Get your docs in a row* with Docling

* reliably organized way

Carol Chen, Red Hat

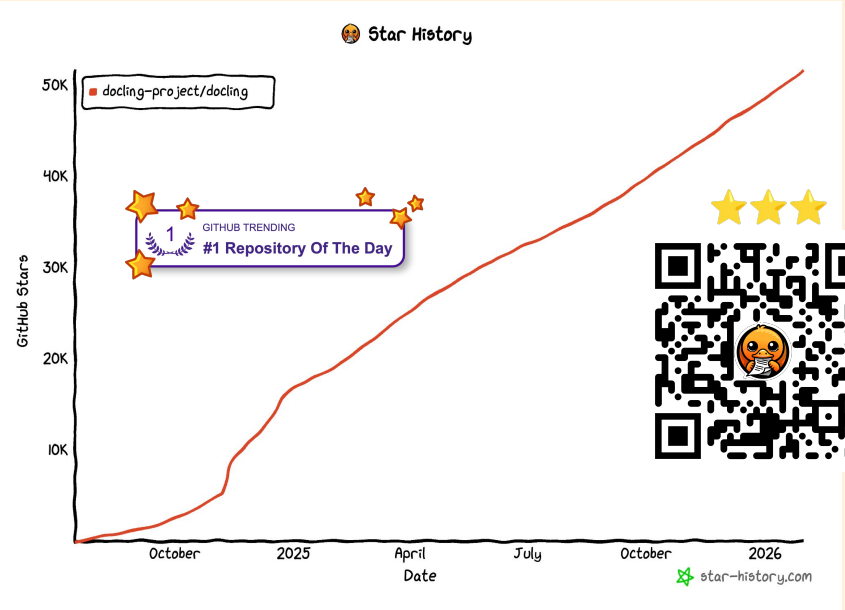
@cybette@mastodon.org.uk | @cybette:matrix.org | linkedin.com/in/cybette

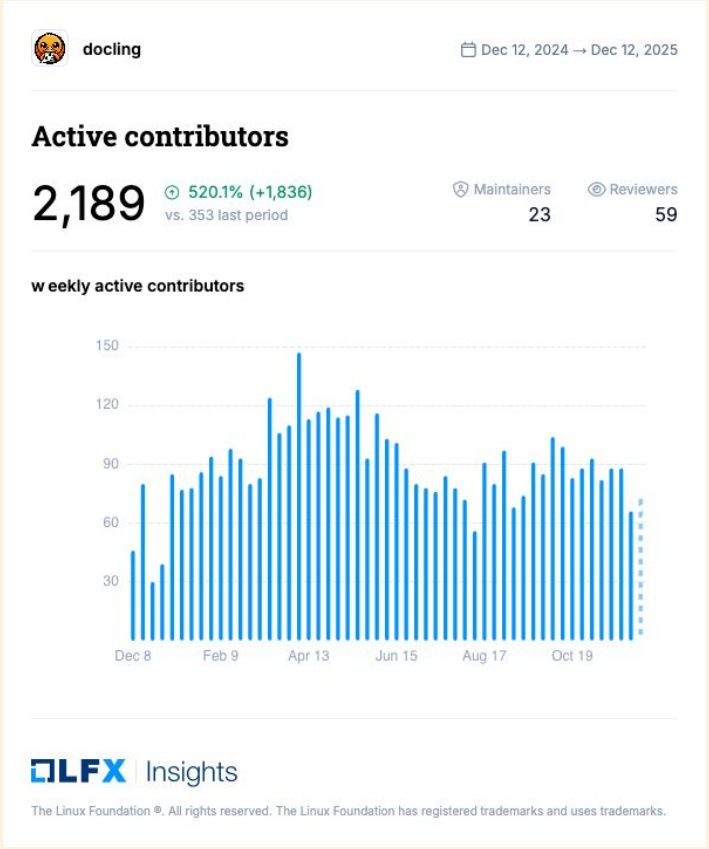
Introducing Docling

- 🗂️ 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, …
- 🧬 Unified, expressive DoclingDocument representation format
- ↪️ Various export formats (Markdown, HTML, JSON)
- 🔒 Local execution for sensitive data and air-gapped environments
- 🤖 Many plug-and-play ecosystem integrations
- 🔍 Extensive OCR support for scanned PDFs and images
- 🥚 Support of Visual Language Models
- 🎙️ Support for Audio with Automatic Speech Recognition model
- 💻 Simple and convenient CLI

Docling project

- Community adoption
- ⭐ 51k+ GitHub stars
- 🐍 2.4M+ downloads last month from PyPI





