## Click to verify



Camelot is a Python library that can help you extract tables from PDFs. Extract tables from PDFs in just a few lines of code: Try it yourself in our interactive quickstart notebook. Or check out a simple example using this pdf. >>> tables = camelot.read pdf('foo.pdf') >>> tables.export('foo.csv', f='csv', compress=True) # json, excel, html, markdown, sqlite >>> tables[0].parsing report { 'accuracy': 99.02, 'whitespace': 12.24, 'order': 1, 'page': 1 } >>> tables[0].df # get a pandas DataFrame! Cycle Name KI (1/km) Distance (mi) Percent Fuel Savings Improved Speed Decreased Accel Eliminate Stops Decreased Idle 2012 2 3.30 1.3 5.9% 9.5% 29.2% 17.4% 2145 1 0.68 11.2 2.4% 0.1% 9.5% 2.7% 1.2% 4171 1 0.07 173.9 58.1% 1.6% 2.1% 0.5% Camelot also comes packaged with a command-line interface! Refer to the QuickStart Guide to quickly get started with Camelot, extract tables from PDFs and explore some basic options. Tip: Visit the parsers and their features. Note: Camelot only works with text-based PDFs and not scanned documents. (As Tabula explains, "If you can click and drag to select text in your table in a PDF viewer, then your PDF is text-based".) You can check out some frequently asked questions here. Why Camelot? Configurability: Camelot gives you control over the table extraction process with tweakable settings. Metrics: You can discard bad tables based on metrics like accuracy and whitespace, without having to manually look at each table. Output: Each table is extracted into a pandas DataFrame, which seamlessly integrates into ETL and data analysis workflows. You can also export tables to multiple formats, which include CSV, JSON, Excel, HTML, Markdown, and Sqlite. See comparison with similar libraries and tools. Installation Using conda The easiest way to install Camelot is with conda, which is a package manager and environment management system for the Anaconda distribution. conda install -c conda-forge camelot-py Using pip After installing the dependencies (tk and ghostscript), you can also just use pip to install -c conda-forge camelot-py [base]" From the source code After installing the dependencies, clone the repo using: git clone and install using pip: cd camelot pip install "." Documentation The documentation is available at . Wrappers camelot-sharp provides a PHP wrapper on Camelot. Related projects camelot-sharp provides a C sharp implementation of Camelot. Contributing The Contributor's Guide has detailed information about contributing issues, documentation, code, and tests. Versioning Camelot uses Semantic Versioning. For the available versions, see the tags on this repository was archived by the owner on Jan 6, 2025. It is now read-only. This repository was archived by the owner on Jan 6, 2025. It is now read-only. You can't perform that action at this time. Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot! Here's how you can extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library that can help you extract tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library tables from PDFs! Note: You can also check out Excalibur, the web interface to Camelot is a Python library tables from PDFs and You can check out the PDF used in this example here. >>> import camelot >>> tables = camelot.read\_pdf('foo.pdf') >>> tables.export('foo.csv', f='csv', compress=True) # json, excel, html, markdown, sqlite >>> tables[0] >>> tables[0].parsing\_report { 'accuracy': 99.02, 'whitespace': 12.24, 'order': 1, 'page': 1 } >>> tables[0].to\_csv('foo.csv') # to\_json, to\_excel, to\_html, to\_markdown, to\_sqlite >>> tables[0].df # get a pandas DataFrame! Cycle Name KI (1/km) Distance (mi) Percent Fuel Savings Improved Speed Decreased Accel Eliminate Stops Decreased Idle 2012 2 3.30 1.3 5.9% 9.5% 29.2% 17.4% 2145 1 0.68 11.2 2.4% 0.1% 9.5% 2.7% 4234 1 0.59 58.7 8.5% 1.3% 8.5% 3.3% 2032 2 0.17 57.8 21.7% 0.3% 2.7% 1.2% 4171 1 0.07 173.9 58.1% 1.6% 2.1% 0.5% Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command-line interface! Note: Camelot also comes packaged with a command with a is text-based".) You can check out some frequently asked questions here. Configurability: Camelot gives you control over the table extracted into a pandas DataFrame, which seamlessly integrates into ETL and data analysis workflows. You can also export tables to multiple formats, which include CSV, JSON, Excel, HTML, Markdown, and Sqlite. See comparison with similar libraries and tools. If Camelot has helped you, please consider supporting its development with a one-time or monthly donation on OpenCollective. The easiest way to install Camelot is with conda, which is a package manager and environment management system for the Anaconda distribution. \$ conda install -c conda-forge camelot-py[base]" After installing the dependencies, clone the repo using: \$ git clone and install Camelot using pip: \$ cd camelot spip install ".