



1 The challenge

A chemist in Product Development at RINGANA needs to work with a product but isn't sure what protective equipment to wear or how to handle it safely.



2 Searching is cumbersome

She has to search through many safety data sheets and documents to find the right information. It takes time and is inefficient.



3 Handing over the problem

She gives the pile of documents to the AI Engineer and asks for a solution that can quickly answer her specific safety questions.



4 Building the solution

The AI Engineer builds a smart system that reads, understands and indexes all the documents using RAG (Retrieval-Augmented Generation).



6 Easy answers, more time for impact

Now she can get answers instantly, work safely and with confidence, and focus on what really matters: developing great products.

Making Safety Sheets Conversational: A RAG Approach with Docling

Anna Saranti

RINGANA - SoFresh IT Solutions GmbH
Docling Community office hours

2026.05.27

- 1 Goal
- 2 Data exploration
- 3 Docling and VectorDB
- 4 Workflow with Langgraph+FastAPI
- 5 Evaluation
- 6 Takeaways
- 7 Acknowledgments

- 1 Goal
- 2 Data exploration
- 3 Docling and VectorDB
- 4 Workflow with Langgraph+FastAPI
- 5 Evaluation
- 6 Takeaways
- 7 Acknowledgments

Goal (1/2)



1 The challenge

A chemist in Product Development at RINGANA needs to work with a product but isn't sure what protective equipment to wear or how to handle it safely.



2 Searching is cumbersome

She has to search through many safety data sheets and documents to find the right information. It takes time and is inefficient.



3 Handing over the problem

She gives the pile of documents to the AI Engineer and asks for a solution that can quickly answer her specific safety questions.



4 Building the solution

The AI Engineer builds a smart system that reads, understands and indexes all the documents using RAG (Retrieval-Augmented Generation).



6 Easy answers, more time for impact

Now she can get answers instantly, work safely and with confidence, and focus on what really matters: developing great products.

AI-generated conceptual workflow illustration

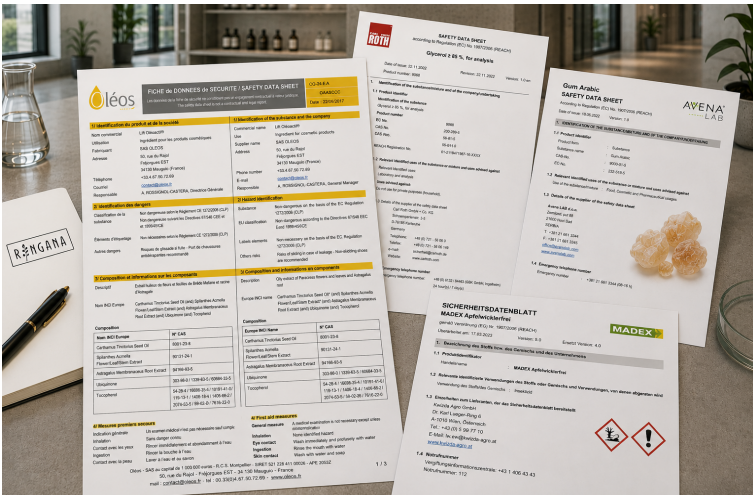


Goal (2/2)

- 1 New colleague in Product Development (PD)
Should he/she read all the ≥ 1000 documents?
- 2 Compare documents, e.g.,
How do sesame oils differ w.r.t. protective equipment?

- 1 Goal
- 2 Data exploration**
- 3 Docling and VectorDB
- 4 Workflow with Langgraph+FastAPI
- 5 Evaluation
- 6 Takeaways
- 7 Acknowledgments

Let's look at our safety sheets first...



Safety Sheets: [13], [14], [17], [18]



What is extracted?

