tabula-py

Sep 09, 2023

Contents

1 Getting Started **3**

1.1 Requirements . . . 3

1.2 Installation . . . 3

1.3 Example . . . 4

2 FAQ **5**

2.1 tabula-py does not work . . . 5

2.2 I can't run from tabula import read_pdf . . . 5

2.3 I got an empty DataFrame. How can I resolve it? . . . 5

2.4 The result is different from tabula-java. Or, stream option seems not to work appropriately . . 6

2.5 Can I use option xxx? . . . 6

2.6 How can I ignore useless area? . . . 6

2.7 I faced ParserError: Error tokenizing data. C error. How can I extract multiple tables? . . . 8

2.8 I want to prevent tabula-py from stealing focus on every call on my mac . . . 8

2.9 I got ? character with results on Windows. How can I avoid it? . . . 8

2.10 I can't extract file/directory names with space on Windows . . . 8

2.11 I want to use a different tabula .jar file . . . 8

2.12 I want to extract multiple tables from a document . . . 9

2.13 Table cell contents sometimes overflow into the next row. . . . 9

2.14 I got a warning/error message from PDFBox including org.apache.pdfbox.pdmodel.. Is it the cause of the empty dataframe? . . . 9

2.15 I can't figure out accurate extraction with tabula-py. Are there any similar Python libraries? . . . 9

3 Contributing to tabula-py **11**

3.1 Code formatting and testing . . . 11

3.2 Documentation . . . 11

4 tabula **13**

4.1 High level interfaces . . . 13

4.2 Internal interfaces . . . 13

5 tabula.errors **15**

6 Indices and tables **17**

`tabula-py` is a simple Python wrapper of tabula-java, which can read table of PDF. You can read tables from PDF and convert them into pandas' DataFrame. tabula-py also converts a PDF file into CSV/TSV/JSON file.

We highly recommend looking at the example notebook and trying it on Google Colab.

For high-level API reference, see *High level interfaces*.

CHAPTER 1

Getting Started

1.1 Requirements

- Java
  - Java 8+
- Python
  - 3.8+

1.2 Installation

Before installing tabula-py, ensure you have Java runtime on your environment.

You can install tabula-py from PyPI with `pip` command.

```
pip install tabula-py
```

Note: conda recipe on conda-forge is not maintained by us. We recommend installing via `pip` to use the latest version of tabula-py.

1.2.1 Get tabula-py working (Windows 10)

This instruction is originally written by @lahoffm. Thanks!

- If you don’t have it already, install Java
- Try to run an example code (replace the appropriate PDF file name).

- If there's a FileNotFoundError when it calls read_pdf(), and when you type java on command line it says 'java' is not recognized as an internal or external command, operable program or batch file, you should set PATH environment variable to point to the Java directory.
- Find the main Java folder like jre... or jdk.... On Windows 10 it was under C:\Program Files\Java
- On Windows 10: **Control Panel** -> **System and Security** -> **System** -> **Advanced System Settings** -> **Environment Variables** -> Select **PATH** –> **Edit**
- Add the bin folder like C:\Program Files\Java\jre1.8.0_144\bin, hit OK a bunch of times.
- On command line, java should now print a list of options, and tabula.read_pdf() should run.

1.3 Example

tabula-py enables you to extract tables from a PDF into a DataFrame, or a JSON. It can also extract tables from a PDF and save the file as a CSV, a TSV, or a JSON.

```
import tabula

# Read pdf into a list of DataFrame
dfs = tabula.read_pdf("test.pdf", pages='all')

# Read remote pdf into a list of DataFrame
dfs2 = tabula.read_pdf("https://github.com/tabulapdf/tabula-java/raw/master/src/test/
↪resources/technology/tabula/arabic.pdf")

# convert PDF into CSV
tabula.convert_into("test.pdf", "output.csv", output_format="csv", pages='all')

# convert all PDFs in a directory
tabula.convert_into_by_batch("input_directory", output_format='csv', pages='all')
```

See example notebook for more detail. I also recommend reading the tutorial article written by @aegis4048 and another tutorial written by @tdpetrou.

Note: If you face some issues, we'd recommend trying tabula.app to see the limitation of tabula-java. Also, see *FAQ* as well.

CHAPTER 2

FAQ

2.1 `tabula-py` does not work

There are several possible reasons, but `tabula-py` is just a wrapper of tabula-java , make sure you've installed Java, and you can use `java` command on your terminal. Many issue reporters forget to set PATH for `java` command.

You can check whether tabula-py can call `java` from the Python process with `tabula.environment_info()` function.

2.2 I can't run `from tabula import read_pdf`

If you've installed `tabula`, it will conflict with the namespace. You should install `tabula-py` after removing `tabula`.

```
pip uninstall tabula
pip install tabula-py
```

2.3 I got an empty DataFrame. How can I resolve it?

tabula-py and tabula-java don't support image-based PDFs. It should contain text-based table information.