[base]" The documentation is available at . camelot-php provides a PHP wrapper on Camelot. The Contributor's Guide has detailed information about contributing issues, documentation, code, and tests. Camelot uses Semantic Versioning. For the available versions, see the tags on this repository. For the changelog, you can check out HISTORY.md. This project is licensed under the MIT License, see the LICENSE file for details. In this tutorial, you will learn how you can extract tables in PDF using camelot library in Python. Camelot is a Python library and a command-line tool that makes it easy for anyone to extract data tables trapped inside PDF files. This part of the documentation covers the steps to install Camelot. Using condaThe easiest way to install -c conda-forge camelot-pyUsing pipAfter installing the dependencies, which include Tkinter and ghostscript, you can simply use pip to install Camelot: pip install camelot-py[cv]For more information, check the official documentationThe PDF used in this tutorial can be downloaded from hereimport camelotTo extract the PDF# PDF file to extract tables fromfile = "foo.pdf"The PDF file called "foo.pdf" is a normal page that contains one tables in a PDF file.To print the number of tables extracted:# number of tables extracted;", tables.n)Output:It contains only one table.Printing this table as a Pandas DataFrame:# print the first table as Pandas DataFrameprint(tables[0].df)Output:Exporting the table to a CSV file:# export all tables all at once:# or export all in a ziptables.export("foo.csv", f="csv", compress=True)The tables can be exported to HTML format:# export to HTMLtables.export("foo.html", f="html")It can also be exported to JSON and Excel formats.Note: Camelot only works with text-based PDFs and not scanned documents When working on Windows, the easiest way to get up and running is through the Conceptive Python SDK. This SDK is a Python distribution targeted at the development and deployment of QT based applications. This all in one installation of Camelot with all its dependencies is available in the shop. First, make sure you have setup tools install to install Camelot, under Linux this would be done by typing: sudo easy\_install camelot Linux distributions often offer packages for various applications, including Camelot from source, you need to make sure all dependencies are installed and available in your PYTHONPATH. Dependencies In addition to PyQt 4.8 and Qt 4.8, Camelot needs these libraries : SQLAlchemy==0.8.0 Jinja2==2.6 chardet==2.1.1 xlwt==0.7.4 xlrd==0.9.0 Releases The source code of a release can be downloaded from the Bitbucket repository: hg clone Adapting PYTHONPATH You need to make sure Camelot and all its dependencies are in the PYTHONPATH before you start using it. To verify if you have Camelot >>> import camelot >>> import camelot. version >>> import sqlalchemy >>> print sqlalchemy. version >>> import PyQt4 None of them should raise an ImportError. linux-64 v1.0.0 osx-64 v1.0. forge/label/cf202003::camelot-py Description Extracting tabular data from PDFs has long been a challenging task. Traditional methods often involve manual copying and pasting, which is not only time-consuming but also prone to errors. Camelot, a Python library, offers a robust solution for this problem, particularly when dealing with tables in PDF documents. In this blog, we'll explore why Camelot is a preferred tool, provide a detailed code sample, discuss its pros, and highlight the industries using it. Additionally, we'll explain how Pysquad can assist in implementing Camelot for your projects. Why Camelot for your projects. Why camelot for your projects are provided to extract the industries using it. efficiently. Here are some reasons why Camelot stands out: Accuracy: Camelot uses a combination of rule-based and machine-learning techniques to accurately extract tables. Flexibility: It supports both stream and lattice methods, allowing it to handle a wide variety of table structures. Open Source: Being open source, it allows for customization and integration into various workflows. Ease of Use: With a simple API, Camelot makes it easy to extract tables with just a few lines of code. Camelot with Python Detailed