/docling-project

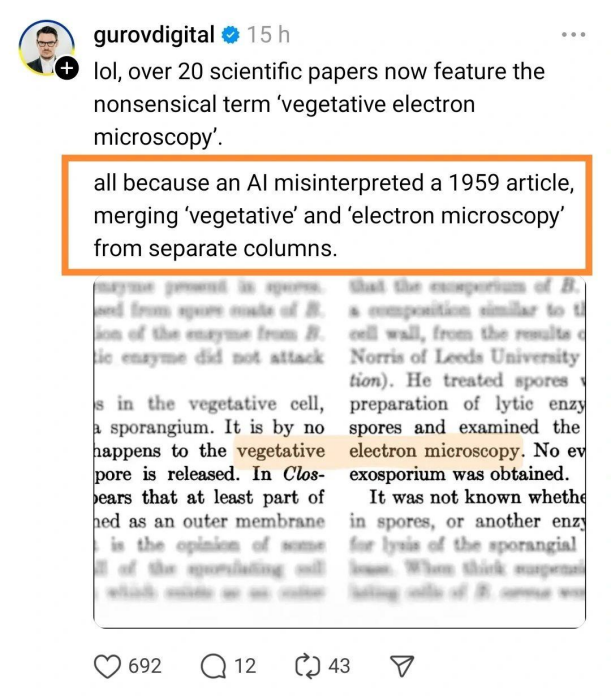
The quick brown fox

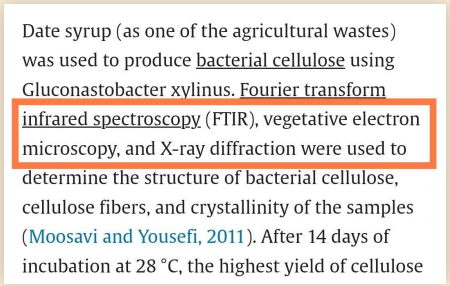
You know the story…
or do you?

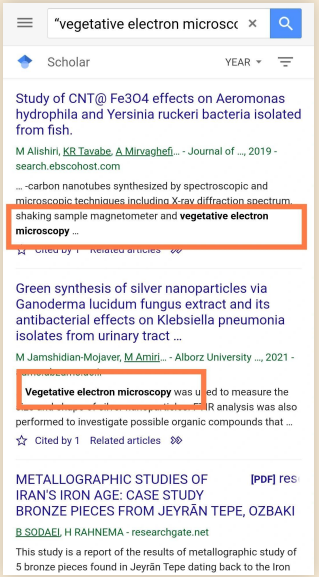
(image generated by Gemini)

Vegetative electron microscopy

https://www.freethink.com/artificial-intelligence/vegetative-electron-microscopy

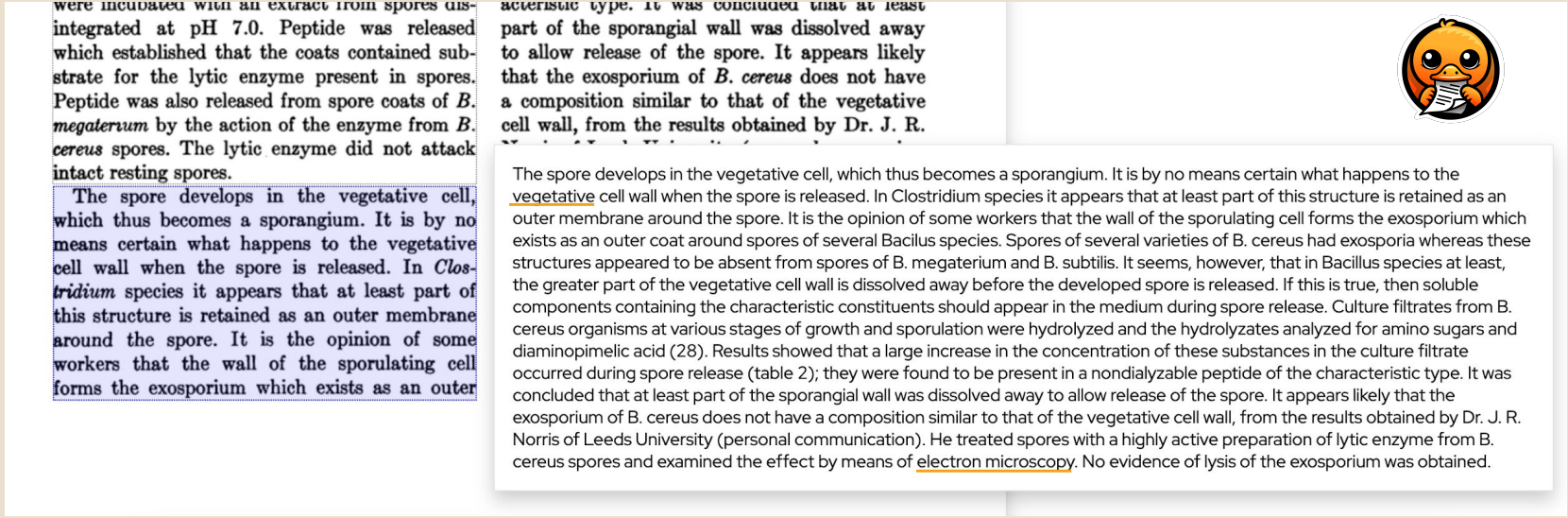






Vegetative cell wall

Output via Docling



were incubated with an extract from spores disintegrated at pH 7.0. Peptide was released which established that the coats contained substrate for the lytic enzyme present in spores. Peptide was also released from spore coats of *B. megaterium* by the action of the enzyme from *B. cereus* spores. The lytic enzyme did not attack intact resting spores.

