■ Modules ■ Data connection ■ Document loaders ■ How-to ■ PDF

PDF

> Portable Document Format (PDF), standardized as ISO 32000, is a file format developed by Adobe in 1992 to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems.

This covers how to load `PDF` documents into the Document format that we use downstream.

Using PyPDF

Load PDF using `pypdf` into array of documents, where each document contains the page content and metadata with `page` number.

```
pip install pypdf
```

```
from langchain.document_loaders import PyPDFLoader

loader = PyPDFLoader("example_data/layout-parser-paper.pdf")
pages = loader.load_and_split()
```

```
pages[0]
```

```
    Document(page_content='LayoutParser : A Uni\x0ced Toolkit for Deep\nLearning Based Document Image
Analysis\nZejiang Shen1( \x00), Ruochen Zhang2, Melissa Dell3, Benjamin Charles Germain\nLee4, Jacob
Carlson3, and Weining Li5\n1Allen Institute for AI\nshannons@allenai.org\n2Brown University\nruochen
zhang@brown.edu\n3Harvard University\nfmelissadell,jacob carlson g@fas.harvard.edu\n4University of
Washington\nbcgl@cs.washington.edu\n5University of Waterloo\nw422li@uwaterloo.ca\nAbstract. Recent advances
in document image analysis (DIA) have been\nprimarily driven by the application of neural networks. Ideally,
research\noutcomes could be easily deployed in production and extended for further\ninvestigation. However,
various factors like loosely organized codebases\nand sophisticated model con\x0cgurations complicate the
easy reuse of im-\nportant innovations by a wide audience. Though there have been on-going\ne\x0borts to
improve reusability and simplify deep learning (DL) model\ndevelopment in disciplines like natural language
processing and computer\nvision, none of them are optimized for challenges in the domain of DIA.\nThis
represents a major gap in the existing toolkit, as DIA is central to\nacademic research across a wide range
of disciplines in the social sciences\nand humanities. This paper introduces LayoutParser , an open-
source\nlibrary for streamlining the usage of DL in DIA research and applica-\ntions. The core LayoutParser
library comes with a set of simple and\nintuitive interfaces for applying and customizing DL models for
layout de-\ntection, character recognition, and many other document processing tasks.\nTo promote
extensibility, LayoutParser also incorporates a community\nplatform for sharing both pre-trained models and
full document digiti-\nzation pipelines. We demonstrate that LayoutParser is helpful for both\nlightweight
and large-scale digitization pipelines in real-word use cases.\nThe library is publicly available at
https://layout-parser.github.io .\nKeywords: Document Image Analysis ·Deep Learning ·Layout
Analysis\n·Character Recognition ·Open Source library ·Toolkit.\n1 Introduction\nDeep Learning(DL)-based
approaches are the state-of-the-art for a wide range of\ndocument image analysis (DIA) tasks including
document image classi\x0ccation [ 11,arXiv:2103.15348v2  [cs.CV]  21 Jun 2021', metadata={'source':
'example_data/layout-parser-paper.pdf', 'page': 0})
```

An advantage of this approach is that documents can be retrieved with page numbers.

We want to use `OpenAIEmbeddings` so we have to get the OpenAI API Key.

```
import os
import getpass
```

```
os.environ['OPENAI_API_KEY'] = getpass.getpass('OpenAI API Key:')
```

```
    OpenAI API Key: ········
```

```
from langchain.vectorstores import FAISS
from langchain.embeddings.openai import OpenAIEmbeddings

faiss_index = FAISS.from_documents(pages, OpenAIEmbeddings())
docs = faiss_index.similarity_search("How will the community be engaged?", k=2)
for doc in docs:
    print(str(doc.metadata["page"]) + ":", doc.page_content[:300])
```

```
    9: 10 Z. Shen et al.
    Fig. 4: Illustration of (a) the original historical Japanese document with layout
    detection results and (b) a recreated version of the document image that achieves
    much better character recognition recall. The reorganization algorithm rearranges
    the tokens based on the their detect
    3: 4 Z. Shen et al.
    Efficient Data AnnotationC u s t o m i z e d  M o d e l  T r a i n i n gModel Cust omizationDI A Model HubDI A Pipeline SharingCommunity PlatformLa y out Detection ModelsDocument Images
    T h e  C o r e  L a y o u t P a r s e r  L i b r a r yOCR ModuleSt or age & VisualizationLa y ou
```

Using MathPix

Inspired by Daniel Gross's https://gist.github.com/danielgross/3ab4104e14faccc12b49200843adab21

```
from langchain.document_loaders import MathpixPDFLoader
```

```
loader = MathpixPDFLoader("example_data/layout-parser-paper.pdf")
```

```
data = loader.load()
```

Using Unstructured

```
from langchain.document_loaders import UnstructuredPDFLoader
```

```
loader = UnstructuredPDFLoader("example_data/layout-parser-paper.pdf")
```

```
data = loader.load()
```

