# Sprint 1 – Carnet du produit

Antoine ADAM

Sylvain BUCHE (Scrum Master)

Guillaume COBAT

**Langage utilisé :** Python 3

**Outils testés :**

- *Pdftotext*
- *PdfMiner*
- *PyPdf2*

**Pdftotext**

**Options :**

Nombreuses options disponibles en ligne de commande dont :

-layout : garde la structure originale du pdf
-table : affichage optimisé pour les données tabulaires
-raw : affiche les lignes de façon brutes
…

**Avantages :**

Garde la structure en colonne du pdf. Il garde à peu près correctement la structure des formules. De la documentation est disponible.

**Inconvénients :**

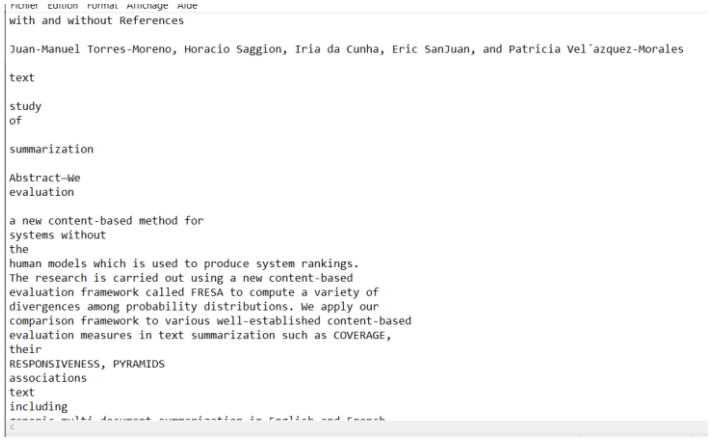
Impossible de retravailler le fichier directement après son extraction.

**PdfMiner**

**Options :** Aucune

**Avantages :**

La structure de la sortie est très optimisée et claire.




**Inconvénients :**

Aucune option n’est disponible.

L’outil n’est plus maintenu.

**PyPdf2**

**Options/Méthodes :**

- getDocumentInfo() : permet de récupérer les informations du PDF (auteur, date de création, etc.)

**Outpout 1 :**

{'/Author': 'Florian Boudin ; Marc El-Beze ; Juan-Manuel Torres-Moreno', '/Title': 'A Scalable MMR Approach to Sentence Scoring for Multi-Document Update Summarization'}

**A Scalable MMR Approach to Sentence Scoring for Multi-Document Update Summarization**

**Florian Boudin** [♮] and **Marc El-Bèze** [♮]
[♮] Laboratoire Informatique d'Avignon
339 chemin des Meinajaries, BP1228,
84911 Avignon Cedex 9, France.
florian.boudin@univ-avignon.fr
marc.elbeze@univ-avignon.fr

**Juan-Manuel Torres-Moreno** [♮,♭]
[♭] École Polytechnique de Montréal
CP 6079 Succ. Centre Ville H3C 3A7
Montréal (Québec), Canada.
juan-manuel.torres@univ-avignon.fr

**Outpout 2 :**

{'/Author': '', '/Title': ''}

A Survey on Automatic Text Summarization

Dipanjan Das André F.T. Martins
Language Technologies Institute
Carnegie Mellon University
{dipanjan, afm}@cs.cmu.edu

November 21, 2007

Dans l'exemple 1, on arrive à récupérer les noms des auteurs et le titre, ce qui ne fonctionne pas dans l'exemple 2.

- getFields()
- getPageLayout()
  et plus encore.

**Avantages :**

Beaucoup d'options pour retravailler le texte et extraire des informations.

**Inconvénients :**

La lecture des mots en italique est parfois ratée, comme ici avec *significant* qui devient *ant* :

previous two works. Two other features were used: the presence of *cue words* (presence of words like *significant*, or *hardly*), and the *skeleton* of the document (whether the sentence is a title or heading). Weights were attached to each of these

```
previoustwoworks.Twootherfeatureswereused:thepresenceof
cuewords
(presenceofwordslike
ant
,or
hardly
),andthe
skeleton
ofthedocument
(whetherthesentenceisatitleorheading).Weightswereattachedtoeachofthese
```

L'option permettant de récupérer les informations principales du pdf ne fonctionne pas dans tous les cas.

## De part ses options permettant l'extraction aisée des informations du PDF, nous pensons utiliser PyPdf2 pour le projet.