Code sample to see how Camelot can be used to extract tables from a PDF document. InstallationFirst, you need to install Camelot. You can do this using pip:pip install camelot-py[cv]Basic UsageHere is a simple example of how to use Camelot to extract tables from the PDF tables = camelot.read pdf(file path) # Print the number of tables foundprint(f"Total tables from the PDF tables = camelot.read pdf(file path) # Print the number of tables from the PDF tables from the PDF tables = camelot.read pdf(file path) # Print the number of tables foundprint(f"Total tables from the PDF tables = camelot.read pdf(file path) # Print the number of tables from table content of the first tableprint(tables[0].df)Advanced UsageFor more control, you can specify parameters like flavor, table areas, and process background: import camelot # Use the lattice flavor to extract tablestables = camelot.read pdf(file path, flavor='lattice', pages='1-end') # Save the tables to CSVtables.export('tables.e accurately detect and extract tables reduces the need for manual intervention. Versatility: With support for both lattice and stream methods, Camelot can handle a wide range of table structures. Customizable: Being open source, it can be tailored to specific needs. Integration: Easy integration with other Python libraries and workflows, enhancing automation capabilities. Camelot is widely used across various industries where data extraction from PDFs is crucial: Finance: For extracting tables from financial reports, statements, and invoices. Healthcare: To extract data from medical records and research papers. Education: For extracting tables from academic papers and reports. Government: To process data from official documents and forms. Legal: For extracting information from contracts and case files. How Pysquad Can Assist in the Implementation Pysquad Can Assist in the Implementation from contracts and case files. your existing workflows.Customization: Tailoring Camelot to meet the specific requirements of your industry.Implementation: Set up Camelot and ensure it works seamlessly with your data processing pipelines.Training: Provide training to your team on how to use and customize Camelot for optimal results.Support: Offering ongoing support and maintenance to ensure smooth operation. References Camelot Documentation Camelot GitHub Repository Camelot offers a powerful and flexible solution for extracting tables from PDFs. Its high accuracy, ease of use, and open-source nature make it an excellent choice for various industries. With the assistance of Pysquad, you can seamlessly integrate Camelot into your workflows, enhancing your data extraction capabilities and improving efficiency. Whether you are in finance, healthcare, education, government, or legal sectors, Camelot can help you handle your data extraction needs with ease. a few lines of code: Try it yourself in our interactive guickstart notebook. Or check out a simple example using this pdf. >>> tables = camelot.read pdf('foo.csv', f='csv', compress=True) # ison, excel, html, markdown, sglite >>> tables[0] >>> tables[0] >>> tables[0] >>> tables = camelot.read pdf('foo.csv', f='csv', compress=True) # ison, excel, html, markdown, sglite >>> tables[0] >>> tabl 'whitespace': 12.24, 'order': 1, 'page': 1 } >>> tables[0].to\_csv('foo.csv') # to\_json, to\_excel, to\_html, to\_markdown, to\_sqlite >>> tables[0].df # get a pandas DataFrame! Cycle Name KI (1/km) Distance (mi) Percent Fuel Savings Improved Speed Decreased Accel Eliminate Stops Decreased Idle 2012 2 3.30 1.3 5.9% 9.5% 29.2% 17.4% 2145 1 0.68 11.2 2.4% 0.1% 9.5% 2.7% 4234 1 0.59 58.7 8.5% 1.3% 8.5% 3.3% 2032 2 0.17 57.8 21.7% 0.3% 2.7% 1.2% 4171 1 0.07 173.9 58.1% 1.6% 2.1% 0.5% Camelot also comes packaged with a command-line interface! Refer to the QuickStart Guide to quickly get started with Camelot, extract tables from PDFs and explore some basic options. Tip: Visit the parser-comparison-notebook to get an overview of all the packed parsers and their features. Note: Camelot only works with text-based PDFs and not scanned documents. (As Tabula explains, "If you can click and drag to select text in your table in a PDF viewer, then your PDF is text-based".) You can check out some frequently asked questions here. Configurability: Camelot gives you control over the table extracted into a pandas DataFrame, which seamlessly integrates into ETL and data analysis workflows. You can also export tables to multiple formats, which include CSV, JSON, Excel, HTML, Markdown, and Sqlite. See comparison with similar libraries