Retain Elements

Under the hood, Unstructured creates different "elements" for different chunks of text. By default we combine those together, but you can easily keep that separation by specifying `mode="elements"`.

```
loader = UnstructuredPDFLoader("example_data/layout-parser-paper.pdf", mode="elements")
```

```
data = loader.load()
```

```
data[0]
```

```
    Document(page_content='LayoutParser: A Unified Toolkit for Deep\nLearning Based Document Image
Analysis\nZejiang Shen1 (�), Ruochen Zhang2, Melissa Dell3, Benjamin Charles Germain\nLee4, Jacob Carlson3,
and Weining Li5\n1 Allen Institute for AI\nshannons@allenai.org\n2 Brown University\nruochen
zhang@brown.edu\n3 Harvard University\n{melissadell,jacob carlson}@fas.harvard.edu\n4 University of
Washington\nbcgl@cs.washington.edu\n5 University of Waterloo\nw422li@uwaterloo.ca\nAbstract. Recent advances
in document image analysis (DIA) have been\nprimarily driven by the application of neural networks. Ideally,
research\noutcomes could be easily deployed in production and extended for further\ninvestigation. However,
various factors like loosely organized codebases\nand sophisticated model configurations complicate the easy
reuse of im-\nportant innovations by a wide audience. Though there have been on-going\nefforts to improve
reusability and simplify deep learning (DL) model\ndevelopment in disciplines like natural language
processing and computer\nvision, none of them are optimized for challenges in the domain of DIA.\nThis
represents a major gap in the existing toolkit, as DIA is central to\nacademic research across a wide range
of disciplines in the social sciences\nand humanities. This paper introduces LayoutParser, an open-
source\nlibrary for streamlining the usage of DL in DIA research and applica-\ntions. The core LayoutParser
library comes with a set of simple and\nintuitive interfaces for applying and customizing DL models for
layout de-\ntection, character recognition, and many other document processing tasks.\nTo promote
extensibility, LayoutParser also incorporates a community\nplatform for sharing both pre-trained models and
full document digiti-\nzation pipelines. We demonstrate that LayoutParser is helpful for both\nlightweight
and large-scale digitization pipelines in real-word use cases.\nThe library is publicly available at
https://layout-parser.github.io.\nKeywords: Document Image Analysis · Deep Learning · Layout Analysis\n·
Character Recognition · Open Source library · Toolkit.\n1\nIntroduction\nDeep Learning(DL)-based approaches
are the state-of-the-art for a wide range of\ndocument image analysis (DIA) tasks including document image
classification [11,\narXiv:2103.15348v2  [cs.CV]  21 Jun 2021\n', lookup_str='', metadata={'file_path':
'example_data/layout-parser-paper.pdf', 'page_number': 1, 'total_pages': 16, 'format': 'PDF 1.5', 'title':
'', 'author': '', 'subject': '', 'keywords': '', 'creator': 'LaTeX with hyperref', 'producer': 'pdfTeX-
1.40.21', 'creationDate': 'D:20210622012710Z', 'modDate': 'D:20210622012710Z', 'trapped': '', 'encryption':
None}, lookup_index=0)
```

Fetching remote PDFs using Unstructured

This covers how to load online pdfs into a document format that we can use downstream. This can be used for various online pdf sites such as https://open.umn.edu/opentextbooks/textbooks/ and https://arxiv.org/archive/

