build success

Parsr: Turn your documents into data!

中文

Parsr, is a minimal-footprint document (image, pdf) cleaning, parsing and extraction toolchain which generates readily available, organized and usable data for data scientists and developers.

It provides users with clean structured and label-enriched information set for ready-to-use applications ranging from data entry and document analysis automation, archival, and many others.

1. Getting Started / Installation

This section will quickly guide you through the installation process.

You can install Parsr either using Docker containers, or directly on your machine. You don't need to do both!

1.1. Docker Installation

Containers are already available on Docker Hub.

The documentation to build and run Docker containers is here.

1.2. Bare-Metal Installation

1.2.1. Installing Dependencies under Linux

Under a **Debian** based distribution:

```
sudo add-apt-repository ppa:ubuntuhandbook1/apps
sudo apt-get update
sudo apt-get install nodejs npm qpdf imagemagick graphicsmagick python-
pdfminer tesseract-ocr libtesseract-dev python3-tk ghostscript python3-pip
pip install camelot-py
```

Under **Arch** Linux :

```
pacman -S nodejs npm qpdf imagemagick graphicsmagick pdfminer tesseract
python-pip
pip install camelot-py
```

1.2.2. Installing Dependencies under MacOS

The package manager we suggest using under MacOS is homebrew. To install it, launch the following in a terminal

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Next, install the required dependencies:

```
brew install node qpdf imagemagick graphicsmagick tesseract tesseract-lang
```

To install the python based dependencies (pdfminer and camelot), install, first install `pip`:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

and then the dependencies:

```
pip install pdfminer.six
pip install ghostscript camelot-py
```

1.2.3. Installing Dependencies under Windows

1. We recommend using Chocolatey as the package manager for installing dependencies under Windows. To install Chocolatey, follow these instructions.
2. Download and install `node.js`
3. For the **pdfminer** extractor for pdfs, follow these steps.
4. Install `qpdf` and `imagemagick` using Powershell (Run as Administrator):

   ```
   choco install qpdf imagemagick
   ```

5. For table detection, install **camelot**.

1.2.3.1. Tesseract

You can download Tesseract 4.0 64-bit for Windows or check out other available formats on the wiki.

Then, you need to add tesseract.exe to your PATH: If you have install it in `C:\Program Files (x86)\Tesseract-OCR`, you can either add it using the user interface execute the following command in Powershell (Run as Administrator):

```
setx PATH "\$env:PATH;C:\Program Files (x86)\Tesseract-OCR" -m
```

1.3. Optional Dependencies

The following dependencies are **completely optional**, and their exclusion does not hinder the proper functioning of the Parsr pipeline.

The functions of each, as well as the installation process are are explained below:

1.3.1. MuPDF

MuPDF, in the Parsr platform is Used to fix certain error-prone or corrupt PDF files on input.

To install MuPDF, follow the steps corresponding to your environment:

- Under a **Debian** based distribution:

```
sudo apt-get install mupdf mupdf-tools
```

- Under **Arch** Linux:

```
pacman -S mupdf-tools
```

- Under MacOS:

```
brew install mupdf-tools
```

- Under Windows:

```
choco install mupdf
```

If MuPDF is not installed, a corrupt/unreadable PDF file at input will be left untreated. A message of such an occurrence will be logged.

1.3.2. Pandoc

Pandoc is a document format conversion program, used under Parsr to generate PDF files from an intermediate Markdown output after the cleaning operation in the pipeline.

To install Pandoc, follow the steps corresponding to your environment:

- Under a **Debian** based distribution:

```
sudo apt-get install pandoc
```

- Under **Arch** Linux:

```
pacman -S pandoc
```

- Under MacOS:

```
brew install pandoc
```

- Under Windows:

```
choco install pandoc
```

If Pandoc is not installed, the user will not be able to generate PDF files on output. Any configuration requiring a PDF file output will be ignored.

1.3.3. ABBYY FineReader

ABBYY FineReader is a proprietary high precision OCR solution for generating rich text from images. One can obtain the ABBYY FineReader Server from here.

ABBYY FineReader is an **optional dependency**, and it's absence should in no way hinder the everyday usage of Parsr's default OCR solution, tesseract.

2. Usage

You can use Parsr in different ways:

- Using the command line
- Using the API
- Using the demo web viewer

2.1. Install npm packages

```
npm install
```

2.2. Run

2.2.1. Configuration

The tool contains a pipeline of modules that process the document step by step and is highly configurable. To change it's default configuration, please refer to the configuration file documentation.