and tools. The easiest way to install -c conda-forge camelot-py After installing the dependencies (tk and ghostscript), you can also just use pip to install "." The documentation is available at . camelot-php provides a PHP wrapper on Camelot. camelot-sharp provides a C sharp implementation of Camelot. The Contributor's Guide has detailed information about contributing issues, documentation, code, and tests. Camelot uses Semantic Versioning. For the available versions, see the tags on this repository. For the changelog, you can check out the releases page. This project is licensed under the MIT License, see the LICENSE file for details. We use GitHub issues to keep track of all issues. Please do not report bugs or issues in this blog's comments. Instead, post them on GitHub as an issue. Before submitting a comment with an issue, please use GitHub issues to keep track of all issues. Document Format) was born out of The Camelot Project to create "a universal way to communicate documents viewable on any modern printer. PDF was built on top of PostScript (a page description language), which had already solved this "view and print anywhere" problem. PDF encapsulates the components required to create a "view and print anywhere" document. These include characters, fonts, graphics and images. A PDF file defines instructions to place characters (and other components) at precise x, y coordinates relative to the bottom-left corner of the page. Words are simulated by placing some characters closer than others. Similarly, spaces are simulated then? You guessed it correctly — by placing words as they would appear in a spreadsheet. The PDF format has no internal representation of a table structure, which makes it difficult to extract tables for analysis. Sadly, a lot of open data is stored in PDFs, which was not designed for tabular data in the first place! Today, we're pleased to announce the release of Camelot, a Python library and command-line tool that makes it easy for anyone to extract data tables trapped inside PDF files! You can check out the documentation at Read the Docs and follow the development on GitHub.Installation is easy! After installing Python packages): pip install camelot-pyExtracting tables from a PDF using Camelot is very simple. Here's how you do it. (Here's the PDF used in the following example.)>>> import camelot.>>> tables = camelot.read pdf('foo.pdf') >>> tables.export('foo.csv', f='csv', compress=True) # json, excel, html >>> tables[0].parsing\_report { 'accuracy': 99.02, 'whitespace': 12.24, 'order': 1, 'page': 1 } >>> tables[0].to\_csv('foo.csv') # to\_json, to\_excel, to\_html >>> tables[0].df # get a pandas DataFrame!You can also check out the command-line interface.Camelot gives you complete control over table extraction by letting you tweak its settings.Bad tables can be discarded based on metrics like accuracy and whitespace, without ever having to manually look at each table.Each table is a pandas DataFrame, which seamlessly integrates into ETL and data analysis workflows. You can export tables to multiple formats, including CSV, JSON, Excel and HTML. Many people use open (Tabula, pdf-table-extract) and closed-source (smallpdf, pdftables) tools to extract tables from PDFs. But they either give a nice output or fail miserably. There is no in between. This is not helpful since everything in the real world, including PDF table extraction, is fuzzy. This leads to the created Camelot to offer users complete control over table extraction. If you can't get your desired output with the default settings, you can tweak them and get the job done!You can check out a comparison of Camelot's output with other open-source PDF table extraction libraries.We've often needed to extract data trapped inside PDFs. The first tool that we tried was Tabula, which has nice user and command-line interfaces, but it either worked perfectly or failed miserably. When it failed, it was difficult to tweak the settings — such as the image thresholding parameters, which influence table detection and can lead to a better output. We also didn't work, we tried pdftotext (an open-source command-line utility). pdftotext extracts text from a PDF while preserving the layout, using spaces. After getting the text, we had to write Python scripts with complicated regexes (regular expressions) to convert the text into tables. This wasn't scalable, since we had to change the regexes for each new table layout. We clearly needed a tweakable PDF table extraction tool, so we started developing one in December 2015. We