Note: all other pdf loaders can also be used to fetch remote PDFs, but `OnlinePDFLoader` is a legacy function, and works specifically with `UnstructuredPDFLoader`.

```
from langchain.document_loaders import OnlinePDFLoader
```

```
loader = OnlinePDFLoader("https://arxiv.org/pdf/2302.03803.pdf")
```

```
data = loader.load()
```

```
print(data)
```

```
    [Document(page_content='A WEAK ( k, k ) -LEFSCHETZ THEOREM FOR PROJECTIVE TORIC ORBIFOLDS\n\nWilliam D.
Montoya\n\nInstituto de Matem´atica, Estat´ıstica e Computa¸c˜ao Cient´ıfica,\n\nIn [3] we proved that, under
suitable conditions, on a very general codimension s quasi- smooth intersection subvariety X in a projective
toric orbifold P d Σ with d + s = 2 ( k + 1 ) the Hodge conjecture holds, that is, every ( p, p ) -cohomology
class, under the Poincar´e duality is a rational linear combination of fundamental classes of algebraic
subvarieties of X . The proof of the above-mentioned result relies, for p ≠ d + 1 − s , on a
```

```
Lefschetz\n\nKeywords: (1,1)- Lefschetz theorem, Hodge conjecture, toric varieties, complete intersection
Email: wmontoya@ime.unicamp.br\n\ntheorem ([7]) and the Hard Lefschetz theorem for projective orbifolds
([11]). When p = d + 1 − s the proof relies on the Cayley trick, a trick which associates to X a quasi-smooth
hypersurface Y in a projective vector bundle, and the Cayley Proposition (4.3) which gives an isomorphism of
some primitive cohomologies (4.2) of X and Y . The Cayley trick, following the philosophy of Mavlyutov in
[7], reduces results known for quasi-smooth hypersurfaces to quasi-smooth intersection subvarieties. The idea
in this paper goes the other way around, we translate some results for quasi-smooth intersection subvarieties
to\n\nAcknowledgement. I thank Prof. Ugo Bruzzo and Tiago Fonseca for useful discus- sions. I also
acknowledge support from FAPESP postdoctoral grant No. 2019/23499-7.\n\nLet M be a free abelian group of rank
d , let N = Hom ( M, Z ) , and N R = N ⊗ Z R .\n\nif there exist k linearly independent primitive elements
e\n\n, . . . , e k ∈ N such that σ = { µ\n\ne\n\n+ ⋯ + µ k e k } . • The generators e i are integral if for
every i and any nonnegative rational number µ the product µe i is in N only if µ is an integer. • Given two
rational simplicial cones σ , σ ′ one says that σ ′ is a face of σ ( σ ′ < σ ) if the set of integral
generators of σ ′ is a subset of the set of integral generators of σ . • A finite set Σ = { σ\n\n, . . . , σ t
} of rational simplicial cones is called a rational simplicial complete d -dimensional fan if:\n\nall faces
of cones in Σ are in Σ ;\n\nif σ, σ ′ ∈ Σ then σ ∩ σ ′ < σ and σ ∩ σ ′ < σ ′ ;\n\nN R = σ\n\n∪ · · · ∪ σ t
.\n\nA rational simplicial complete d -dimensional fan Σ defines a d -dimensional toric variety P d Σ having
only orbifold singularities which we assume to be projective. Moreover, T : = N ⊗ Z C ∗ ≃ ( C ∗ ) d is the
torus action on P d Σ . We denote by Σ ( i ) the i -dimensional cones\n\nFor a cone σ ∈ Σ, ˆ σ is the set of
1-dimensional cone in Σ that are not contained in σ\n\nand x ˆ σ : = ∏ ρ ∈ ˆ σ x ρ is the associated monomial
in S .\n\nDefinition 2.2. The irrelevant ideal of P d Σ is the monomial ideal B Σ : =< x ˆ σ | σ ∈ Σ > and the
zero locus Z ( Σ ) : = V ( B Σ ) in the affine space A d : = Spec ( S ) is the irrelevant locus.\n\nProposition
2.3 (Theorem 5.1.11 [5]) . The toric variety P d Σ is a categorical quotient A d \ Z ( Σ ) by the group Hom (
Cl ( Σ ) , C ∗ ) and the group action is induced by the Cl ( Σ ) - grading of S .\n\nNow we give a brief
introduction to complex orbifolds and we mention the needed theorems for the next section. Namely: de Rham
theorem and Dolbeault theorem for complex orbifolds.\n\nDefinition 2.4. A complex orbifold of complex
dimension d is a singular complex space whose singularities are locally isomorphic to quotient singularities
C d / G , for finite sub- groups G ⊂ Gl ( d, C ) .\n\nDefinition 2.5. A differential form on a complex orbifold
Z is defined locally at z ∈ Z as a G -invariant differential form on C d where G ⊂ Gl ( d, C ) and Z is locally
isomorphic to d\n\nRoughly speaking the local geometry of orbifolds reduces to local G -invariant
geometry.\n\nWe have a complex of differential forms ( A • ( Z ) , d ) and a double complex ( A • , • ( Z ) ,
∂, ¯ ∂ ) of bigraded differential forms which define the de Rham and the Dolbeault cohomology groups (for a
fixed p ∈ N ) respectively:\n\n(1,1)-Lefschetz theorem for projective toric orbifolds\n\nDefinition 3.1. A
```