Before tuning the tabula-py option, you have to check you set an appropriate `pages` option. By default, tabula-py extracts tables from the first page of your PDF, with `pages=1` argument. If you want to extract from all pages, you need to set `pages` option like `pages="all"` or `pages=[1, 2, 3]`. You might want to extract multiple tables from multiple pages, if so you need to set `multiple_tables=True` together.

Depending on the PDF's complexity, it might be difficult to extract table contents accurately.

Tuning points of tabula-py are limited:

- Set specific `area` for accurate table detection

- Try `lattice=True` option for the table having explicit lines. Or try `stream=True` option

To know the limitation of tabula-java, I highly recommend using tabula app, the GUI version of tabula-java.

tabula app can:

- specify the area with GUI
- show a preview of the extraction with lattice or stream mode
- export template that is reusable for tabula-py

Even if you can't extract tabula-py for those table contents which can be extracted tabula app appropriately, file an issue on GitHub.

2.4 The result is different from `tabula-java`. Or, `stream` option seems not to work appropriately

`tabula-py` set `guess` option `True` by default, for beginners. It is known to make a conflict between `stream` option. If you feel something strange with your result, please set `guess=False`.

2.5 Can I use option `xxx`?

Yes. You can use `options` argument as follows. The format is the same as CLI of tabula-java.

```
read_pdf(file_path, options="--columns 10.1,20.2,30.3")
```

2.6 How can I ignore useless area?

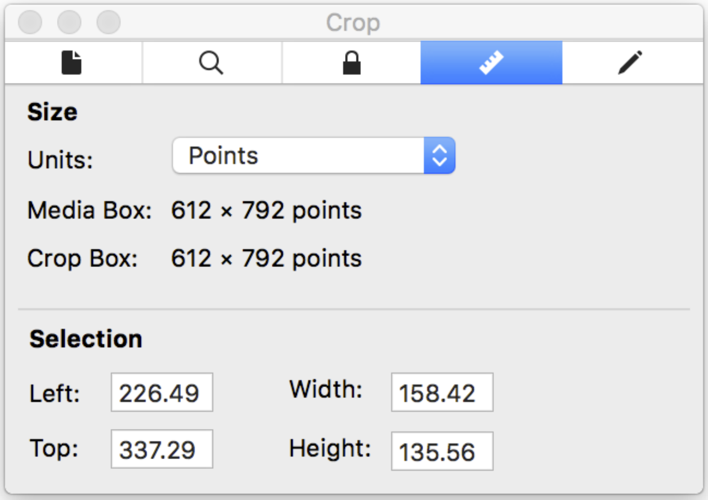
In short, you can extract with `area` and `spreadsheet` options.

```
In [4]: tabula.read_pdf('./table.pdf', spreadsheet=True, area=(337.29, 226.49, 472.85,
↪ 384.91))
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8
Out[4]:
  Unnamed: 0 Col2 Col3 Col4 Col5
0          A    B   12    R    G
1        NaN    R    T   23    H
2          B    B   33    R    A
3          C    T   99    E    M
4          D    I   12   34    M
5          E    I    I    W   90
6        NaN    1    2    W    h
7        NaN    4    3    E    H
8          F    E   E4    R    4
```

2.6.1 How to use `area` option

According to tabula-java wiki, there is an explanation of how to specify the area: https://github.com/tabulapdf/tabula-java/wiki/Using-the-command-line-tabula-extractor-tool#grab-coordinates-of-the-table-you-want

For example, using macOS's preview, I got area information of this PDF:



Size

Units: Points

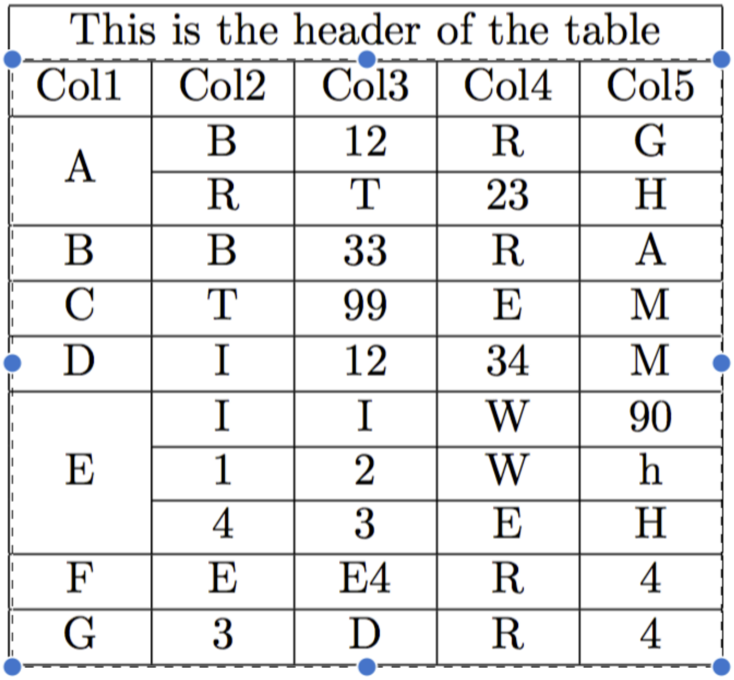
Media Box: 612 × 792 points

Crop Box: 612 × 792 points

Selection

Left: 226.49 Width: 158.42

Top: 337.29 Height: 135.56



This is the header of the table				
Col1	Col2	Col3	Col4	Col5
A	B	12	R	G
	R	T	23	H
B	B	33	R	A
C	T	99	E	M
D	I	12	34	M
E	I	I	W	90
	1	2	W	h
	4	3	E	H
F	E	E4	R	4
G	3	D	R	4

```
java -jar ./target/tabula-1.0.1-jar-with-dependencies.jar -p all -a $y1,$x1,$y2,$x2 -o $csvfile $filename
```

given

```
# Note the left, top, height, and width parameters and calculate the following:

y1 = top
x1 = left
```

(continues on next page)

(continued from previous page)

```
y2 = top + height
x2 = left + width
```

I confirmed with tabula-java:

```
java -jar ./tabula/tabula-1.0.1-jar-with-dependencies.jar -a "337.29,226.49,472.85,
↪384.91" table.pdf
```

Without -r(same as --spreadsheet) option, it does not work properly.

2.7 I faced ParserError: Error tokenizing data. C error. How can I extract multiple tables?

This error occurs when pandas tries to extract multiple tables with different column size at once. Use multiple_tables option, then you can avoid this error.

2.8 I want to prevent tabula-py from stealing focus on every call on my mac

Set java_options=["-Djava.awt.headless=true"]. kudos @jakekara

2.9 I got ? character with results on Windows. How can I avoid it?

If the encoding of PDF is UTF-8, you should set chcp 65001 on your terminal before launching a Python process.

```
chcp 65001
```

Then you can extract UTF-8 PDF with java_options="-Dfile.encoding=UTF8" option. This option will be added with encoding='utf-8' option, which is also set by default.

```
# This is an example for java_options is set explicitly
df = read_pdf(file_path, java_options="-Dfile.encoding=UTF8")
```

Replace 65001 and UTF-8 appropriately, if the file encoding isn't UTF-8.

2.10 I can't extract file/directory names with space on Windows

You should escape the file/directory name yourself.

2.11 I want to use a different tabula .jar file

You can specify the jar location via environment variable

```
export TABULA_JAR=".../tabula-x.y.z-jar-with-dependencies.jar"
```

2.12 I want to extract multiple tables from a document

You can use the following example code

```
df = read_pdf(file_path, multiple_tables=True)
```

The result will be a list of DataFrames. If you want separate tables across all pages in a document, use the `pages` argument.

2.13 Table cell contents sometimes overflow into the next row.

You can try using `lattice=True`, which will often work if there are lines separating cells in the table.

2.14 I got a warning/error message from PDFBox including `org.apache.pdfbox.pdmodel.`. Is it the cause of the empty dataframe?

No.

Sometimes, you might see a message like ‘‘ Jul 17, 2019 10:21:25 AM org.apache.pdfbox.pdmodel.font.PDType1Font WARNING: Using fallback font NimbusSanL-Regu for Univers. Nothing was parsed from this one.‘‘ This error message came from Apache PDFBox which is used under tabula-java, and this is caused by the PDF itself. Neither tabula-py nor tabula-java can’t handle the warning itself, except for the silent option that suppresses the warning.

2.15 I can’t figure out accurate extraction with tabula-py. Are there any similar Python libraries?

I know tabula-py has limitations depending on tabula-java. Sometimes your PDF is too complex to tabula-py. If you want to find plan B, there are similar packages as the following:

- https://github.com/jsvine/pdfplumber
- https://camelot-py.readthedocs.io/en/master/

CHAPTER 3

Contributing to tabula-py

Interested in helping out? I'd love to have your help!

You can help by:

- Reporting a bug.
- Adding or editing documentation.
- Contributing code via a Pull Request.
- Write a blog post or spread the word about `tabula-py` to people who might be able to benefit from using it.

3.1 Code formatting and testing

If you want to become a contributor, you can install dependency after cloning the repo as follows:

```
pip install -e .[dev, test]
pip install nox
```

For running tests and linter, run nox command.

```
nox .
```

3.2 Documentation

You can build document on your environment as follows:

```
pip install -e .[doc]
cd docs && make html
```

The documentation source is under `docs/` directory and the document is published on Read the Docs automatically.

CHAPTER 4

tabula

4.1 High level interfaces

4.1.1 tabula.io

4.1.2 tabula.util

4.2 Internal interfaces

4.2.1 tabula.template

4.2.2 tabula.file_util

CHAPTER 5

tabula.errors

CHAPTER 6

Indices and tables

- genindex
- modindex
- search