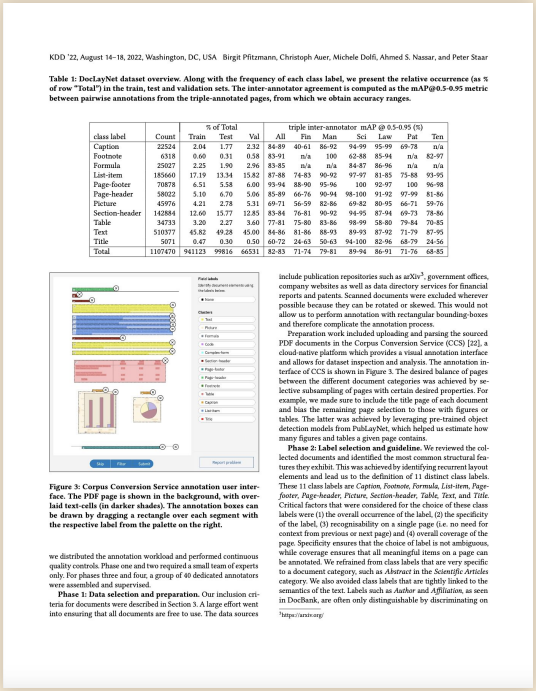
The spore develops in the vegetative cell, which thus becomes a sporangium. It is by no means certain what happens to the vegetative cell wall when the spore is released. In *Clostridium* species it appears that at least part of this structure is retained as an outer membrane around the spore. It is the opinion of some workers that the wall of the sporulating cell forms the exosporium which exists as an outer

acteristic type. It was concluded that at least part of the sporangial wall was dissolved away to allow release of the spore. It appears likely that the exosporium of *B. cereus* does not have a composition similar to that of the vegetative cell wall, from the results obtained by Dr. J. R.

The spore develops in the vegetative cell, which thus becomes a sporangium. It is by no means certain what happens to the vegetative cell wall when the spore is released. In Clostridium species it appears that at least part of this structure is retained as an outer membrane around the spore. It is the opinion of some workers that the wall of the sporulating cell forms the exosporium which exists as an outer coat around spores of several Bacilus species. Spores of several varieties of B. cereus had exosporia whereas these structures appeared to be absent from spores of B. megaterium and B. subtilis. It seems, however, that in Bacillus species at least, the greater part of the vegetative cell wall is dissolved away before the developed spore is released. If this is true, then soluble components containing the characteristic constituents should appear in the medium during spore release. Culture filtrates from B. cereus organisms at various stages of growth and sporulation were hydrolyzed and the hydrolyzates analyzed for amino sugars and diaminopimelic acid (28). Results showed that a large increase in the concentration of these substances in the culture filtrate occurred during spore release (table 2); they were found to be present in a nondialyzable peptide of the characteristic type. It was concluded that at least part of the sporangial wall was dissolved away to allow release of the spore. It appears likely that the exosporium of B. cereus does not have a composition similar to that of the vegetative cell wall, from the results obtained by Dr. J. R. Norris of Leeds University (personal communication). He treated spores with a highly active preparation of lytic enzyme from B. cereus spores and examined the effect by means of electron microscopy. No evidence of lysis of the exosporium was obtained.

Recovering structured content from PDF

with low level PDF parsers



KDD '22, August 14–18, 2022, Washington, DC, USA Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar

Table 1: DocLayNet dataset overview. Along with the frequency of each class label, we present the relative occurrence (as % of row "Total") in the train, test and validation sets. The inter-annotator agreement is computed as the mAP@0.5-0.95 metric between pairwise annotations from the triple-annotated pages, from which we obtain accuracy ranges.

		% of Total			triple inter-annotator mAP @ 0.5-0.95 (%)						
class label	Count	Train	Test	Val	All	Fin	Man	Sci	Law	Pat	Ten
Caption	22524	2.04	1.77	2.32	84-89	40-61	86-92	94-99	95-99	69-78	n/a
Footnote	6318	0.60	0.31	0.58	83-91	n/a	100	62-88	85-94	n/a	82-97
Formula	25027	2.25	1.90	2.96	83-85	n/a	n/a	84-87	86-96	n/a	n/a
List-item	185660	17.19	13.34	15.82	87-88	74-83	90-92	97-97	81-85	75-88	93-95
Page-footer	70878	6.51	5.58	6.00	93-94	88-90	95-96	100	92-97	100	96-98
Page-header	58022	5.10	6.70	5.06	85-89	66-76	90-94	98-100	91-92	97-99	81-86
Picture	45976	4.21	2.78	5.31	69-71	56-59	82-86	69-82	80-95	66-71	59-76
Section-header	142884	12.60	15.77	12.85	83-84	76-81	90-92	94-95	87-94	69-73	78-86
Table	34733	3.20	2.27	3.60	77-81	75-80	83-86	98-99	58-80	79-84	70-85
Text	510377	45.82	49.28	45.00	84-86	81-86	88-93	89-93	87-92	71-79	87-95
Title	5071	0.47	0.30	0.50	60-72	24-63	50-63	94-100	82-96	68-79	24-56
Total	1107470	941123	99816	66531	82-83	71-74	79-81	89-94	86-91	71-76	68-85

Figure 3: Corpus Conversion Service annotation user interface. The PDF page is shown in the background, with overlaid text-cells (in darker shades). The annotation boxes can be drawn by dragging a rectangle over each segment with the respective label from the palette on the right.

we distributed the annotation workload and performed continuous quality controls. Phase one and two required a small team of experts only. For phases three and four, a group of 40 dedicated annotators were assembled and supervised.