Safety data sheet

according to Regulation (EC) No. 1907/2006 (REACH), amended by 2020/878/EU



1-Decanol ≥99 %, for synthesis

article number: 4151

Relevant PNECs of components						
Name of substance	CAS No	End-point	Threshold level	Organism	Environmental compartment	Exposure time
1-Dodecanol	112-53-8	PNEC	0 mg/l	aquatic organisms	marine water	short-term (single instance)
1-Dodecanol	112-53-8	PNEC	0,666 mg/kg	aquatic organisms	freshwater sediment	short-term (single instance)
1-Dodecanol	112-53-8	PNEC	0,067 mg/kg	aquatic organisms	marine sediment	short-term (single instance)
1-Dodecanol	112-53-8	PNEC	0,132 mg/kg	terrestrial organisms	soil	short-term (single instance)

8.2 Exposure controls

Individual protection measures (personal protective equipment)

Eye/face protection



Use safety goggle with side protection.

<https://www.carlroth.com/medias/SDB-4151-IE-EN.pdf?context=bwFzdG9yYHN1Y3VyaXR5RGFOYXNoZWV0c3wyNjQ1ODN8YXBwbGljYXRpb24vcGRmfGFESmxMMmcyTkM4NU1qSTBNekF6TURVd056Z3lMMU5FUWw4ME1UVXhYMxGWDBWTOxuQmtaZ3wyOTYzNTZkMzYjFjYzZjMjk2NDEzN2IOZGFjNzdkOWMwNDNhNThmY2MOZTViYzEyNjc4MjdjMzBiMmVi>

- 1 Text
- 2 Tables
- 3 Images with text (company name, website, phone number)
- 4 Hazard pictograms without text

Hazard pictograms



- 1 Goal
- 2 Data exploration
- 3 Docling and VectorDB**
- 4 Workflow with Langgraph+FastAPI
- 5 Evaluation
- 6 Takeaways
- 7 Acknowledgments

Docling Processing Workflow (1/4)

```
import fitz

...

doc = fitz.open(pdf_file)
```

PyMuPDF: <https://pymupdf.readthedocs.io/en/latest/>

```
chunker = HybridChunker(
    tokenizer=tokenizer,
    merge_peers=True,
)
chunk_iter = chunker.chunk(dl_doc=result.document)
docling_chunks = list(chunk_iter)
```

HybridChunker: [5], [6]

Docling Processing Workflow (2/4)

```
pipeline_options = PdfPipelineOptions()
pipeline_options.ocr_options =
    TesseractCliOcrOptions(force_full_page_ocr=True)
pipeline_options.images_scale = 2
pipeline_options.generate_picture_images = True
pipeline_options.do_ocr = True
```

```
converter = DocumentConverter(
    format_options={
        InputFormat.PDF: PdfFormatOption(
            pipeline_options=pipeline_options,
            backend=PyPdfiumDocumentBackend,
        )
    }
)
result = converter.convert(input_pdf_to_chunk)
```

Docling Processing Workflow (3/4)

Example chunk:

```
{
  "chunk_text": "Violent reaction with: strong oxidizer,
                acetic anhydride, acid chlorides, inorganic",
  "source_file": "SDB-Decanol_4151_IE_EN_pymupdf.pdf",
  "section": "Possibility of hazardous reactions",
  "chunk_type": "text",
  "page": 8,
  "coordinates": {
    "origin": "bottom-left",
    "left": 74.67,
    "top": 452.67,
    "right": 458.67,
    "bottom": 442.67
  }, ...
}
```

Docling Processing Workflow (4/4)