```
subvariety X ⊂ P d Σ is quasi-smooth if V ( I X ) ⊂ A #Σ ( 1 ) is smooth outside\n\nExample 3.2 . Quasi-
smooth hypersurfaces or more generally quasi-smooth intersection sub-\n\nExample 3.2 . Quasi-smooth
hypersurfaces or more generally quasi-smooth intersection sub- varieties are quasi-smooth subvarieties (see
[2] or [7] for more details).\n\nRemark 3.3 . Quasi-smooth subvarieties are suborbifolds of P d Σ in the
sense of Satake in [8]. Intuitively speaking they are subvarieties whose only singularities come from the
ambient\n\nProof. From the exponential short exact sequence\n\nwe have a long exact sequence in
cohomology\n\nH 1 (O ∗ X ) → H 2 ( X, Z ) → H 2 (O X ) ≃ H 0 , 2 ( X )\n\nwhere the last isomorphisms is due
to Steenbrink in [9]. Now, it is enough to prove the commutativity of the next diagram\n\nwhere the last
isomorphisms is due to Steenbrink in [9]. Now,\n\nH 2 ( X, Z ) / / H 2 ( X, O X ) ≃ Dolbeault H 2 ( X, C )
deRham ≃ H 2 dR ( X, C ) / / H 0 , 2 ¯ ∂ ( X )\n\nof the proof follows as the ( 1 , 1 ) -Lefschetz theorem in
[6].\n\nRemark 3.5 . For k = 1 and P d Σ as the projective space, we recover the classical ( 1 , 1 ) -
Lefschetz theorem.\n\nBy the Hard Lefschetz Theorem for projective orbifolds (see [11] for details) we\n\nBy
the Hard Lefschetz Theorem for projective orbifolds (see [11] for details) we get an isomorphism of
cohomologies :\n\ngiven by the Lefschetz morphism and since it is a morphism of Hodge structures, we
have:\n\nH 1 , 1 ( X, Q ) ≃ H dim X − 1 , dim X − 1 ( X, Q )\n\nCorollary 3.6. If the dimension of X is 1 , 2
or 3 . The Hodge conjecture holds on X\n\nProof. If the dim C X = 1 the result is clear by the Hard Lefschetz
theorem for projective orbifolds. The dimension 2 and 3 cases are covered by Theorem 3.5 and the Hard
Lefschetz.\n\nCayley trick and Cayley proposition\n\nThe Cayley trick is a way to associate to a quasi-smooth
intersection subvariety a quasi- smooth hypersurface. Let L 1 , . . . , L s be line bundles on P d Σ and let
π ∶ P ( E ) → P d Σ be the projective space bundle associated to the vector bundle E = L 1 ⊕ ⋯ ⊕ L s . It
is known that P ( E ) is a ( d + s − 1 ) -dimensional simplicial toric variety whose fan depends on the
degrees of the line bundles and the fan Σ. Furthermore, if the Cox ring, without considering the grading, of
P d Σ is C [ x 1 , . . . , x m ] then the Cox ring of P ( E ) is\n\nMoreover for X a quasi-smooth
intersection subvariety cut off by f 1 , . . . , f s with deg ( f i ) = [ L i ] we relate the hypersurface Y
cut off by F = y 1 f 1 + · · · + y s f s which turns out to be quasi-smooth. For more details see Section 2 in
[7].\n\nWe will denote P ( E ) as P d + s − 1 Σ ,X to keep track of its relation with X and P d Σ .\n\nThe
following is a key remark.\n\nRemark 4.1 . There is a morphism ι ∶ X → Y ⊂ P d + s − 1 Σ ,X . Moreover every
point z ∶ = ( x, y ) ∈ Y with y ≠ 0 has a preimage. Hence for any subvariety W = V ( I W ) ⊂ X ⊂ P d Σ there
exists W ′ ⊂ Y ⊂ P d + s − 1 Σ ,X such that π ( W ′ ) = W , i.e., W ′ = { z = ( x, y ) ∣ x ∈ W } .\n\nFor X ⊂
P d Σ a quasi-smooth intersection variety the morphism in cohomology induced by the inclusion i ∗ ∶ H d − s (
P d Σ , C ) → H d − s ( X, C ) is injective by Proposition 1.4 in [7].\n\nDefinition 4.2. The primitive
cohomology of H d − s prim ( X ) is the quotient H d − s ( X, C )/ i ∗ ( H d − s ( P d Σ , C )) and H d − s
prim ( X, Q ) with rational coefficients.\n\nH d − s ( P d Σ , C ) and H d − s ( X, C ) have pure Hodge
```

```
structures, and the morphism i ∗ is com- patible with them, so that H d − s prim ( X ) gets a pure Hodge
structure.\n\nThe next Proposition is the Cayley proposition.\n\nProposition 4.3. [Proposition 2.3 in [3] ]
Let X = X 1 ∩· · ·∩ X s be a quasi-smooth intersec- tion subvariety in P d Σ cut off by homogeneous polynomials
f 1 . . . f s . Then for p ≠ d + s − 1 2 , d + s − 3 2\n\nRemark 4.5 . The above isomorphisms are also true
with rational coefficients since H • ( X, C ) = H • ( X, Q ) ⊗ Q C . See the beginning of Section 7.1 in [10]
for more details.\n\nTheorem 5.1. Let Y = { F = y 1 f 1 + ··· + y k f k = 0 } ⊂ P 2 k + 1 Σ ,X be the quasi-
smooth hypersurface associated to the quasi-smooth intersection surface X = X f 1 ∩ · · · ∩ X f k ⊂ P k + 2 Σ
. Then on Y the Hodge conjecture holds.\n\nthe Hodge conjecture holds.\n\nProof. If H k,k prim ( X, Q ) = 0
we are done. So let us assume H k,k prim ( X, Q ) ≠ 0. By the Cayley proposition H k,k prim ( Y, Q ) ≃ H 1 ,
1 prim ( X, Q ) and by the ( 1 , 1 ) -Lefschetz theorem for projective\n\ntoric orbifolds there is a non-zero
