Document parser (supporting extracting tables in PDF for LLM):

LlamaParse/LlamaIndex

LlamaParse is a GenAI-native document parser that can parse complex document data for any downstream LLM use case (RAG, agents).

Parse a variety of unstructured file types (.pdf, .pptx, .docx, .xlsx, .html) with text, tables, visual elements, weird layouts, and more.

Parse embedded tables accurately into text and semi-structured representations.

Extract visual elements (images/diagrams) into structured formats and return image chunks using the latest multimodal models.

Input custom prompt instructions to customize the output the way you want it.

PaperQA2

PaperQA2 is a package for doing high-accuracy retrieval augmented generation (RAG) on PDFs or text files, with a focus on the scientific literature. See our recent 2024 paper to see examples of PaperQA2's superhuman performance in scientific tasks like question answering, summarization, and contradiction detection.

In this example we take a folder of research paper PDFs, magically get their metadata - including citation counts with a retraction check, then parse and cache PDFs into a full-text search index, and finally answer the user question with an LLM agent.

Document parser supporting PDF:

nv-ingest

A microservice that:

Accepts a JSON Job description, containing a document payload, and a set of ingestion tasks to perform on that payload.

Allows the results of a Job to be retrieved; the result is a JSON dictionary containing a list of Metadata describing objects extracted from the base document, as well as processing annotations and timing/trace data.

Supports PDF, Docx, pptx, and images.

Supports multiple methods of extraction for each document type in order to balance trade-offs between throughput and accuracy. For example, for PDF documents we support extraction via pdfium, Unstructured.io, and Adobe Content Extraction Services.

Supports various types of pre and post processing operations, including text splitting and chunking; transform, and filtering; embedding generation, and image offloading to storage.

NVIDIA-Ingest is a scalable, performance-oriented document content and metadata extraction microservice. Including support for parsing PDFs, Word and PowerPoint documents, it uses specialized NVIDIA NIM microservices to find, contextualize, and extract text, tables, charts and images for use in downstream generative applications.

NVIDIA Ingest enables parallelization of the process of splitting documents into pages where contents are classified (as tables, charts, images, text), extracted into discrete content, and further contextualized via optical character recognition (OCR) into a well defined JSON schema. From there, NVIDIA Ingest can optionally manage computation of

embeddings for the extracted content, and also optionally manage storing into a vector database Milvus.

markitdown

MarkItDown is a utility for converting various files to Markdown (e.g., for indexing, text analysis, etc). It supports:

PDF
PowerPoint
Word
Excel
Images (EXIF metadata and OCR)
Audio (EXIF metadata and speech transcription)
HTML
Text-based formats (CSV, JSON, XML)
ZIP files (iterates over contents)

MegaParse

Versatile Parser: MegaParse is a powerful and versatile parser that can handle various types of documents with ease.
No Information Loss: Focus on having no information loss during parsing.
Fast and Efficient: Designed with speed and efficiency at its core.
Wide File Compatibility: Supports Text, PDF, Powerpoint presentations, Excel, CSV, Word documents.
Open Source: Freedom is beautiful, and so is MegaParse. Open source and free to use.

OCR-based document parser supporting PDF:

paperless-ngx

Organize and index your scanned documents with tags, correspondents, types, and more.
Your data is stored locally on your server and is never transmitted or shared in any way.
Performs OCR on your documents, adding searchable and selectable text, even to documents scanned with only images.
Utilizes the open-source Tesseract engine to recognize more than 100 languages.
Documents are saved as PDF/A format which is designed for long term storage, alongside the unaltered originals.
Uses machine-learning to automatically add tags, correspondents and document types to your documents.
Supports PDF documents, images, plain text files, Office documents (Word, Excel, Powerpoint, and LibreOffice equivalents) and more.
Paperless stores your documents plain on disk. Filenames and folders are managed by paperless and their format can be configured freely with different configurations assigned to different documents.

docling

Parsing of multiple document formats incl. PDF, DOCX, XLSX, HTML, images, and more
Advanced PDF understanding incl. page layout, reading order, table structure, code, formulas, image classification, and more
Unified, expressive DoclingDocument representation format
Various export formats and options, including Markdown, HTML, and lossless JSON
Local execution capabilities for sensitive data and air-gapped environments
Plug-and-play integrations incl. LangChain, LlamaIndex, Crew AI & Haystack for agentic AI
Extensive OCR support for scanned PDFs and images
Simple and convenient CLI

OCR-based document parser supporting extracting tables in PDF:
https://github.com/VikParuchuri/surya

Surya is a document OCR toolkit that does: OCR in 90+ languages that benchmarks favorably vs cloud services、Line-level text detection in any language、Layout analysis (table, image, header, etc detection)、Reading order detection、Table recognition (detecting rows/columns)、LaTeX OCR
This command will write out a json file with the detected text and bboxes: surya_ocr DATA_PATH
DATA_PATH can be an image, pdf, or folder of images/pdfs

getomni-ai/zerox

A dead simple way of OCR-ing a document for AI ingestion. Documents are meant to be a visual representation after all. With weird layouts, tables, charts, etc. The vision models just make sense!
The general logic:
Pass in a file (pdf, docx, image, etc.)
Convert that file into a series of images
Pass each image to GPT and ask nicely for Markdown
Aggregate the responses and return Markdown

Layout analysis:

https://github.com/RapidAI/RapidLayout
该项目主要是汇集全网开源的版面分析的项目,具体来说,就是分析给定的文档类别图像(论文截图、研报等），定位其中类别和位置，如标题、段落、表格和图片等各个部分。

Other knowledge extraction tools:

ArchiveBox

Open source self-hosted web archiving. Take URLs/browser history/bookmarks/Pocket/Pinboard/etc. and keep in formats that other programs can

read directly. As output, we save standard HTML, PNG, PDF, TXT, JSON, WARC, SQLite, all guaranteed to be readable for decades to come.
It saves snapshots of the URLs you feed it in several redundant formats. It also detects any content featured inside pages & extracts it out into a folder:
HTML/Any websites: original HTML+CSS+JS, singlefile HTML, screenshot PNG, PDF, WARC, title, article text, favicon, headers, ...
Social Media/News: post content TXT, comments, title, author, images, ...
YouTube/SoundCloud/etc.: MP3/MP4s, subtitles, metadata, thumbnail, ...
Github/Gitlab/etc. links: clone of GIT source code, README, images, ...

Firecrawl
Firecrawl is an API service that takes a URL, crawls it, and converts it into clean markdown or structured data.

PDF2Audio
This code can be used to convert PDFs into audio podcasts, lectures, summaries, and more. It uses OpenAI's GPT models for text generation and text-to-speech conversion. You can also edit a draft transcript (multiple times) and provide specific comments, or overall directives on how it could be adapted or improved.