Phase 1: Data selection and preparation. Our inclusion criteria for documents were described in Section 3. A large effort went into ensuring that all documents are free to use. The data sources include publication repositories such as arXiv[3], government offices, company websites as well as data directory services for financial reports and patents. Scanned documents were excluded wherever possible because they can be rotated or skewed. This would not allow us to perform annotation with rectangular bounding-boxes and therefore complicate the annotation process.

Preparation work included uploading and parsing the sourced PDF documents in the Corpus Conversion Service (CCS) [22], a cloud-native platform which provides a visual annotation interface and allows for dataset inspection and analysis. The annotation interface of CCS is shown in Figure 3. The desired balance of pages between the different document categories was achieved by selective subsampling of pages with certain desired properties. For example, we made sure to include the title page of each document and bias the remaining page selection to those with figures or tables. The latter was achieved by leveraging pre-trained object detection models from PubLayNet, which helped us estimate how many figures and tables a given page contains.

Phase 2: Label selection and guideline. We reviewed the collected documents and identified the most common structural features they exhibit. This was achieved by identifying recurrent layout elements and lead us to the definition of 11 distinct class labels. These 11 class labels are *Caption*, *Footnote*, *Formula*, *List-item*, *Page-footer*, *Page-header*, *Picture*, *Section-header*, *Table*, *Text*, and *Title*. Critical factors that were considered for the choice of these class labels were (1) the overall occurrence of the label, (2) the specificity of the label, (3) recognisability on a single page (i.e. no need for context from previous or next page) and (4) overall coverage of the page. Specificity ensures that the choice of label is not ambiguous, while coverage ensures that all meaningful items on a page can be annotated. We refrained from class labels that are very specific to a document category, such as *Abstract* in the *Scientific Articles* category. We also avoided class labels that are tightly linked to the semantics of the text. Labels such as *Author* and *Affiliation*, as seen in DocBank, are often only distinguishable by discriminating on

[3] https://arxiv.org/

! undesired page headers

KDD '22, August 14–18, 2022, Washington, DC, USA Birgit Pfitzmann, Christoph Auer, Michele Dolfi, A Nassar, and Peter Staar

Table 1: DocLayNet dataset overview. Along with the frequency of each class label, we present the re occurrence (as %
of row "Total") in the train, test and validation sets. The inter-annotator agreement is computed as th mAP@0.5-0.95 metric
between pairwise annotations from the triple-annotated pages, from which we obtain accuracy range

% of Total

triple inter-annotator mAP @ 0.5-0.95 (%)

[...]
Count
22524
6318
25027
185660
70878
58022
45976
142884
34733
510377
5071
1107470

! Tables not understood

! Image content missing

[...]

include publication repositories such as arXiv3, government o"ces, company websites as well as data directory services for #nancial reports and patents. Scanned documents were excluded wherever possible because they can be rotated or skewed. This would not allow us to perform annotation with rectangular bounding-boxes and therefore complicate the annotation process.