algebraic basis λ C 1 , . . . , λ C n with rational coefficients of H 1 , 1 prim ( X, Q ) , that is, there are
n : = h 1 , 1 prim ( X, Q ) algebraic curves C 1 , . . . , C n in X such that under the Poincar ́e duality the
class in homology [ C i ] goes to λ C i , [ C i ] ↦ λ C i . Recall that the Cox ring of P k + 2 is contained
in the Cox ring of P 2 k + 1 Σ ,X without considering the grading. Considering the grading we have that if α
∈ Cl ( P k + 2 Σ ) then ( α, 0 ) ∈ Cl ( P 2 k + 1 Σ ,X ) . So the polynomials defining C i ⊂ P k + 2 Σ can be
interpreted in P 2 k + 1 X, Σ but with different degree. Moreover, by Remark 4.1 each C i is contained in Y =
{ F = y 1 f 1 + ··· + y k f k = 0 } and\n\nfurthermore it has codimension k .\n\nClaim: { C i } ni = 1 is a
basis of prim ( ) . It is enough to prove that λ C i is different from zero in H k,k prim ( Y, Q ) or
equivalently that the cohomology classes { λ C i } ni = 1 do not come from the ambient space. By
contradiction, let us assume that there exists a j and C ⊂ P 2 k + 1 Σ ,X such that λ C ∈ H k,k ( P 2 k + 1 Σ
,X , Q ) with i ∗ ( λ C ) = λ C j or in terms of homology there exists a ( k + 2 ) -dimensional algebraic
subvariety V ⊂ P 2 k + 1 Σ ,X such that V ∩ Y = C j so they are equal as a homology class of P 2 k + 1 Σ ,X
,i.e., [ V ∩ Y ] = [ C j ] . It is easy to check that π ( V ) ∩ X = C j as a subvariety of P k + 2 Σ where π
: ( x, y ) ↦ x . Hence [ π ( V ) ∩ X ] = [ C j ] which is equivalent to say that λ C j comes from P k + 2 Σ
which contradicts the choice of [ C j ] .\n\nRemark 5.2 . Into the proof of the previous theorem, the key
fact was that on X the Hodge conjecture holds and we translate it to Y by contradiction. So, using an
analogous argument we have:\n\nargument we have:\n\nProposition 5.3. Let Y = { F = y 1 f s +···+ y s f s = 0 }
⊂ P 2 k + 1 Σ ,X be the quasi-smooth hypersurface associated to a quasi-smooth intersection subvariety X = X
f 1 ∩ · · · ∩ X f s ⊂ P d Σ such that d + s = 2 ( k + 1 ) . If the Hodge conjecture holds on X then it holds
as well on Y .\n\nCorollary 5.4. If the dimension of Y is 2 s − 1 , 2 s or 2 s + 1 then the Hodge conjecture
holds on Y .\n\nProof. By Proposition 5.3 and Corollary 3.6.\n\n[\n\n] Angella, D. Cohomologies of certain
orbifolds. Journal of Geometry and Physics\n\n(\n\n),\n\n−\n\n[\n\n] Batyrev, V. V., and Cox, D. A. On the
Hodge structure of projective hypersur- faces in toric varieties. Duke Mathematical
```

```
Journal\n\n,\n\n(Aug\n\n). [\n\n] Bruzzo, U., and Montoya, W. On the Hodge conjecture for quasi-smooth in-
tersections in toric varieties. S˜ao Paulo J. Math. Sci. Special Section: Geometry in Algebra and Algebra in
Geometry (\n\n). [\n\n] Caramello Jr, F. C. Introduction to orbifolds. a\n\niv:\n\nv\n\n(\n\n). [\n\n] Cox,
D., Little, J., and Schenck, H. Toric varieties, vol.\n\nAmerican Math- ematical Soc.,\n\n[\n\n] Griffiths,
P., and Harris, J. Principles of Algebraic Geometry. John Wiley & Sons, Ltd,\n\n[\n\n] Mavlyutov, A. R.
Cohomology of complete intersections in toric varieties. Pub- lished in Pacific J. of
Math.\n\nNo.\n\n(\n\n),\n\n–\n\n[\n\n] Satake, I. On a Generalization of the Notion of Manifold. Proceedings
of the National Academy of Sciences of the United States of America\n\n,\n\n(\n\n),\n\n–\n\n[\n\n]
Steenbrink, J. H. M. Intersection form for quasi-homogeneous singularities. Com- positio
Mathematica\n\n,\n\n(\n\n),\n\n–\n\n[\n\n] Voisin, C. Hodge Theory and Complex Algebraic Geometry I,
vol.\n\nof Cambridge Studies in Advanced Mathematics . Cambridge University Press,\n\n[\n\n] Wang, Z. Z., and
Zaffran, D. A remark on the Hard Lefschetz theorem for K¨ahler orbifolds. Proceedings of the American
Mathematical Society\n\n,\n\n(Aug\n\n).\n\n[2] Batyrev, V. V., and Cox, D. A. On the Hodge structure of
projective hypersur- faces in toric varieties. Duke Mathematical Journal 75, 2 (Aug 1994).\n\n[\n\n] Bruzzo,
U., and Montoya, W. On the Hodge conjecture for quasi-smooth in- tersections in toric varieties. S˜ao Paulo
J. Math. Sci. Special Section: Geometry in Algebra and Algebra in Geometry (\n\n).\n\n[3] Bruzzo, U., and
Montoya, W. On the Hodge conjecture for quasi-smooth in- tersections in toric varieties. S˜ao Paulo J. Math.
Sci. Special Section: Geometry in Algebra and Algebra in Geometry (2021).\n\nA. R. Cohomology of complete
intersections in toric varieties. Pub-', lookup_str='', metadata={'source':
'/var/folders/ph/hhm7_zyx4l13k3v8z02dwp1w0000gn/T/tmpgq0ckaja/online_file.pdf'}, lookup_index=0)]
```

