Tải dữ liệu pdf

[!WARNING]
Hỗ trợ Pdf là thử nghiệm và có thể thay đổi.

Bộ dữ liệu Pdf có các cột loại Pdf chứa các đối tượng thợ sửa ống nước pdf.

[!TIP]
Để làm việc với bộ dữ liệu pdf, bạn cần cài đặt gói pdfplumber. Kiểm tra
hướng dẫn cài đặt để tìm hiểu cách cài đặt nó.

Khi bạn tải tập dữ liệu pdf và gọi cột pdf, các tệp pdf sẽ được giải mã dưới dạng pdfplumber PDF:

```
>>> from datasets import load_dataset, Pdf

>>> dataset = load_dataset("path/to/pdf/folder", split="train")
>>> dataset[0]["pdf"]
<pdfplumber.pdf.PDF at 0x1075bc320>
```

[!WARNING]
Lập chỉ mục vào tập dữ liệu pdf bằng cách sử dụng chỉ mục hàng trước rồi đến cột pdf - dataset[0]["pdf"] - to avoid creating all the pdf objects in the dataset. Otherwise, this
có thể là một quá trình chậm và tốn thời gian nếu bạn có một tập dữ liệu lớn.

Để biết hướng dẫn về cách tải bất kỳ loại tập dữ liệu nào, hãy xem hướng dẫn tải chung.

Đọc trang

Truy cập các trang trực tiếp từ pdf bằng thuộc tính .pages.

Sau đó, bạn có thể sử dụng các hàm pdfplumber để đọc văn bản, bảng và hình ảnh, ví dụ:

```
>>> pdf = dataset[0]["pdf"]
>>> first_page = pdf.pages[0]
>>> first_page
<Page:1>
>>> first_page.extract_text()
```
Báo cáo kỹ thuật ghi chép
Phiên bản1.0
ChristophAuer MaksymLysak AhmedNassar MicheleDolfi NikolaosLivathinos
PanosVagenas CesarBerrospiRamis MatteoOmenetti FabianLindlbauer
KasperDinkla LokeshMishra YusikKim ShubhamGupta RafaelTeixeiradeLima
ValeryWeber LucasMorin IngmarMeijer ViktorKuropiatnyk PeterW.J.Staar
AI4KGroup,IBMNghiên cứu
Ru¨schlikon,Thụy Sĩ
Tóm tắt
Báo cáo kỹ thuật này giới thiệu Docling, một công cụ dễ sử dụng, khép kín của MIT-
gói nguồn mở được cấp phép để chuyển đổi tài liệu PDF.
None
```
>>> first_page.images
In [24]: first_page.images
Out[24]:
[{'x0': 256.5,
   'y0': 621.0,
   'x1': 355.49519999999995,
   'y1': 719.9952,
   'chiều rộng': 98,99519999999995,
   'chiều cao': 98,99519999999995,
   'tên': 'Im1',
   'stream': <PDFStream(44): raw=88980, {'Type': /'XObject', 'Subtype': /'Image', 'BitsPerComponent
   'srcsize': (1024, 1024),
   'mặt nạ hình ảnh': Không,
   'bit': 8,
   'colorspace': [/'DeviceRGB'],
   'mcid': Không,
   'thẻ': Không,
   'object_type': 'hình ảnh',
   'số_trang': 1,
   'trên cùng': 72.00480000000005,
   'dưới cùng': 171.0,
   'doctop': 72.00480000000005}]
```

```
>>> first_page.extract_tables()
[]
```

Bạn cũng có thể tải từng trang dưới dạng PIL.Image :

```
>>> import PIL.Image
>>> import io
>>> first_page.to_image()
<pdfplumber.display.PageImage at 0x107d68dd0>
>>> buffer = io.BytesIO()
>>> first_page.to_image().save(buffer)
>>> img = PIL.Image.open(buffer)
>>> img
<PIL.PngImagePlugin.PngImageFile image mode=P size=612x792>
```

Note that you can pass resolution= to .to_image() to render the image in higher resolution that the default (72 ppi).

