LouisonRABREAU – AlexisGOHIER – SimonCHEVE L3 INF/CMI

Sprint 1

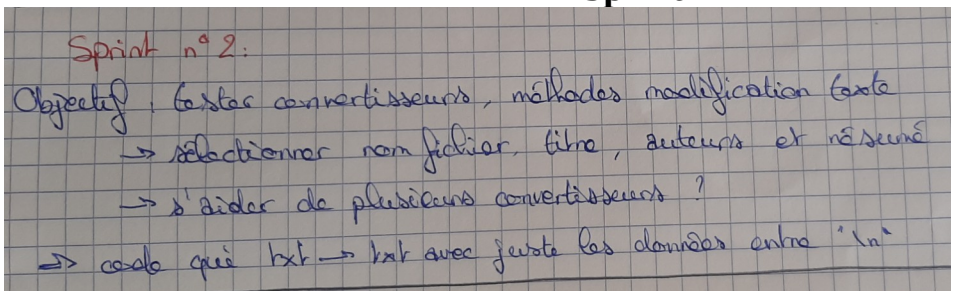
Sprint n°1:
Objectif : chercher convertisseur qui convient le plus
→ le plus efficace / simple à utiliser / lisible
→ le plus de fonctionnalités / options
⇒ rapport à rendre 1/2 pages
Logiciel : Python Outil : pdfminer, pypdf, pypdfium, pdftotext, pdf2txt,

Résumé des consignes et objectif du sprint1 : description de la compréhension des personnes de l'équipe après lecture du sujet, et choix du logiciel qui sera utilisé (Python), étant connu et déjà utilisé par l'équipe.

```
|           Poppler + pip install pdftotext -> Windows
import pdftotext
with open("input.pdf", "rb") as f:              # Load your PDF
            pdf = pdftotext.PDF(f)              #option "secret" if it's password-protected
with open('output.txt', 'w') as f:              # Save all text to a txt file
            f.write("\n\n".join(pdf))
- len(pdf) = nombre de pages

Debian, Ubuntu, and friends : "sudo apt install build-essential libpoppler-cpp-dev pkg-config python3-dev"
Fedira, Red Hat, and friends :"sudo yum install gcc-c++ pkgconfig poppler-cpp-devel python3-devel" ????
Windows : "conda install -c conda-forge poppler"
            https://github.com/jalan/pdftotext

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pip install PyPDF2
import PyPDF2
pdffileobj=open('input.pdf','rb')   #create file object variable and open it in "read binary" mode
pdfreader=PyPDF2.PdfFileReader(pdffileobj) #create reader variable that will read the pdffileobj
x=pdfreader.numPages                            #store the number of pages of this pdf file
pageobj=pdfreader.getPage(x+1)                  #create a variable that will select the selected number of pages
text=pageobj.extractText()                      #create text variable which will store all text datafrom pdf file
file1=open(r"output.txt","a")                   #save the extracted data from pdf to a txt file ('r' before the file path)
file1.writelines(text)

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pip install aspose-words
import aspose.words as aw
doc = aw.Document("document.pdf")               # Load your PDF
doc.save("pdf-to-text.txt")                                 # Save all text to a txt file

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pdf2txt
https://github.com/euske/pdfminer/blob/master/tools/pdf2txt.py

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pip install pypdf
from pypdf import PdfReader
reader = PdfReader("input.pdf")
text = ""
for page in reader.pages:
    text += page.extract_text() + "\n"

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pip install pypdfium2

import pypdfium2 as pdfium
text = ""
pdf = pdfium.PdfDocument(data)
for i in range(len(pdf)):
    page = pdf.get_page(i)
    textpage = page.get_textpage()
    text += textpage.get_text()
    text += "\n"
    [g.close() for g in (textpage, page)]
pdf.close()
```

Recherches sur les differents convertisseurs de pdf → txt sur Python (Recherche_de_parseurs.txt) : comment les utiliser et quelques options.

Sprint 2



Résumé des consignes et objectif du sprint2 : description de la compréhension des personnes de l'équipe après lecture du sujet.

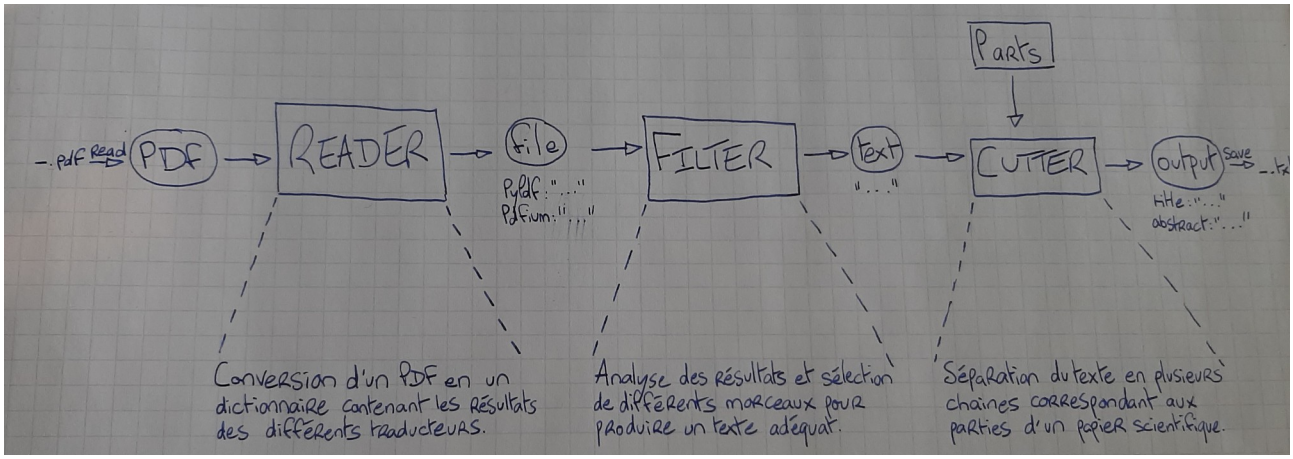


Schéma provisoire du fonctionnement/des différentes parties du projet : le input.pdf sera converti en plusieurs temporaire.txt en fonction des différents parseurs utilisés, les résultats seront analysés pour produire un input.txt, le plus correct possible. Un dernier programme découpera juste les élèments demandés en un output.txt