What can go wrong? - Corrupt *.pdf

```

confidence_obj = result.confidence
logger.info("Confidence Score:")
logger.info(f"Overall grade:
            {confidence_obj.mean_grade}")
logger.info(f"Worst section grade:
            {confidence_obj.low_grade}")
logger.info(f"Numeric score (0-1):
            {confidence_obj.layout_score}")
logger.info(f"OCR score:
            {confidence_obj.ocr_score}")
logger.info(f"Parse score:
            {confidence_obj.parse_score}")
    
```

```

Chunk 32 | Page: 4 | Type: text
Headings: ["' 0"]
Text: 4 ' %([ " H ( & ? ([ %([ " ) 4
J &" ' ' #4 A: ! < -
5 * ?="'% & %([ ...
-----
Chunk 33 | Page: 4 | Type: text
Headings: ['6$. $ / !']
Text: : 4'( ) ) = - > ' (%&& '(% ...
-----
Chunk 34 | Page: 4 | Type: text
Headings: [': ; ! " !']
Text: @ ' 8 & ( (4 "( & % !( ) ) % / * 4
-----
Chunk 35 | Page: 4 | Type: text
Headings: ['''']
Text: ?= ' %#4 5 "="( ' H * ' ' %
-----
Chunk 36 | Page: 4 | Type: text
Headings: ['> # = #']
Text: @ 3) %#4( % (
+ 2 8 ' ' ) ( #4 + 4 * ' 8
    
```


Vector DB schema (2/2)

- Principles for designing the VectorDB schema: [4], [1]
⇒ 1 namespace \forall ingredient
- **BM25** [8],
https:
`//docs.pinecone.io/models/pinecone-sparse-english-v0`
or
TF-IDF ? [7]
`https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html`
Keywords that appear only once
f.e.: emergency telephone number

Text extraction from images (1/7)

Use **Docling** to extract all images from the *.pdf files:

```
from docling.backend.pypdfium2_backend
    import PyPdfiumDocumentBackend

...
pipeline_options = PdfPipelineOptions()
pipeline_options.generate_picture_images = True
...
doc = convert_with_backend(PyPdfiumDocumentBackend)
for element, _ in doc.document.iterate_items():
    if isinstance(element, PictureItem):
        img = element.get_image(doc.document)
        ...
        if element.prov:
            page_number = element.prov[0].page_no
        img.save(...
```

Text extraction from images (2/7)

- Best results for textual extraction from images with PaddleOCR: [9]

```
from PIL import Image
from paddleocr import PaddleOCR
...
img = pil_to_cv2(pil_img)
results = paddle_ocr.ocr(img)
texts_list = []
if results is not None and results[0] is not None:
    texts_list = [line[1][0] for line in results[0]]
```

Text extraction from images (3/7)



```
'enk',  
'DISTRIBUTED BY:',  
'Ingredients',  
'DENK Ingredients GmbH',  
'Neherstrasse 9',  
'81675 Muenchen, Germany',  
'Tel +4989-230029-400',  
'Fax +49 89 - 23 00 29 - 444',  
'info@denkingredients.de',  
'www.denkingredients.de']
```

<https://www.denkingredients.de/kontakt/>
<https://www.ibbnetzwerk-gmbh.com/en/partners/denk-ingredients-gmbh>

- Impressive result, but completely inadequate for the hazard pictograms
- Sometimes, the textures are misrecognized as words; f.e. hand gloves like the word “m”, protective glasses like “OO”

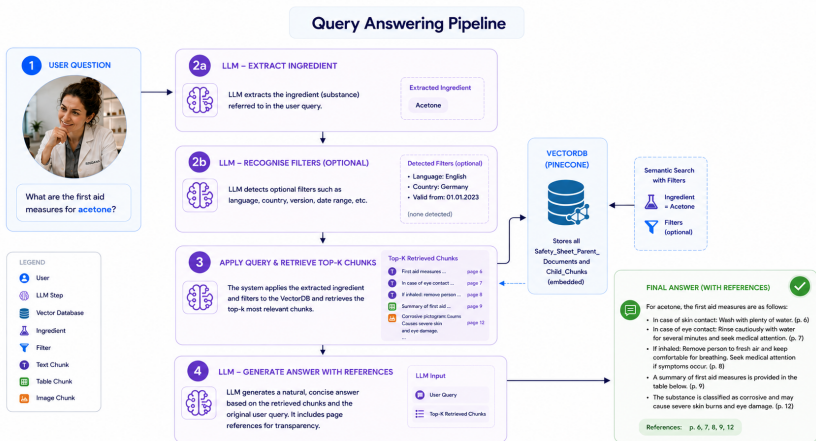
Text extraction from images (4/7)

```
prompt = ... "You are a safety compliance expert specialising in chemical safety data sheets (SDS/GHS). The image is a single safety pictogram or label extracted from a safety data sheet. If you see an image containing text, don't analyze it and return an empty string. Else, it may be ONE of the following categories — identify which one applies:  
A) GHS HAZARD PICTOGRAM: red-bordered diamond with a white background and a black hazard symbol (e.g. flame, skull-and-crossbones, exclamation mark, corrosion, health hazard silhouette, dead tree/fish, oxidiser flame-over-circle, gas cylinder, exploding bomb).  
B) PPE / MANDATORY ACTION SIGN: blue circular sign showing personal protective equipment (e.g. gloves, goggles, face shield, respirator, lab coat, safety boots) or a mandatory action (e.g. eye-wash station).  
C) FIRST-AID / SAFE CONDITION SIGN: green rectangular or square sign (e.g. white cross on green background, first-aid kit, emergency shower)."
```



