OpenMinted PDF Extractor

1. Component Description

The OpenMinted PDF Extractor is a stand-alone component used for extracting textual, pictorial and tabular information from documents following the Portable Document Format standard [1: https://www.adobe.com/devnet/pdf/pdf_reference_archive.html].

The component is installed and run locally over Linux systems. It accepts as input a directory path, where the PDF files to be processed are located.

The component creates a directory for each input document, which contains the following items:

- A plain text file containing the document’s metadata (e.g. author(s), creation date, title, etc.);
- A plain text file containing the raw text retrieved from the document;
- A plain text file containing the document’s outline;
- An XML file describing the references (if any) found in the document;
- A BibTex document describing those references that include a DOI link, if such references exist;
- Any figure found in the document as image files, in the format found within the document;
- A JSON document containing information on the aforementioned figures, as well as, the tables (if any) found in the input document. More specifically, the document defines *Figure* and *Table* objects. Each *Figure* object declares the caption and size and positioning metadata for the respective figure. Similarly, each *Table* object declares the caption and positioning of the respective table within the PDF document. Additionally, it presents the content’s of the table’s cells in separate brackets.

2. Component Installation

2.1. Component Dependencies

The following frameworks and tools are required before installing and running the component:

- A distribution of Python v2. The component was tested with the latest v2.7 version. Python is normally pre-installed in most Linux distributions;
- The PDF Miner Python library [2: https://pypi.python.org/pypi/pdfminer/].
- The Poppler Python library [3: https://poppler.freedesktop.org/].
- The pdf-extract tool [4: https://github.com/CrossRef/pdfextract].
- The PDF Figures tool [5: https://github.com/allenai/pdffigures].

2.2. Installation of Dependencies

2.2.1. Python v2

Python is normally pre-installed in most major Linux distributions. The component was tested with the latest (v2.7) version at the time of development. For checking if your Linux distribution comes with Python run the following command:

```
python -V
```

In case no Python version is found, it can be installed by giving the following commands:

```
• wget http://www.python.org/ftp/python/2.7.3/Python-2.7.3.tgz
• tar -xzf Python-2.7.3.tgz
• cd Python-2.7.3
• ./configure --prefix=/usr --enable-shared
• make
• make install
```

2.2.2. PDF Miner

The following actions are required to install PDF Miner:

- Download the source code from https://github.com/euske/pdfminer/ and unpack it a system directory.
- Run *setup.py*:

```
$ python setup.py install
```

- Do the following test:

```
$ pdf2txt.py samples/simple1.pdf
```

2.2.3. Poppler

For debian-based Linux systems, run the following command:

```
sudo apt-get install python-poppler
```

For RHEL-based linux systems, run the following command:

```
yum install poppler-utils
```

2.2.4. pdf-extract

In debian-based Linux systems, install the following packages:

```
sudo apt-get install build-essential

sudo apt-get install ruby-full

sudo apt-get install zlib1g-dev

sudo gem install nokogiri

sudo apt-get install libsqlite3-dev

sudo gem install specific_install

sudo gem specific_install https://github.com/EbookGlue/libsvm-ruby-swig.git

sudo gem install pdf-reader -v 1.1.1

sudo gem install prawn -v 0.12.0\
```

In RHEL-based Linux systems, install the following packages:

```
sudo yum groupinstall "Development Tools"

sudo yum install ruby

sudo yum install zlib-devel

sudo gem install nokogiri

sudo yum install sqlite-devel

sudo gem install specific_install

sudo gem specific_install https://github.com/EbookGlue/libsvm-ruby-swig.git

sudo gem install pdf-reader -v 1.1.1

sudo gem install prawn -v 0.12.0\
```

2.2.5. PDF Figures

- Download the source code from https://github.com/allenai/pdffigures and unpack it a system directory.
- Execute the following command:

```
make DEBUG=0
```

2.3. PDF Extractor Installation

Download the *PDF_Extractor.py2* script (https://github.com/openminted/uc-tdm-agriculture/tree/master/PDF%20Extractor) into a local directory. No further steps are required for using the component.

3. Relevant Data Processing Scenarios

3.1. Retrieve information from a collection of PDF documents

The user aims to analyze a collection of publications as PDF documents. To this end, she creates a directory in the local system's filesystem:

```
mkdir <dirname>
```

She copies the relevant PDF documents in the created directory:

```
cp <filename>.pdf <dirname>/.
```

Finally, she executes the PDF Extractor script:

```
sudo python PDF_Extractor.py2 <dirname>
```

Upon completion, the script produces the following items:

- A file <filename>.txt containing the raw text of the <filename>.pdf document;
- A <filename>.metadata text file containing metadata information for the respective PDF document;
- A <filename>.outline text file containing the outline of the respective PDF document;
- The figures found in the PDF document as image files. The naming convention for the images follows the pattern <filename>-<page number>-<increment>.
- A <filename>.references XML file, containing the references found in the respective PDF document.
- A <filename>.refs.bib BibTex file, containing the references associated with a DOI.
- A <filename>.json file, containing information on the figures and tables discovered in the respective PDF document.