Using PyPDFium2

```
from langchain.document_loaders import PyPDFium2Loader
```

```
loader = PyPDFium2Loader("example_data/layout-parser-paper.pdf")
```

```
data = loader.load()
```

Using PDFMiner

```
from langchain.document_loaders import PDFMinerLoader
```

```
loader = PDFMinerLoader("example_data/layout-parser-paper.pdf")
```

```
data = loader.load()
```

Using PDFMiner to generate HTML text

This can be helpful for chunking texts semantically into sections as the output html content can be parsed via `BeautifulSoup` to get more structured and rich information about font size, page numbers, pdf headers/footers, etc.

```
from langchain.document_loaders import PDFMinerPDFasHTMLLoader
```

```
loader = PDFMinerPDFasHTMLLoader("example_data/layout-parser-paper.pdf")
```

```
data = loader.load()[0]   # entire pdf is loaded as a single Document
```

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(data.page_content,'html.parser')
content = soup.find_all('div')
```

```
import re
cur_fs = None
cur_text = ''
snippets = []   # first collect all snippets that have the same font size
for c in content:
    sp = c.find('span')
    if not sp:
        continue
    st = sp.get('style')
    if not st:
        continue
    fs = re.findall('font-size:(\d+)px',st)
    if not fs:
        continue
    fs = int(fs[0])
    if not cur_fs:
        cur_fs = fs
    if fs == cur_fs:
        cur_text += c.text
    else:
        snippets.append((cur_text,cur_fs))
        cur_fs = fs
        cur_text = c.text
snippets.append((cur_text,cur_fs))
# Note: The above logic is very straightforward. One can also add more strategies such as removing duplicate
snippets (as
# headers/footers in a PDF appear on multiple pages so if we find duplicatess safe to assume that it is
redundant info)
```

```
from langchain.docstore.document import Document
cur_idx = -1
semantic_snippets = []
# Assumption: headings have higher font size than their respective content
for s in snippets:
    # if current snippet's font size > previous section's heading => it is a new heading
    if not semantic_snippets or s[1] > semantic_snippets[cur_idx].metadata['heading_font']:
        metadata={'heading':s[0], 'content_font': 0, 'heading_font': s[1]}
        metadata.update(data.metadata)
        semantic_snippets.append(Document(page_content='',metadata=metadata))
        cur_idx += 1
        continue

    # if current snippet's font size <= previous section's content => content belongs to the same section
(one can also create
    # a tree like structure for sub sections if needed but that may require some more thinking and may be
data specific)
    if not semantic_snippets[cur_idx].metadata['content_font'] or s[1] <=
semantic_snippets[cur_idx].metadata['content_font']:
        semantic_snippets[cur_idx].page_content += s[0]
        semantic_snippets[cur_idx].metadata['content_font'] = max(s[1],
semantic_snippets[cur_idx].metadata['content_font'])
        continue

    # if current snippet's font size > previous section's content but less tha previous section's heading
than also make a new
    # section (e.g. title of a pdf will have the highest font size but we don't want it to subsume all
sections)
    metadata={'heading':s[0], 'content_font': 0, 'heading_font': s[1]}
    metadata.update(data.metadata)
    semantic_snippets.append(Document(page_content='',metadata=metadata))
    cur_idx += 1
```