Tệp cục bộ

You can load a dataset from the pdf path. Use the cast_column() function to accept a column đường dẫn tệp pdf và giải mã nó thành pdfplumber pdf với tính năng Pdf:

```
>>> from datasets import Dataset, Pdf

>>> dataset = Dataset.from_dict({"pdf": ["path/to/pdf_1", "path/to/pdf_2", ..., "path/to/pdf_n"]})
>>> dataset[0]["pdf"]
<pdfplumber.pdf.PDF at 0x1657d0280>
```

Nếu bạn chỉ muốn tải đường dẫn cơ bản tới tập dữ liệu pdf mà không giải mã đối tượng pdf, set decode=False in the Pdf feature:

```
>>> dataset = dataset.cast_column("pdf", Pdf(decode=False))
>>> dataset[0]["pdf"]
{'bytes': None,
  'path': 'path/to/pdf/folder/pdf0.pdf'}
```

Thư mục Pdf

Bạn cũng có thể tải tập dữ liệu bằng trình tạo tập dữ liệu PdfFolder mà không yêu cầu viết
trình tải dữ liệu tùy chỉnh. Điều này làm cho PdfFolder trở nên lý tưởng để tạo và tải nhanh các tập dữ liệu pd
với hàng nghìn tệp pdf cho các nhiệm vụ thị giác khác nhau. Cấu trúc tập dữ liệu pdf của bạn sẽ trông giống
cái này:

```
thư mục/train/sơ yếu lý lịch/0001.pdf
thư mục/train/sơ yếu lý lịch/0002.pdf
thư mục/train/sơ yếu lý lịch/0003.pdf

thư mục/tàu/hóa đơn/0001.pdf
folder/train/invoice/0002.pdf
folder/train/invoice/0003.pdf
```

If the dataset follows the PdfFolder structure, then you can load it directly with load_dataset():

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("username/dataset_name")
>>> # OR locally:
>>> dataset = load_dataset("/path/to/folder")
```

For local datasets, this is equivalent to passing pdffolder manually in load_dataset() and the
thư mục trong data_dir :

```
>>> dataset = load_dataset("pdffolder", data_dir="/path/to/folder")
```

Sau đó, bạn có thể truy cập các tệp pdf dưới dạng đối tượng pdfplumber.pdf.PDF:

```
>>> dataset["train"][0]
{"pdf": <pdfplumber.pdf.PDF at 0x161715e50>, "label": 0}

>>> dataset["train"][-1]
{"pdf": <pdfplumber.pdf.PDF at 0x16170bd90>, "label": 1}
```

To ignore the information in the metadata file, set drop_metadata=True in load_dataset():

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("username/dataset_with_metadata", drop_metadata=True)
```

Nếu bạn không có tệp siêu dữ liệu, PdfFolder sẽ tự động suy ra tên nhãn từ tệp
tên thư mục.
If you want to drop automatically created labels, set drop_labels=True .
Trong trường hợp này, tập dữ liệu của bạn sẽ chỉ chứa cột pdf:

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("username/dataset_without_metadata", drop_labels=True)
```

Cuối cùng, đối số bộ lọc cho phép bạn chỉ tải một tập hợp con của tập dữ liệu, dựa trên một điều kiện
trên nhãn hoặc siêu dữ liệu. Điều này đặc biệt hữu ích nếu siêu dữ liệu ở định dạng Parquet,
vì định dạng này cho phép lọc nhanh. Bạn cũng nên sử dụng đối số này với
streaming=True , because by default the dataset is fully downloaded before filtering.

```
>>> filters = [("label", "=", 0)]
>>> dataset = load_dataset("username/dataset_name", streaming=True, filters=filters)
```

[!TIP]
Để biết thêm thông tin về cách tạo tập dữ liệu PdfFolder của riêng bạn, hãy xem
Tạo hướng dẫn tập dữ liệu pdf.

giải mã pdf

Theo mặc định, các tệp pdf được giải mã tuần tự dưới dạng tệp PDF pdfplumber khi bạn lặp lại trên một tập
Nó giải mã tuần tự siêu dữ liệu của các tệp pdf và không đọc các trang pdf cho đến khi bạn
truy cập chúng.

Tuy nhiên, có thể tăng tốc đáng kể tập dữ liệu bằng cách sử dụng giải mã đa luồng:

```
>>> import os
>>> num_threads = num_threads = min(32, (os.cpu_count() or 1) + 4)
>>> dataset = dataset.decode(num_threads=num_threads)
>>> for example in dataset:# up to 20 times faster !
None
```

Bạn có thể kích hoạt đa luồng bằng cách sử dụng num_threads. Điều này đặc biệt hữu ích để tăng tốc truyền dữ liệu từ xa.
However it can be slower than num_threads=0 for local data on fast disks.

Nếu bạn không quan tâm đến các tài liệu được giải mã dưới dạng PDF của thợ sửa ống nước pdf và muốn thay vào đó, hãy truy cập vào đường dẫn/byte, bạn có thể tắt giải mã:

```
>>> dataset = dataset.decode(False)
```

Note: IterableDataset.decode() is only available for streaming datasets at the moment.