Rapport du projet PDF to Text du module Génie Logiciel

Lucas Peplinski / Sofia Ould Ammar / Dylan Mastrolia
lucas.peplinski@alumni.univ-avignon.fr
sofia.ould-ammar@alumni.univ-avignon.fr
dylan.mastrolia@alumni.univ-avignon.fr

Abstract

De nos jours, la majorité des documents sont sous format PDF, un format graphique idéal pour tous, permettant de rendre ses documents formel et les partager à travers le monde.

Mais ce format "graphique" propose des problèmes pour les algorithmes d'analyse de document, ayant un format trop complexe pour être analysé, à cause de cela, il est parfois nécessaire de retranscrire ces documents sous d'autres formats ; tel que le plain text ou le xml.

1 Introduction

Les chercheurs au Laboratoire Informatique d'Avignon (LIA) doivent lire des articles scientifiques publiés partout dans le monde, ils n'ont pas le temps de tout lire et voudraient avoir un système qui les présente un aperçu de l'article. Cela permettra de leur faire gagner du temps. Mais les fichiers pdf sont loin d'être facile à analyser par les systèmes de Traitement Automatique de Langues (TAL)

Nous avons donc réalisé un algorithme permettant de transcrire les fichiers PDF, au format text ou xml pour faciliter cette analyse.

Des outils tel que pdftotext ou pdf2txt étaient possible pour réaliser cet objectif, et après quelques recherches et phases de test, nous avons finalement décidé d'utiliser l'outil pdftotext car il propose des méthodes de conversions plus adaptées, avec un meilleur formatage des caractères.

2 Méthode (explication du système)

Le programme de conversion a été entièrement réalisé avec le langage de programmation Python. Le projet fonctionne sous GNU/Linux en ligne de commande en exécutant le script main.py suivi des arguments comme le chemin des fichiers PDF à convertir, et le format choisi (.txt ou .xml). Le script main.py utilise 2 scripts auxiliaires, le 1er script-xml.py gère la conversion des fichiers à parser en xml, et le 2nd script-plaintext.py pour la conversion en text.

```
python3 main.py <répertoire contenant les fichiers PDF> -t ou -x
```

2.1 Fonctionnement du programme

Il faut d'abord se placer sur le répertoire où se trouve le projet.

Le script principal main.py permet de choisir la conversion qu'on a envie de faire donc soit .txt ou .xml avec également le choix des fichiers à convertir. Le système de choix inclut 3 différents types, si l'utilisateur tape sur la clé esc il sort du choix, all il choisit de convertir tous les fichiers qui se trouvent dans le répertoire, y/n pour convertir ce fichier en particulier ou pas.

Le script script-plaintext.py découpe d'abord l'abstract et ajoute le reste est ajouté en utilisant le programme pdftotext un outil Linux à installer avec cette commande **sudo apt-get install poppler-utils**, avec une mise en page.

Le script script-xml.py gère la conversion d'un fichier HTML (généré à partir du fichier pdf original) et ensuite le convertit en xml. Cette étape a été d'une certaine aide pour pouvoir prendre chaque partie à partir des balises HTML. Il récupére ainsi les informations comme le titre, les auteurs et leurs e-mails, l'abstract, la conclusion ou discussion, et les références bibliographiques à partir du fichier HTML. Ces informations seront ensuite affichées en xml.

À la fin de la conversion en xml, le fichier contiendra les sections découpées suivantes :

- Preamble - Nom du fichier originel
- Titre
- Auteurs + Emails des auteurs
- Abstract
- Introduction
- Corps du développement du papier
- Conclusion
- Discussion
- Bibliographie

3 Resultats

Après avoir manipulé les fonctionnalités de pdftotext pendant plusieurs semaines, nous avons finalement atteint des fichiers resultant de l'algorithme, ayant au moins un tiers d'erreur quant à la délimitation des différentes parties. Malgré la forte détection correcte de certains

éléments, tel le titre, les auteurs, ou les références, certaines parties plus instables dans les documents pdf, tel que l'abstract, l'introduction ou les résultats, sont parfois mal découpés dû à un manque de régularité dans l'écriture des documents originels.

Le taux de précision final en frontières strictes est de 53%.
Et le taux de précision final en frontières souples est de 60%.

Il est bon de noter que "A Benders Decomposition Approach toCorrelation Clustering.pdf" fait bugger le script à cause des espaces dans le nom du fichier, la précision de ce document a donc était de 0.

4 Conclusions

En conclusion, les résultats sont majoritairement influencés à la fois par le manque de régularité du format des documents, et ainsi que par le manque de temps attribué au projet qui aurai permit de prendre en compte plus de cas particulier dans les parties instables des documents.

Au final, la version actuel est tout de même intéressante, notamment au format XML, qui permet de rapidement récupérer les informations générales des documents, tel que le titre, les auteurs et la bibliographie par exemple.

Le format plain text est aussi utile du point de vue humain car il propose un format lissible et propre correspondant à l'organisation initiale du PDF, mais dans un format éditable. Cela pourait notamment être utilisé pour écrire des notes personnels sur un document scientifique.