projet réalisé durant le semestre nous a permis d'apprendre à travailler en respectant la méthodologie SCRUM, méthode de travail désormais très répendu dans le monde du travail, de par sa capacité à optimiser la productivité.

Tous réunis régulièrement à chaque réunion de plannification de Sprint, mais aussi lors de nombreuses Daily Scrum, nous avons ainsi pu mettre en relation nos différentes tâches à réaliser et le travail fourni de chacun, permettant ainsi de travailler de la façon la plus productive possible, en plus d'être prêt à toute adaptation nécessaire quant aux nouveaux besoin de nos clients.