! Line wraps not understood

[...] ! Multi-column often breaks order

✓ Very fast and cheap

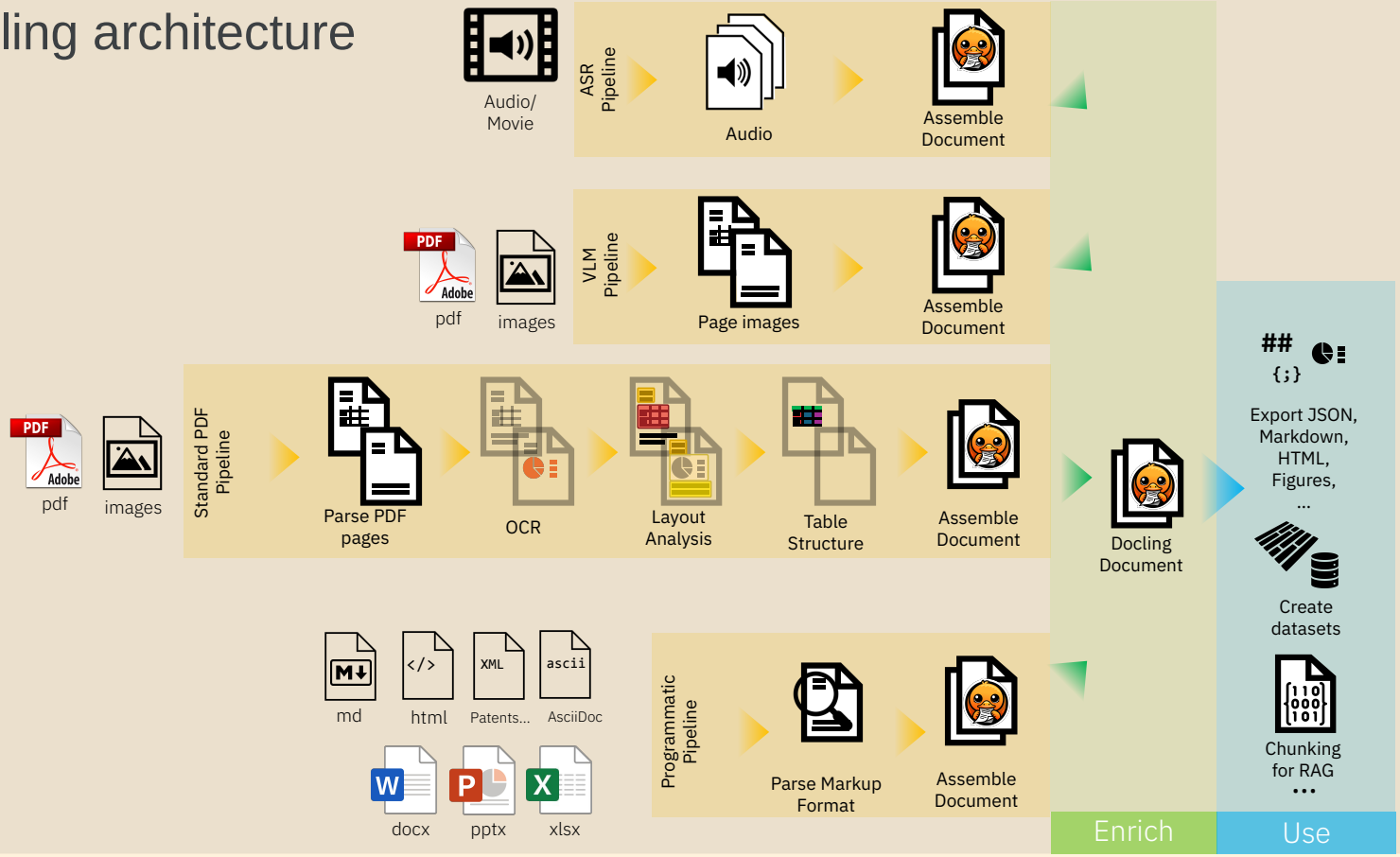
✗ Incomplete

✗ Loss of structure

✗ Noisy

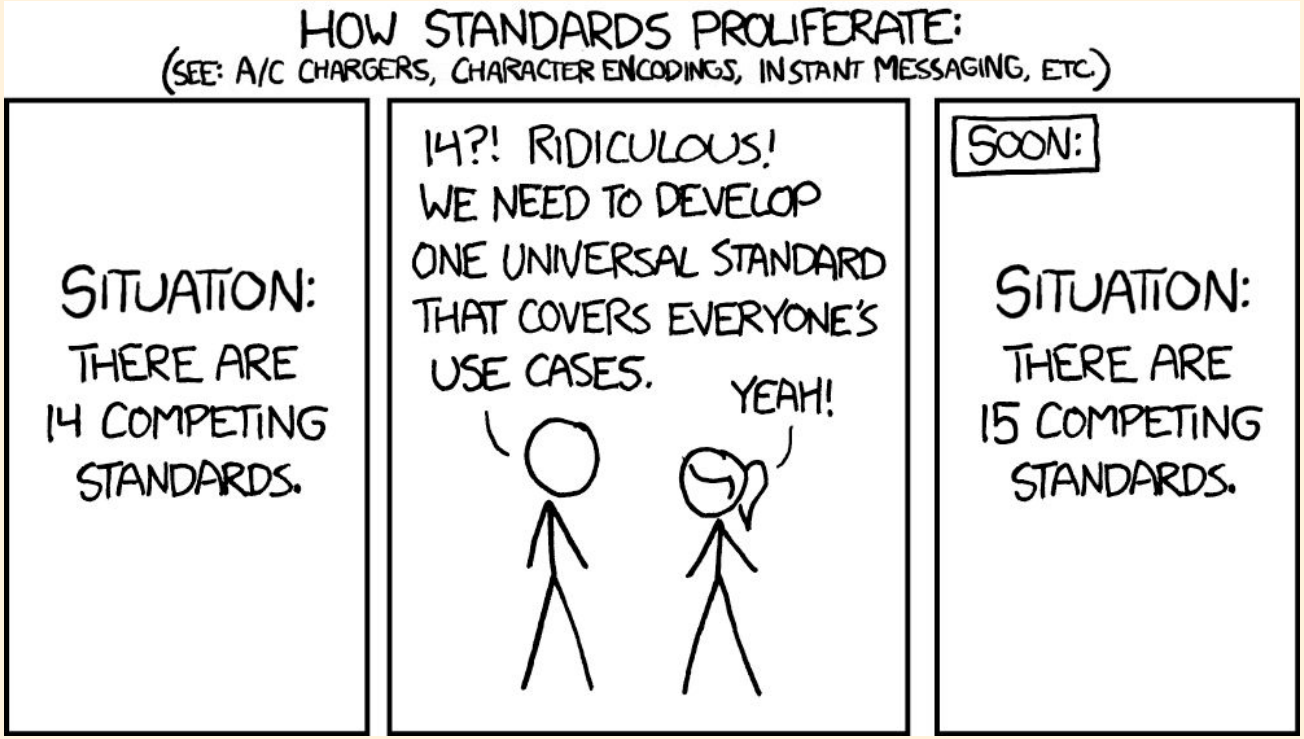
➔ Unfit for most use cases

Docling architecture



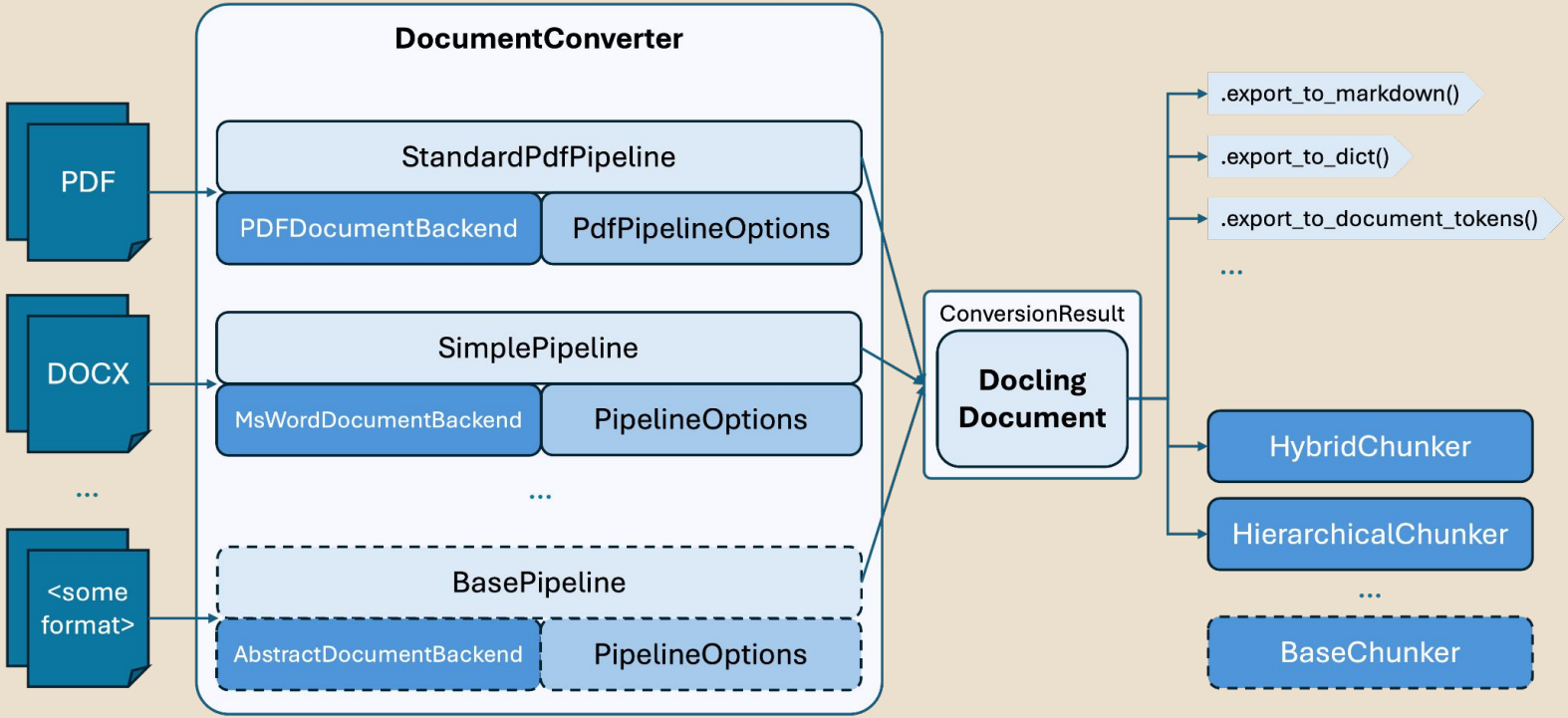
Obligatory xkcd

https://xkcd.com/927/



Docling architecture

https://docling-project.github.io/docling/concepts/architecture/

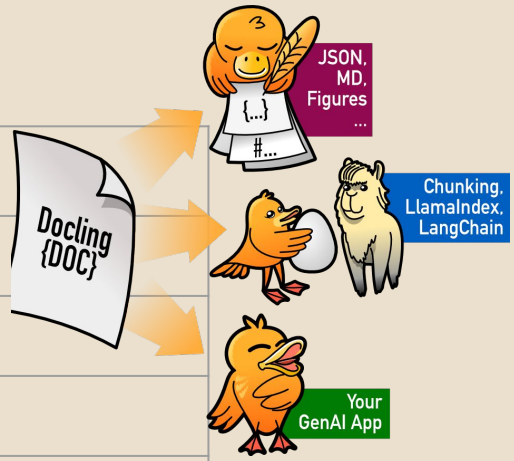


Supported input formats

Format	Description
PDF	
DOCX, XLSX, PPTX	Default formats in MS Office 2007+, based on Office Open XML
Markdown	
AsciiDoc	Human-readable, plain-text markup language for structured technical content
HTML, XHTML	
CSV	
PNG, JPEG, TIFF, BMP, WEBP	Image formats
WebVTT	Web Video Text Tracks format for displaying timed text