started with the idea of giving the tool back to the community, which had given us so many open-source tools to work with. We knew that Tabula classifies PDF tables into two classes. It has two methods to extract these different classes: Lattice (to extract tables with clearly defined lines between cells) and Stream (to extract tables with spaces between cells). We named Camelot's table extraction flavors, Lattice and Stream, after Tabula's methods. Tabula uses a combination of scraping the vector elements and raster lines. image processing. After more exploration, we settled on morphological transformations, which gave the exact line segments. From here, representing the table trapped inside a PDF was straightforward. To get more information on how Lattice and Stream work in Camelot, check out the "How It Works" section of the documentation. We've battle tested Camelot by using it in a variety of projects, both for one-off and automated table extraction. For Atlan Grid, our curated data from 600+ sources (primarily PDF reports) for each of the 17 Sustainable Development Goals. For example, one of our sources for Goal 3 ("Good Health and Well-Being for People") is the National Family Health Survey (NFHS) report released by IIPS. To get data from these PDF sources, we created an internal web interface built on top of Camelot, where our data analysts could upload PDF reports and extract tables in their preferred format. We also set up an ETL workflow using Apache Airflow to track disease outbreaks in India. The workflow scrapes the Integrated Disease Surveillance Programme (IDSP) website for weekly PDFs of disease outbreak data, and then it extracts tables from the PDFs using Camelot, sends alerts to our team, and loads the data into a data warehouse. Camelot has some limitations. (We're developing solutions!) Here are a couple of them:When using Stream, tables aren't autodetected. Stream treats the whole page as a single table, which gives bad output when there are multiple tables on the page.Camelot only works with text-based PDFs and not scanned documents. (As Tabula explains, "If you can click-and-drag to select text in your table in a PDF viewer... then your PDF is text-based".)You can check out the GitHub repository for more information.You can help too — every contribution counts! Check out the Contributor's Guide for guidelines around contributor's Guide for guidelines around contributor's Guide for guidelines around contribution.You can help too — every contributor's Guide for guidelines around contribution counts! Check out the Contributor's Guide for guidelines around contributor's Guide for guideline and "good first issue".We urge organizations to release open data in a "data friendly" format like the CSV. But while tables are trapped inside PDF files, there's Camelot Note: This blog was updated on 2nd November 2018 after we learnt that Tabula uses a combination of scraping the vector elements and raster lines, and not the Hough Transform as mentioned in this blog. Photo by Jason Wong on Unsplash Camelot When working on Windows, the easiest way to get up and running is through the Conceptive Python SDK. This SDK is a Python distribution targeted at the development and deployment of Ot based applications. This all in one installation of Camelot with all its dependencies is available in the shop. First, make sure you have setup tools installed, Setup tools. If you are using a debian based distribution, you can type: sudo apt-get install camelot Linux distributions often offer packages for various applications, including Camelot and its dependencies : When installing Camelot from source, you need to make sure all dependencies are installed and available in your PYTHONPATH. Dependencies In addition to PyQt 4.8 and Qt 4.8, Camelot needs these libraries : SQLAlchemy==1.0.8 Jinja2==2.7.2 chardet==2.2.1 xlwt-future==0.8.0 xlrd==0.9.3 six==1.10.0 pycrypto==2.6.1 Releases The source code of a release can be downloaded from the Python Package Index and then extracted: tar xzvf Camelot-10.07.02.tar.gz Repository: hg clone Adapting PYTHONPATH You need to make sure Camelot and all its dependencies are in the PYTHONPATH before you start using it. To verify if you have Camelot installed and available in the PYTHONPATH, fire up a python interpreter: and issue these commands: >>> import camelot. version >>> import splatchemy. version >>> import splatchemy. should raise an ImportError. © Copyright 2009 - 2014, Conceptive Engineering. Last updated on Dec 19, 2016. Sphinx theme provided by Read the Docs