- 1 Goal
- 2 Data exploration
- 3 Docling and VectorDB
- 4 Workflow with Langgraph+FastAPI**
- 5 Evaluation
- 6 Takeaways
- 7 Acknowledgments

Workflow (1/4)



Workflow (2/4)

UI with FastAPI: <https://fastapi.tiangolo.com/>, [10]

The screenshot shows a web browser window with a chat interface. The address bar shows '127.0.0.1:8000/chat'. On the left, there is a sidebar with a '+ New Chat' button and a text input field containing 'I need the emergency number of...'. The main chat area contains the following messages:

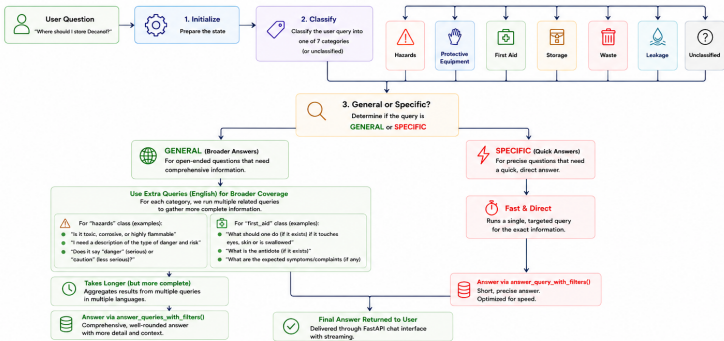
- User: I need the emergency number of coconut
- Assistant: The emergency phone number for coconut products is 1-██████████ (USA).
The information in the answer was retrieved from the following document(s) and page number(s): (Coconut Water Powder_MSDS_pymupdf.pdf, page number: 1)
- User: OK und was muss ich machen falls Kokosnuss auf meine Augen kommt?
- Assistant: Falls Kokosnuss(-pulver) in die Augen gelangt, sollten Sie zur Vermeidung mechanischer Reizung die Augen mit sauberem, schwach fließendem Wasser ausspülen.
Weitere Maßnahmen sind in der Regel nicht erforderlich, da keine akuten oder verzögerten Symptome bekannt sind.
The information in the answer was retrieved from the following document(s) and page number(s): (Coconut Water Powder_MSDS_pymupdf.pdf, page number: 1)

Workflow (3/4)

- 1 Ingredient name extraction
- 2 Optionally applying filtering
- 3 Computing the semantic and sparse vectors of the query with the pre-computed TF-IDF data structures (one \forall language)
- 4 Multi-lingual: use the ISO 639-1 two-letter code format (“en”, “de”)
- 5 Retrieval post-processing: from many contexts, sorted by their relevance, create a coherent answer [1]

Workflow (4/4)

Safety Sheets Assistant – LangGraph Overview



EXTRA QUERIES (ENGLISH) PER CATEGORY

To support GENERAL questions, we expand the search using multiple related queries.