Supported output formats

Format	Description
HTML	Both image embedding and referencing are supported
Markdown	
JSON	Lossless serialization of Docling Document
Text	Plain text, i.e. without Markdown markers
Doctags	Markup format for efficiently representing the full content and layout characteristics of a document



https://docling-project.github.io/docling/usage/supported_formats/

DoclingDocument

DoclingDocument is a unified document representation format which can express several features common to documents, such as:

- Text, Tables, Pictures, and more
- Document hierarchy with sections and groups
- Disambiguation between main body and headers, footers (furniture)
- Layout information (i.e. bounding boxes) for all items, if available
- Provenance information

It also brings a set of document construction APIs to build up a DoclingDocument from scratch.

https://docling-project.github.io/docling/concepts/docling_document/

Introducing Duckling

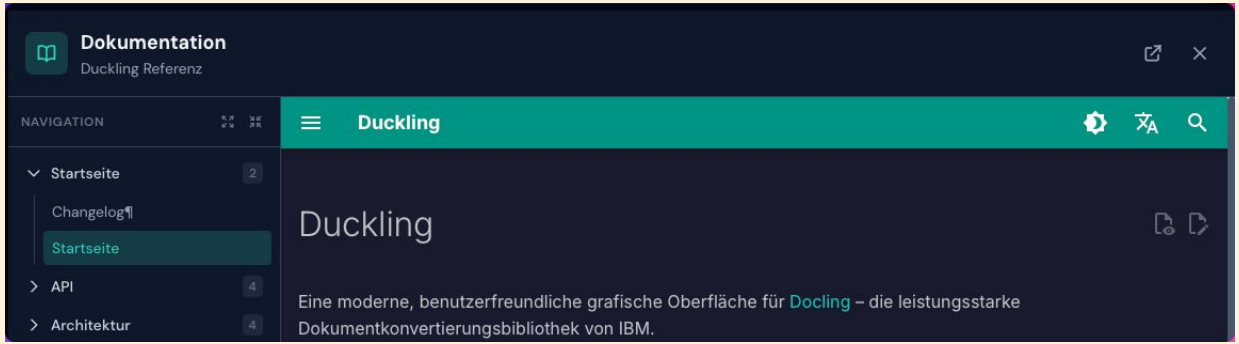
A modern, user-friendly graphical interface for Docling

Key features:

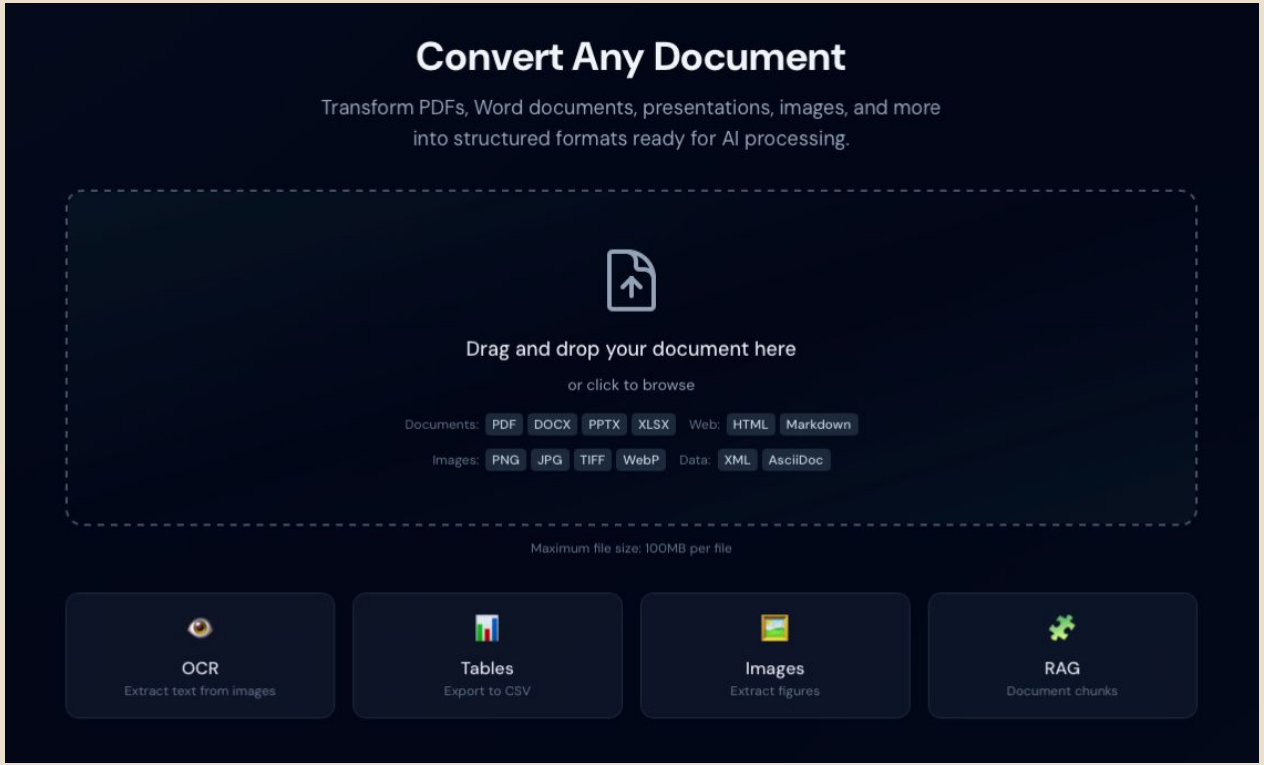
- Drag and drop files for instant processing
- Multiple input & output formats
- Image and table extraction
- RAG-ready chunking
- Conversion history