```
semantic_snippets[4]
```

```
    Document(page_content='Recently, various DL models and datasets have been developed for layout
analysis\ntasks. The dhSegment [22] utilizes fully convolutional networks [20] for segmen-\ntation tasks on
historical documents. Object detection-based methods like Faster\nR-CNN [28] and Mask R-CNN [12] are used for
identifying document elements [38]\nand detecting tables [30, 26]. Most recently, Graph Neural Networks [29]
have also\nbeen used in table detection [27]. However, these models are usually implemented\nindividually and
there is no unified framework to load and use such models.\nThere has been a surge of interest in creating
open-source tools for document\nimage processing: a search of document image analysis in Github leads to
5M\nrelevant code pieces 6; yet most of them rely on traditional rule-based methods\nor provide limited
functionalities. The closest prior research to our work is the\nOCR-D project7, which also tries to build a
complete toolkit for DIA. However,\nsimilar to the platform developed by Neudecker et al. [21], it is
designed for\nanalyzing historical documents, and provides no supports for recent DL models.\nThe
DocumentLayoutAnalysis project8 focuses on processing born-digital PDF\ndocuments via analyzing the stored
PDF data. Repositories like DeepLayout9\nand Detectron2-PubLayNet10 are individual deep learning models
trained on\nlayout analysis datasets without support for the full DIA pipeline. The Document\nAnalysis and
Exploitation (DAE) platform [15] and the DeepDIVA project [2]\naim to improve the reproducibility of DIA
methods (or DL models), yet they\nare not actively maintained. OCR engines like Tesseract [14], easyOCR11
and\npaddleOCR12 usually do not come with comprehensive functionalities for other\nDIA tasks like layout
analysis.\nRecent years have also seen numerous efforts to create libraries for promoting\nreproducibility and
reusability in the field of DL. Libraries like Dectectron2 [35],\n6 The number shown is obtained by specifying
the search type as ‘code’.\n7 https://ocr-d.de/en/about\n8 https://github.com/BobLd/DocumentLayoutAnalysis\n9
https://github.com/leonlulu/DeepLayout\n10 https://github.com/hpanwar08/detectron2\n11
https://github.com/JaidedAI/EasyOCR\n12 https://github.com/PaddlePaddle/PaddleOCR\n4\nZ. Shen et al.\nFig. 1:
The overall architecture of LayoutParser. For an input document image,\nthe core LayoutParser library
provides a set of off-the-shelf tools for layout\ndetection, OCR, visualization, and storage, backed by a
carefully designed layout\ndata structure. LayoutParser also supports high level customization via
efficient\nlayout annotation and model training functions. These improve model accuracy\non the target samples.
The community platform enables the easy sharing of DIA\nmodels and whole digitization pipelines to promote
reusability and reproducibility.\nA collection of detailed documentation, tutorials and exemplar projects
make\nLayoutParser easy to learn and use.\nAllenNLP [8] and transformers [34] have provided the community
with complete\nDL-based support for developing and deploying models for general computer\nvision and natural
language processing problems. LayoutParser, on the other\nhand, specializes specifically in DIA tasks.
LayoutParser is also equipped with a\ncommunity platform inspired by established model hubs such as Torch Hub
[23]\nand TensorFlow Hub [1]. It enables the sharing of pretrained models as well as\nfull document
processing pipelines that are unique to DIA tasks.\nThere have been a variety of document data collections to
facilitate the\ndevelopment of DL models. Some examples include PRImA [3](magazine layouts),\nPubLayNet [38]
(academic paper layouts), Table Bank [18](tables in academic\npapers), Newspaper Navigator Dataset [16, 17]
(newspaper figure layouts) and\nHJDataset [31](historical Japanese document layouts). A spectrum of
models\ntrained on these datasets are currently available in the LayoutParser model zoo\nto support different
use cases.\n', metadata={'heading': '2 Related Work\n', 'content_font': 9, 'heading_font': 11, 'source':
'example_data/layout-parser-paper.pdf'})
```

Using PyMuPDF

This is the fastest of the PDF parsing options, and contains detailed metadata about the PDF and its pages, as well as returns one document per page.

```
from langchain.document_loaders import PyMuPDFLoader
```

```
loader = PyMuPDFLoader("example_data/layout-parser-paper.pdf")
```

```
data = loader.load()
```

```
data[0]
```

```
    Document(page_content='LayoutParser: A Unified Toolkit for Deep\nLearning Based Document Image
Analysis\nZejiang Shen1 (�), Ruochen Zhang2, Melissa Dell3, Benjamin Charles Germain\nLee4, Jacob Carlson3,
and Weining Li5\n1 Allen Institute for AI\nshannons@allenai.org\n2 Brown University\nruochen
zhang@brown.edu\n3 Harvard University\n{melissadell,jacob carlson}@fas.harvard.edu\n4 University of
Washington\nbcgl@cs.washington.edu\n5 University of Waterloo\nw422li@uwaterloo.ca\nAbstract. Recent advances
in document image analysis (DIA) have been\nprimarily driven by the application of neural networks. Ideally,
research\noutcomes could be easily deployed in production and extended for further\ninvestigation. However,
various factors like loosely organized codebases\nand sophisticated model configurations complicate the easy
reuse of im-\nportant innovations by a wide audience. Though there have been on-going\nefforts to improve
reusability and simplify deep learning (DL) model\ndevelopment in disciplines like natural language
processing and computer\nvision, none of them are optimized for challenges in the domain of DIA.\nThis
represents a major gap in the existing toolkit, as DIA is central to\nacademic research across a wide range
of disciplines in the social sciences\nand humanities. This paper introduces LayoutParser, an open-
source\nlibrary for streamlining the usage of DL in DIA research and applica-\ntions. The core LayoutParser
library comes with a set of simple and\nintuitive interfaces for applying and customizing DL models for
layout de-\ntection, character recognition, and many other document processing tasks.\nTo promote
extensibility, LayoutParser also incorporates a community\nplatform for sharing both pre-trained models and
full document digiti-\nzation pipelines. We demonstrate that LayoutParser is helpful for both\nlightweight
and large-scale digitization pipelines in real-word use cases.\nThe library is publicly available at
https://layout-parser.github.io.\nKeywords: Document Image Analysis · Deep Learning · Layout Analysis\n·
Character Recognition · Open Source library · Toolkit.\n1\nIntroduction\nDeep Learning(DL)-based approaches
are the state-of-the-art for a wide range of\ndocument image analysis (DIA) tasks including document image
classification [11,\narXiv:2103.15348v2  [cs.CV]  21 Jun 2021\n', lookup_str='', metadata={'file_path':
'example_data/layout-parser-paper.pdf', 'page_number': 1, 'total_pages': 16, 'format': 'PDF 1.5', 'title':
'', 'author': '', 'subject': '', 'keywords': '', 'creator': 'LaTeX with hyperref', 'producer': 'pdfTeX-
1.40.21', 'creationDate': 'D:20210622012710Z', 'modDate': 'D:20210622012710Z', 'trapped': '', 'encryption':
None}, lookup_index=0)
```

Additionally, you can pass along any of the options from the PyMuPDF documentation as keyword arguments in the `load` call, and it will be pass along to the `get_text()` call.

PyPDF Directory

Load PDFs from directory