Hazards	"hazard", "risk", "danger", "health effects", ...	Waste	"disposal", "waste", "empty", "discard", "waste code", ...
Protective Equipment	"PPE", "goggles", "goggles", "respiratory protection", ...	Leakage	"leak", "leak", "release", "containment", "cleanup", ...
First Aid	"first aid", "what to do", "medical attention", ...		
Storage	"storage", "store", "keep", "conditions", "temperature", ...		

GENERAL vs SPECIFIC

Key Point 2: Speed vs Completeness

<p>GENERAL</p> <ul style="list-style-type: none"> Broader questions Uses extra queries fast Searches more content More comprehensive answer Takes longer <p>Best when the user wants a full explanation.</p> <p>More Info More Time</p>	<p>SPECIFIC</p> <ul style="list-style-type: none"> Precise questions Single targeted query Minimal content needed Short, direct answer Very fast <p>Best when the user needs a quick, initial answer.</p> <p>Less Time Quick Answer</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Key Point 1: These extra queries allow the system to retrieve more relevant passages and provide richer, more complete answers for GENERAL questions.

Key Point 2: We automatically choose the right path: more depth (slow) for GENERAL, or speed (fast) for SPECIFIC.

How it works

- User asks a question
- System classifies the category
- Determines General or Specific
- Routes to the right processing path
- Returns the best answer

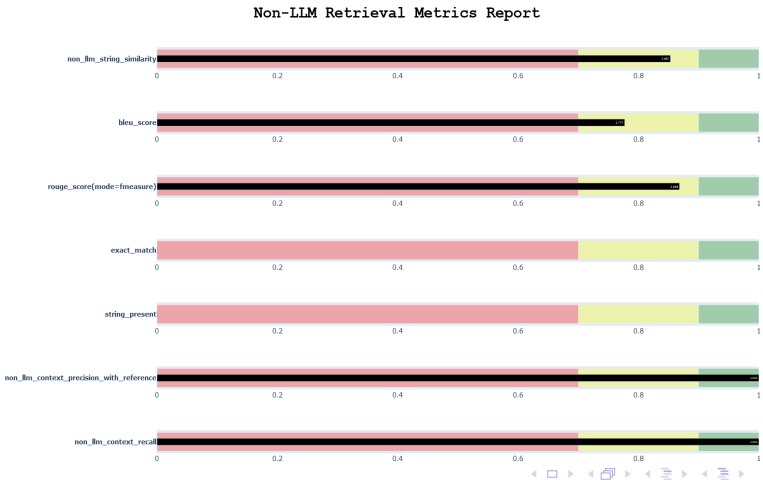
LEGEND

- General (broader)
- Specific (quick)
- Key Highlights

- 1 Goal
- 2 Data exploration
- 3 Docling and VectorDB
- 4 Workflow with Langgraph+FastAPI
- 5 Evaluation**
- 6 Takeaways
- 7 Acknowledgments

Evaluation with RAGAS metrics (1/2)

RAGAS [11], [12] metrics reports with plotly:
<https://plotly.com/python/>



First checks before RAGAS (2/2)

Query types that should have **top** results:

- 1 Queries with a term that occurs only once (1) in the document
f.e. “emergency number”.
- 2 Queries based only on the semantics; f.e. the document speaks about how to extinguish a fire, and the query sentence is “I am burning”.
- 3 Queries that contain both recognisable keywords and well-recognisable semantic similarity, like “eyes”, “swallowing”.
- 4 “Combinations” of queries f.e. “The product is in my eyes, what should I do and who should I call?”

- 1 Goal
- 2 Data exploration
- 3 Docling and VectorDB
- 4 Workflow with Langgraph+FastAPI
- 5 Evaluation
- 6 Takeaways**
- 7 Acknowledgments

Takeaway

Retrieval is
to the point
when successful 😊.

If you are unhappy
with the results,
keep debugging,
keep improving.

Everything else
relies on that.

⇒ with **Docling**,
it's possible!