2.2.2. Demo: Web Viewer

To start the web viewer demo, simply run:

2.2.2.1. Under Linux/MacOS:

```
npm run start:web:vue
```

2.2.2.2. Under Windows:

In two different terminals, first:

```
npm run start:api
```

then in the other one:

```
cd demo/vue-viewer && npm install && npm run serve
```

Open localhost:8080 with your favorite browser to use the GUI.

2.2.3. Command Line Usage

Under Mac OS X, Linux:

```
npm run run:debug -- --input-file samples/t1.pdf --output-folder dist/ --
document-name example --config server/defaultConfig.json --pretty-logs
```

Under Windows:

```
cmd /C "npm run run:debug -- --input-file samples/t1.pdf --output-folder
samples --document-name example --config server/defaultConfig.json --
pretty-logs"
```

2.3. API

Install the API server with:

```
npm run install:api
```

And then start the API server with:

```
npm run start:api
```

You can then call endpoints on localhost:3001.

The documentation for the API can be found here.

2.4. Test

```

```

```
npm run test
```

3. ABBYY FineReader Server

The ABBYY FineReader is a high-precision OCR option provided to the users of the Parsr platform. It is to be noted that it is completely optional, and that the default OCR solution supported under Parsr is tesseract, which is a dependency of the solution.

3.1. Server Configuration

When ABBYY FineReader Server is chosen as the working OCR extraction solution, the following environment variables need to be set on the host running Parsr:

1. `ABBYY_SERVER_URL` : The network address of the ABBYY FineReader Server.
2. `ABBYY_SERVER_VER` : The major version number of the ABBYY FineReader Server. For example: 14 for ABBYY FineReader Server 14.01.
3. `ABBYY_WORKFLOW` : The name of the server 's workflow to be called to process the file.

On the side of the ABBYY FineReader Server, make sure the XML output is configured for the selected workflow:

1. Double click on the workflow to be used.
2. In the tab titled 'output', make sure the list of file formats exported contains the XML format if not, add it with the 'New' button.
3. Make sure the following settings are enabled on the XML format 's settings:
    1. Character Attributes
    2. Extended Character Attributes
    3. Coordinates of the Original Image
    4. Character Formatting

4. Dependencies Explanation

4.1. Base Dependencies

The following ***required*** dependencies need to be installed for Parsr to work properly:

1. `node.js` : The underlying framework upon which the platform is built.
2. `qpdf` : For reading password-protected PDFs.
3. `imagemagick` : For converting between file formats.

4.2. Extraction Dependencies

Depending upon the type of documents to be treated by the platform, one or multiple of the following dependencies should be installed.

If simple PDFs containing digital (or ***selectable***) textual elements are to be fed into the system, the **`pdfminer`** library needs to be installed.

If images (`jpg`, `png`, `tiff`, etc.) are to be used with the tool, then the tool also supports the use of the following two OCR based solutions as an underlying extraction module:

1. `tesseract` : Open source, support for over ~100 languages, Google's Tesseract is a free, on premise OCR solution. However, text formatting, or tabular data is not detected.
2. `ABBYY FineReader Server` : Proprietary OCR solution with extremely high recognition accuracy, formatting recognition and tabular data extraction. It is an optional dependency.

4.3. Optional Dependencies

The following *optional* dependencies may to be installed:

1. `mupdf-tools`: For error-correcting corrupt PDFs at input.
2. `pandoc`: Generate PDF files from an intermediate Markdown output after the cleaning operation in the pipeline.

5. Contribute

Please refer to the guidelines in CONTRIBUTING.md.

6. Third Party Licenses

Third Party Libraries licenses :

1. **QPDF**: Apache http://qpdf.sourceforge.net
2. **GraphicsMagick**: MIT http://www.graphicsmagick.org/index.html
3. **ImageMagick**: Apache 2.0 https://imagemagick.org/script/license.php
4. **Pdfminer.six**: MIT https://github.com/pdfminer/pdfminer.six/blob/master/LICENSE
5. **Tesseract**: Apache 2.0 https://github.com/tesseract-ocr/tesseract
6. **Camelot**: MIT https://github.com/camelot-dev/camelot
7. **MuPDF** (Optional dependency): AGPL https://mupdf.com/license.html
8. **Pandoc** (Optional dependency): GPL https://github.com/jgm/pandoc

7. License