Unstructured File Loader

Contents

This notebook covers how to use Unstructured to load files of many types. Unstructured currently supports loading of text files, powerpoints, html, pdfs, images, and more.

```
# # Install package
!pip install "unstructured[local-inference]"
!pip install
"detectron2@git+https://github.com/facebookresearch/detectron2.git@v0.6#egg=detectr
on2"
!pip install layoutparser[layoutmodels,tesseract]
```

```
# # Install other dependencies
# # https://github.com/Unstructured-
IO/unstructured/blob/main/docs/source/installing.rst
# !brew install libmagic
# !brew install poppler
# !brew install tesseract
# # If parsing xml / html documents:
# !brew install libxml2
# !brew install libxslt
```

```
# import nltk
# nltk.download('punkt')
```

```
from langchain.document_loaders import UnstructuredFileLoader
```

```
loader = UnstructuredFileLoader("./example_data/state_of_the_union.txt")
```

Skip to main content

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

```
docs[0].page_content[:400]
```

```
'Madam Speaker, Madam Vice President, our First Lady and Second Gentleman. Members
of Congress and the Cabinet. Justices of the Supreme Court. My fellow
Americans.\n\nLast year COVID-19 kept us apart. This year we are finally together
again.\n\nTonight, we meet as Democrats Republicans and Independents. But most
importantly as Americans.\n\nWith a duty to one another to the American people to
the Constit'
```

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 = UnstructuredFileLoader("./example_data/state_of_the_union.txt",
mode="elements")
```

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

```
docs[:5]
```

```
[Document(page_content='Madam Speaker, Madam Vice President, our First Lady and
Second Gentleman. Members of Congress and the Cabinet. Justices of the Supreme
Court. My fellow Americans.', lookup_str='', metadata={'source':
'../../state_of_the_union.txt'}, lookup_index=0),
 Document(page_content='Last year COVID-19 kept us apart. This year we are finally
together again.', lookup_str='', metadata={'source':
'../../state_of_the_union.txt'}, lookup_index=0),
 Document(page_content='Tonight, we meet as Democrats Republicans and
Independents. But most importantly as Americans.', lookup_str='', metadata=
{'source': '../../state_of_the_union.txt'}, lookup_index=0),
 Document(page_content='With a duty to one another to the American people to the
Constitution.', lookup_str='', metadata={'source':
'../../state_of_the_union.txt'}, lookup_index=0),
```

Skip to main content

```
triumph over tyranny.', lookup_str='', metadata={'source':
'../../state_of_the_union.txt'}, lookup_index=0)]
```

Define a Partitioning Strategy

Unstructured document loader allow users to pass in a `strategy` parameter that lets `unstructured` know how to partitioning the document. Currently supported strategies are `"hi_res"` (the default) and `"fast"`. Hi res partitioning strategies are more accurate, but take longer to process. Fast strategies partition the document more quickly, but trade-off accuracy. Not all document types have separate hi res and fast partitioning strategies. For those document types, the `strategy` kwarg is ignored. In some cases, the high res strategy will fallback to fast if there is a dependency missing (i.e. a model for document partitioning). You can see how to apply a strategy to an `UnstructuredFileLoader` below.

```
from langchain.document_loaders import UnstructuredFileLoader
```

```
loader = UnstructuredFileLoader("layout-parser-paper-fast.pdf", strategy="fast",
mode="elements")
```

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

```
docs[:5]
```

```
[Document(page_content='1', lookup_str='', metadata={'source': 'layout-parser-
paper-fast.pdf', 'filename': 'layout-parser-paper-fast.pdf', 'page_number': 1,
'category': 'UncategorizedText'}, lookup_index=0),
 Document(page_content='2', lookup_str='', metadata={'source': 'layout-parser-
paper-fast.pdf', 'filename': 'layout-parser-paper-fast.pdf', 'page_number': 1,
'category': 'UncategorizedText'}, lookup_index=0),
 Document(page_content='0', lookup_str='', metadata={'source': 'layout-parser-
paper-fast.pdf', 'filename': 'layout-parser-paper-fast.pdf', 'page_number': 1,
'category': 'UncategorizedText'}, lookup_index=0),
 Document(page_content='2', lookup_str='', metadata={'source': 'layout-parser-
paper-fast.pdf', 'filename': 'layout-parser-paper-fast.pdf', 'page_number': 1,
'category': 'UncategorizedText'}, lookup_index=0),
 Document(page_content='n', lookup_str='', metadata={'source': 'layout-parser-
paper-fast.pdf', 'filename': 'layout-parser-paper-fast.pdf', 'page_number': 1,
'category': 'Title'}, lookup_index=0)]
```

Skip to main content

PDF Example

Processing PDF documents works exactly the same way. Unstructured detects the file type and extracts the same types of `elements`.

```
!wget  https://raw.githubusercontent.com/Unstructured-IO/unstructured/main/example-docs/layout-parser-paper.pdf -P "../../"
```

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

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

```
docs[:5]
```

```
[Document(page_content='LayoutParser : A Unified Toolkit for Deep Learning Based Document Image Analysis', lookup_str='', metadata={'source': '../../layout-parser-paper.pdf'}, lookup_index=0),
 Document(page_content='Zejiang Shen 1 ( (ea)\n ), Ruochen Zhang 2 , Melissa Dell 3 , Benjamin Charles Germain Lee 4 , Jacob Carlson 3 , and Weining Li 5', lookup_str='', metadata={'source': '../../layout-parser-paper.pdf'}, lookup_index=0),
 Document(page_content='Allen Institute for AI shannons@allenai.org', lookup_str='', metadata={'source': '../../layout-parser-paper.pdf'}, lookup_index=0),
 Document(page_content='Brown University ruochen zhang@brown.edu', lookup_str='', metadata={'source': '../../layout-parser-paper.pdf'}, lookup_index=0),
 Document(page_content='Harvard University { melissadell,jacob carlson } @fas.harvard.edu', lookup_str='', metadata={'source': '../../layout-parser-paper.pdf'}, lookup_index=0)]
```