Future Directions (1/2)

- More complex documents - receipts, legal, advertisement
- Section titles on top of images ⚠️

https://rin-wp-media.s3.eu-central-1.amazonaws.com/2025/01/0ayxCJH4-RINGANA_Transparency_Report_EN_V2_112024.pdf

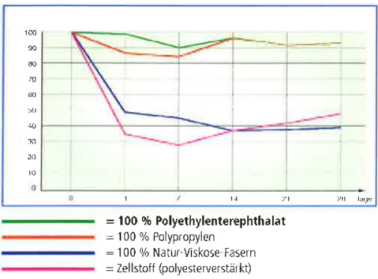
HOW MUCH WE CAN IMPROVE PEOPLE IF THEY SIMPLY USE AMAZING PRODUCTS TO THIS WAY OF THINKING.

and this value system can improve our

How long have you already worked for as a RINGANA Partner?
 I became a Partner in 2015. I was getting to know my now wife at the time, and she had already been a Partner since 2009. I thought the company RINGANA and the products were cool, but network marketing had some negative connotations for me. This is because I used to be a business consultant and carried out the restructure of a network marketing company where it didn't work out so well. For the sake of keeping the family peace, though, I got involved and sometimes went to events. But the decision to get serious about it

It's clear to us how much our children benefit from all this flexibility. That doesn't exist in any other job. The financial opportunities are one thing, the other is the quality of life and this value system that we can pass on to our children.

How many hours do you invest each week?
 That's hard to say because it doesn't feel at all like an investment. (Laughs). I would say four to five hours a day and my wife does the same again. Around one to two hours a week are spent on administrative tasks like accounting and tax



Future Directions (2/2)

- 1 Multimodal LLMs
- 2 AgenticRAG
- 3 Weaviate Query Agent
<https://docs.weaviate.io/agents/query/usage>
- 4 PageIndex <https://pageindex.ai/>

```
f"Create a comparison of the
hazard warnings one:
{hazards_product_1}"
f"and the hazard warnings two:
{hazards_product_2}. "
f"Use maximally
{number_to_words_max_str}
({max_nr_comp_sentences})
sentences for your comparison."
f"The language of the comparison
must be the one that was used "
f"in the main query."
```


- 1 Goal
- 2 Data exploration
- 3 Docling and VectorDB
- 4 Workflow with Langgraph+FastAPI
- 5 Evaluation
- 6 Takeaways
- 7 Acknowledgments**

Acknowledgments (2/2)


- 🌐 Austrian Chamber for Workers and Employees (**AK**)
- 🌐 BOKU Staff council - scientific staff
- 🌐 Austrian Data Protection Authority (**DSB**)
- 🌐 Austrian Federal Ministry of the Interior
(Bundesministerium für Inneres)
- 🌐 Federal Criminal **Police** Office (Bundeskriminalamt)
- 🌐 Victims of **Cybercrime** in the City of Vienna
(Opfer von Internet-Kriminalität der Stadt Wien)

Questions?


Contact **only** on LinkedIn:

 <https://www.linkedin.com/in/dr-techn-dipl-ing-anna-saranti-865b7812a/>

due to reasons explained on:

 <https://annasaranti.ai/personal-statement/>

 <https://www.ringana.com/>

 <https://www.sofresh-it.com/>

References (1/10)

- [1] <https://annasaranti.ai/context-engineering-in-practice-at-the-vienna-ai-engineering-meetup/>
- [2] Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, Lokesh Mishra, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, and Peter W. J. Staar
Docling Technical Report
arXiv preprint arXiv:2408.09869, 2024

References (2/10)

- [3] [Tony Larson](#)
Vector Database Engineering: Building Scalable AI Search & Retrieval Systems with FAISS, Milvus, Pinecone, Weaviate, RAG Pipelines, Embeddings, High Dimension Indexing (with Mathematical Equations)
[Amazon Digital Services LLC - Kdp, 2025](#)
- [4] [Christian Hinkler](#)
Practical Weaviate: RAG and Vector Database Patterns for LLMs
[Independently published, 2025.](#)

References (3/10)

- [5] Shuchen Wu, Noémi Éltető, Ishita Dasgupta, and Eric Schulz
Learning Structure from the Ground up—Hierarchical Representation Learning by Chunking
Advances in Neural Information Processing Systems, 2022, 35. Jg.,
S. 36706-36721.
- [6] Wensheng Lu, Keyu Chen, Ruizhi Qiao, and Xing Sun
Hichunk: Evaluating and enhancing retrieval-augmented generation with hierarchical chunking.
arXiv preprint arXiv:2509.11552, 2025.

