CHAUVEL LAUREN
KAHASHA GUY-ALEXANDRE
RANDRIMIHAJA LALATIANA

04/02/2024

SPRINT 1

1.Commande pdftotext

Utilisation dans un terminal :

pdftotext [options] [*PDF-file] [text-file*]

Utilisation avec Python :

```
import subprocess
subprocess.run([“pdftotext”, “-layout”, pdf_path, output_path], check=True)
```

Les options qui pourraient nous être utiles :

-layout : permet de garder la mise en forme du texte

Par exemple, si le PDF est sous la forme de 2 colonnes, le fichier texte le sera aussi.
Inconvénient : Pour certaines pages, c’est la 2e colonne qui est réécrite dans le fichier texte avant la première ce qui ne garde pas l’ordre du texte tel qu’on le souhaiterait.

-raw : garde les chaînes de caractères dans le même ordre

Test sur le fichier ‘kessler94715.pdf’ qui est un fichier à 2 colonnes contenant un tableau.
Avantage : l’ordre des mots est bien conservé même lorsque l’on passe d’une colonne à l’autre.

Inconvénients : les grandes parties ne sont pas séparées par des espaces donc peuvent être plus difficiles à repérer. De plus, le tableau du fichier est affiché sous la forme suivante :

```
Collection NC CCTP
Total number of documents 58 402 3665
Without pre-processing
Total number of words 130 309 230 962 734
Total number of different words 93 000 20 6264
Average words/document 125.3 63 018.48
Fig. 2. statistics of the collection
```

Il est donc difficile de séparer les nombres et de savoir que ce qui doit être lu sur la ligne Total number of words est 130 309 et 230 962 734.

-table : similaire à l’option -layout mais optimisé pour les tableaux

Test sur le fichier ‘kessler94715.pdf’ qui est un fichier à 2 colonnes contenant un tableau.

Avantages : On garde bien la mise en forme et les colonnes restent bien dans l'ordre (pas comme pour -layout). Le tableau est plus lisible car les données sont espacées.

Inconvénients : On ne voit pas bien la séparation entre 2 pages. Il y a des problèmes de mise en forme à certains endroits :

will	be	removed	in	the	next	module.	We	do	not include a
stop	list	to	keep	n-grams	with	prepositions,	for	the	purpose
described			in	the	remainder.	Then,	we	tokenize	the entire
collection			before		using	TreeTagger	[22]	to	get the part-of-

a defined threshold. We varied this threshold between 1 and 50 and results presented in Section VI-B have been obtained with a threshold of 10.

2.Commande pdf2txt

Utilisation dans un terminal :

pdf2txt [options] [text-file] [PDF-file]

Utilisation avec Python :

from pdfminer.high_level import extract

extract(pdf_file_path)

Options de Lancement :

-o : output.txt spécifie le fichier de sortie pour le texte extrait.

-p : Spécifie les numéros de pages à extraire, par exemple -p 1,2 pour les pages 1 et 2.

-m : Le nombre maximum de pages à extraire

-V : extraire du texte vertical lors du traitement des documents PDF

le texte qui est disposé verticalement sur la page plutôt que de manière horizontale.

-t : Choix du format de sortie (text, html, xml, tag). Par exemple, -t html pour la sortie au format HTML.

-c : Spécifie le codec à utiliser pour le texte de sortie, par exemple -c utf-8.

-Y layout_mode, --layout_mode layout_mode : Spécifie le mode de mise en page, qui peut être 'normal', 'exact', 'loose' ou 'continuous'. (Ne marche pas sur les pdf du corpus donné)

Avantages :

Utile pour une mise en page complexes, pour du texte dans des images, ou nécessite une préservation précise de la structure.

Fourni par PDFMiner et plus flexible pour l'intégration dans environnement Python.

3.CONCLUSION

-**Pdftotext :** On constate que pdftotext est limité pour certaines mises en forme plus complexes et son éventail d'options est assez restreint

-Pdf2txt : Fournit un résultat correct pour la majorité des fichiers et possède beaucoup plus d'options

Eu égard des faits cités précédemment, nous avons décidé d'utiliser PDF2TXT qui, selon nous, par son large éventail d'option et sa meilleure gestion des mises en forme complexes, nous permettra de mieux gérer les cas particuliers