```
from langchain.document_loaders import PyPDFDirectoryLoader
```

```
loader = PyPDFDirectoryLoader("example_data/")
```

```
docs = loader.load()
```

Using pdfplumber

Like PyMuPDF, the output Documents contain detailed metadata about the PDF and its pages, and returns one document per page.

```
from langchain.document_loaders import PDFPlumberLoader
```

```
loader = PDFPlumberLoader("example_data/layout-parser-paper.pdf")
```

```
data = loader.load()
```

```
data[0]
```

```
    Document(page_content='LayoutParser: A Unified Toolkit for Deep\nLearning Based Document Image
Analysis\nZejiang Shen1 ((cid:0)), Ruochen Zhang2, Melissa Dell3, Benjamin Charles Germain\nLee4, Jacob
Carlson3, and Weining Li5\n1 Allen Institute for AI\n1202 shannons@allenai.org\n2 Brown University\nruochen
zhang@brown.edu\n3 Harvard University\nnuJ {melissadell,jacob carlson}@fas.harvard.edu\n4 University of
Washington\nbcgl@cs.washington.edu\n12 5 University of Waterloo\nw422li@uwaterloo.ca\n]VC.sc[\nAbstract.
Recentadvancesindocumentimageanalysis(DIA)havebeen\nprimarily driven by the application of neural networks.
Ideally, research\noutcomescouldbeeasilydeployedinproductionandextendedforfurther\ninvestigation. However,
various factors like loosely organized codebases\nand sophisticated model configurations complicate the easy
reuse of im-\n2v84351.3012:viXra portantinnovationsbyawideaudience.Thoughtherehavebeenon-going\nefforts to
improve reusability and simplify deep learning (DL)
model\ndevelopmentindisciplineslikenaturallanguageprocessingandcomputer\nvision, none of them are optimized
for challenges in the domain of DIA.\nThis represents a major gap in the existing toolkit, as DIA is central
to\nacademicresearchacross awiderangeof disciplinesinthesocialsciences\nand humanities. This paper introduces
LayoutParser, an open-source\nlibrary for streamlining the usage of DL in DIA research and applica-\ntions.
The core LayoutParser library comes with a set of simple
and\nintuitiveinterfacesforapplyingandcustomizingDLmodelsforlayoutde-
\ntection,characterrecognition,andmanyotherdocumentprocessingtasks.\nTo promote extensibility, LayoutParser
also incorporates a community\nplatform for sharing both pre-trained models and full document digiti-\nzation
pipelines. We demonstrate that LayoutParser is helpful for both\nlightweight and large-scale digitization
pipelines in real-word use cases.\nThe library is publicly available at https://layout-
parser.github.io.\nKeywords: DocumentImageAnalysis·DeepLearning·LayoutAnalysis\n· Character Recognition ·
Open Source library · Toolkit.\n1 Introduction\nDeep Learning(DL)-based approaches are the state-of-the-art
```

```
for a wide range of\ndocumentimageanalysis(DIA)tasksincludingdocumentimageclassification[11,', metadata=
{'source': 'example_data/layout-parser-paper.pdf', 'file_path': 'example_data/layout-parser-paper.pdf',
'page': 1, 'total_pages': 16, 'Author': '', 'CreationDate': 'D:20210622012710Z', 'Creator': 'LaTeX with
hyperref', 'Keywords': '', 'ModDate': 'D:20210622012710Z', 'PTEX.Fullbanner': 'This is pdfTeX, Version
3.14159265-2.6-1.40.21 (TeX Live 2020) kpathsea version 6.3.2', 'Producer': 'pdfTeX-1.40.21', 'Subject': '',
'Title': '', 'Trapped': 'False'})
```