References (4/10)

- [7] Stephen Robertson, and Hugo Zaragoza
The Probabilistic Relevance Framework: BM25 and Beyond (Vol. 4)
Now Publishers Inc, 2009
- [8] Sparck Jones, Karen
A statistical interpretation of term specificity and its application in retrieval
1972, 28. Jg., Nr. 1, S. 11-21

References (5/10)

[9] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, Dianhai Yu, Yanjun Ma
PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System
arXiv preprint arXiv:2206.03001, 2022.

References (6/10)

- [10] De Luca, Giunio
FastAPI Cookbook: Develop high-performance APIs and web applications with Python
Packt Publishing Ltd, 2024.

References (7/10)

- [11] Shahul Es, Jithin James, Luis Espinosa Anke and Steven Schockaert
Ragas: Automated Evaluation of Retrieval Augmented Generation
Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations., 2024
- [12] Nelson F. Liu, Tianyi Zhang and Percy Liang
Evaluating Verifiability in Generative Search Engines
arXiv preprint arXiv:2304.09848, 2023

References (8/10)

Safety Sheets:

- [13] <https://share.google/IVbCMi4nCmfacf6Ua>
- [14] <https://www.carlroth.com/medias/SDB-6962-IE-EN.pdf?context=bWFzdGVyfHhNlY3VyaXR5RGFOYXNoZWV0c3wyNDMxNDJ8YXBwbGljYXRpb24vcGRmfGFEazRMMmhqTXk4NU1UYzVPVF14TmpnNE1EazBMMU5FUWw4Mk9UWX1YMGxGWDBWTOxUQmtaZ3w2NTk1Y2Y3NGRkYjVhNDQwNGUxNTQxNWI2Yzk4OTEyZ2ZlMDdlZW11NWYwZDMwZDkxY2I3ZjA5NDJiOTEyODUz>
- [15] https://hugohaeffner.com/wp-content/import_productfiles/specsheets/DE_HAEF106290_ZITRONENSAEURE_FLUESSIG_50_TECHN_DE_DE_2_0.PDF
- [16] <https://www.carlroth.com/medias/SDB-4151-IE-EN.pdf?context=bWFzdGVyfHhNlY3VyaXR5RGFOYXNoZWV0c3wyNjQ1ODN8YXBwbGljYXRpb24vcGRmfGFEsmxMMmcyTkM4NU1qSTBNekF6TURVd056Z3lMMU5FUWw4ME1UVXhYMGxGWDBWTOxUQmtaZ3wyOTYzNTZkMzM2MmWQ3YjFjYzZjMjk2NDEzN2IOZGFjNzdkOWMwNDNhNThmY2MOZTViYzEyNjc4MjdjMzBiMmVi>

References (9/10)

- [17] https://avenalab.com/images/0001DOKUMENTACIJA/SDS/Gum_Arabic_SDS_ENG_signed.pdf
- [18] https://shop.garten-bienen.at/files/pdf/pdf_Sicherheitsdatenblatt/SDS_Madex_Apfelwicklerfrei_DE_03_2023.pdf?srsltid=AfmB0opPDRGzaLw-STnjErYeoor5LaCplQGEHh4ZNg0Gw5cinSXsgPLS

References (10/10)

RIGNANA Transparency Report:

- [19] https://rin-wp-media.s3.eu-central-1.amazonaws.com/2025/01/0ayxCJH4-RINGANA_Transparency_Report_EN_V2_112024.pdf