https://duckling-ui.org/

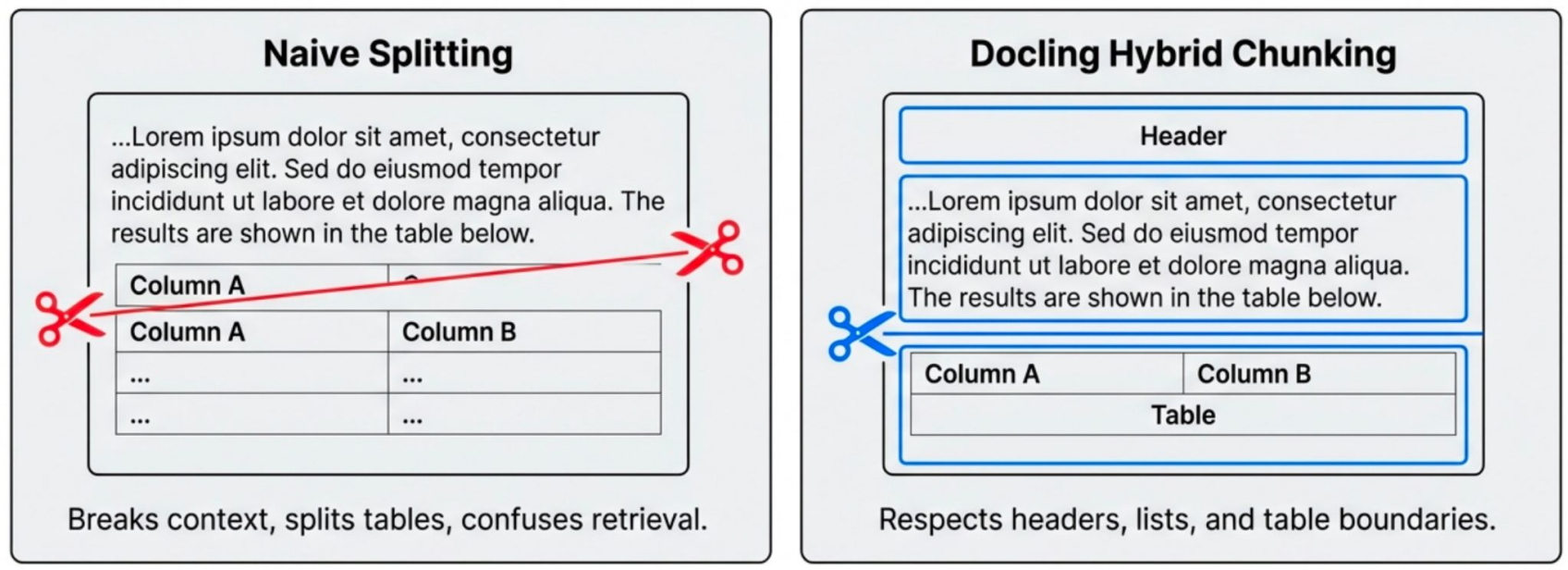
Latest release with i18n



Demo (maybe)


Convert Any Document
Transform PDFs, Word documents, presentations, images, and more into structured formats ready for AI processing.
Drag and drop your document here
or click to browse
Documents: PDF DOCX PPTX XLSX Web: HTML Markdown
Images: PNG JPG TIFF WebP Data: XML AsciiDoc
Maximum file size: 100MB per file
OCR
Extract text from images
Tables
Export to CSV
Images
Extract figures
RAG
Document chunks

Chunking for RAG



Enriched metadata per chunk helps embedding models understand layout and context, reducing hallucination.

Dockling

- Website: https://www.docling.ai/
- Github: https://github.com/docling-project
- HF: https://huggingface.co/docling-project
- Discord: https://www.docling.ai/discord

Duckling

- Website: https://duckling-ui.org/
- Github: https://github.com/davidgs/duckling

Resources

Thank you!

Acknowledgements:

Docling team (IBM Research Zurich) - Michele Dolfi, Peter Staar, Panos Vagenas, Ming Zhao

Duckling author (Red Hat) - David Simmons