



Connect

Unlocking Your Company's Knowledge: Building an AI Assistant with RHEL AI





Mahalakshmi Vijayakumar

Consultant
Red Hat



Paulo Menon

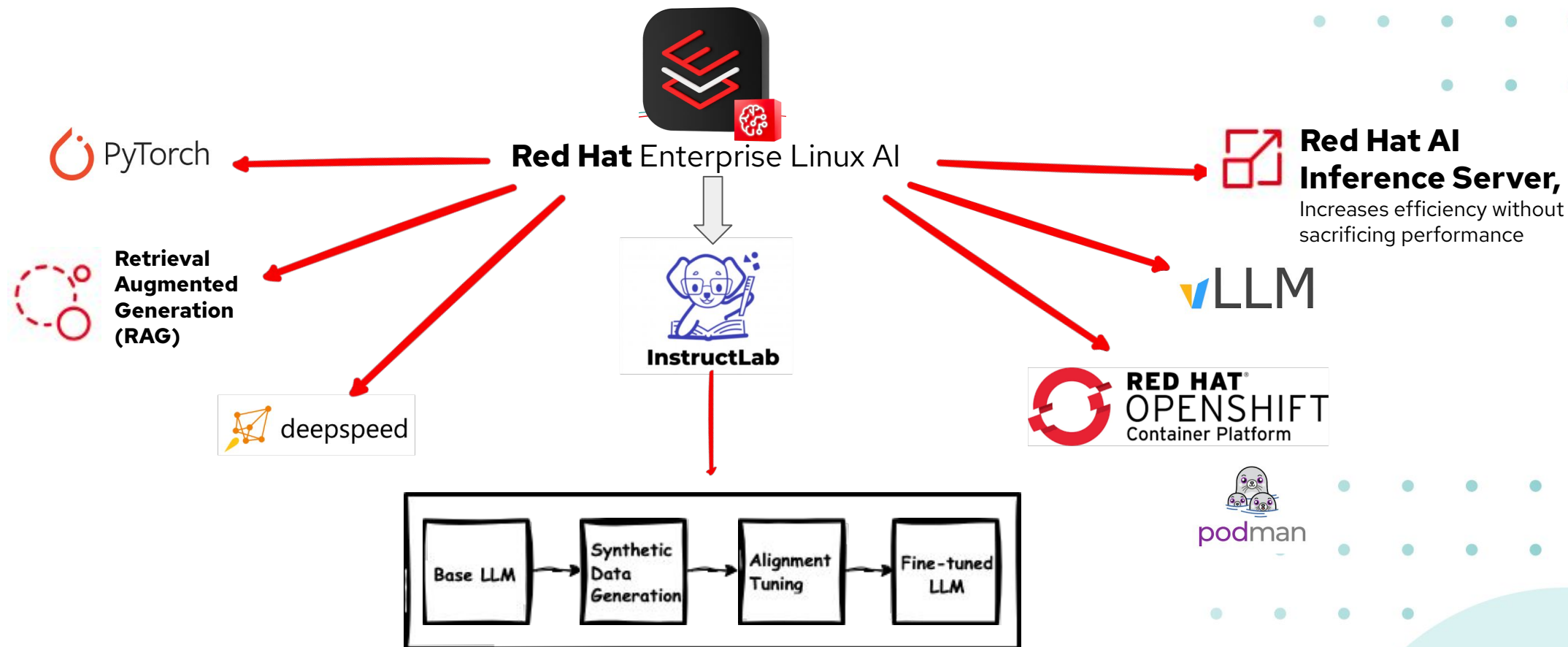
Senior Architect
Red Hat



Problem Statement

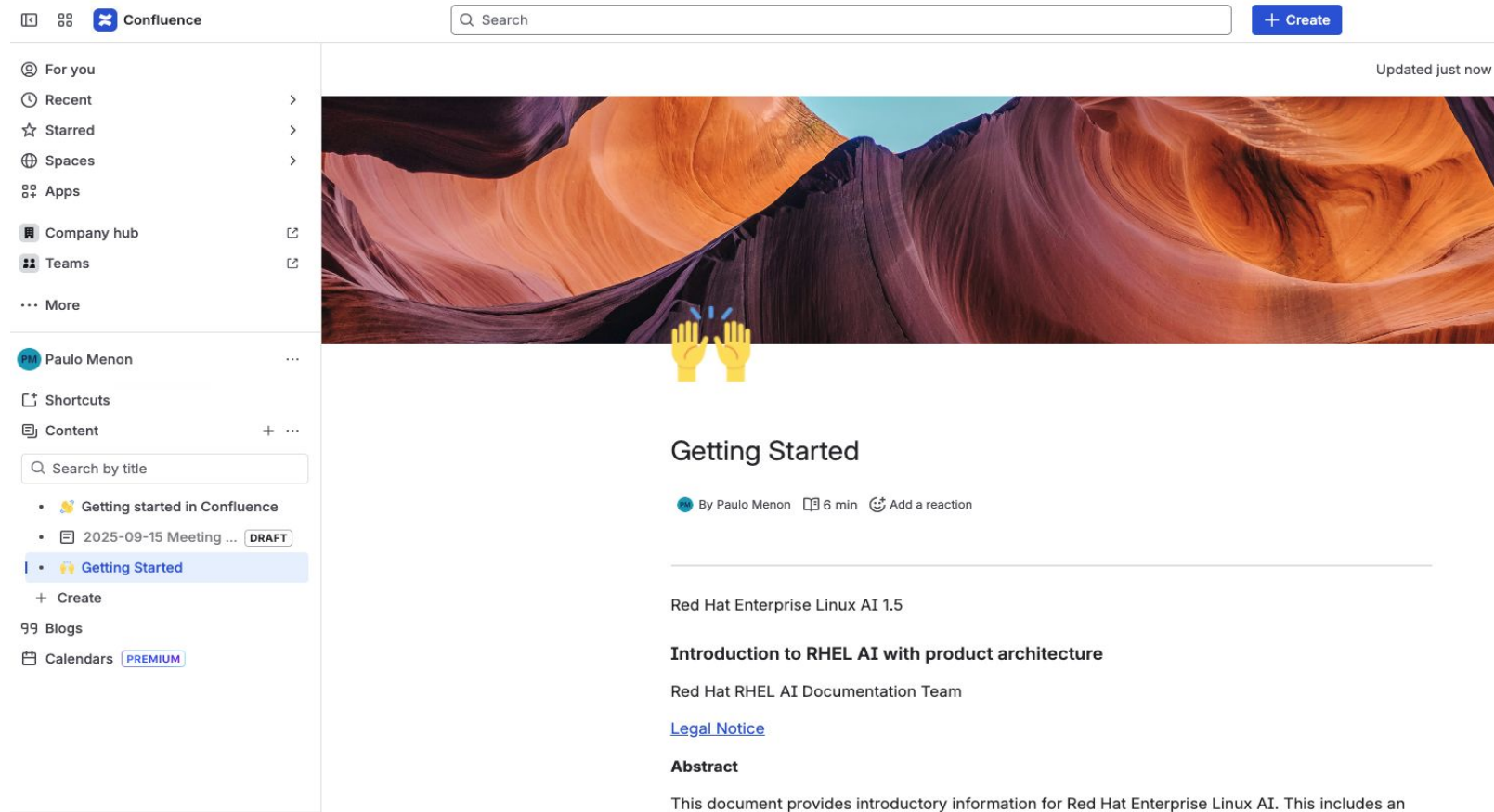


What is Red Hat Enterprise Linux AI?



Collecting Knowledge Base

Where the data for this demo come from?



Confluence

Q Search

+ Create

Updated just now

For you

Recent

Starred

Spaces

Apps

Company hub

Teams

More

PM Paulo Menon

Shortcuts

Content

Search by title

- Getting started in Confluence
- 2025-09-15 Meeting ... DRAFT
- Getting Started**

+ Create

Blogs

Calendars PREMIUM

clapping hands

Getting Started

By Paulo Menon 6 min Add a reaction

Red Hat Enterprise Linux AI 1.5

Introduction to RHEL AI with product architecture

Red Hat RHEL AI Documentation Team

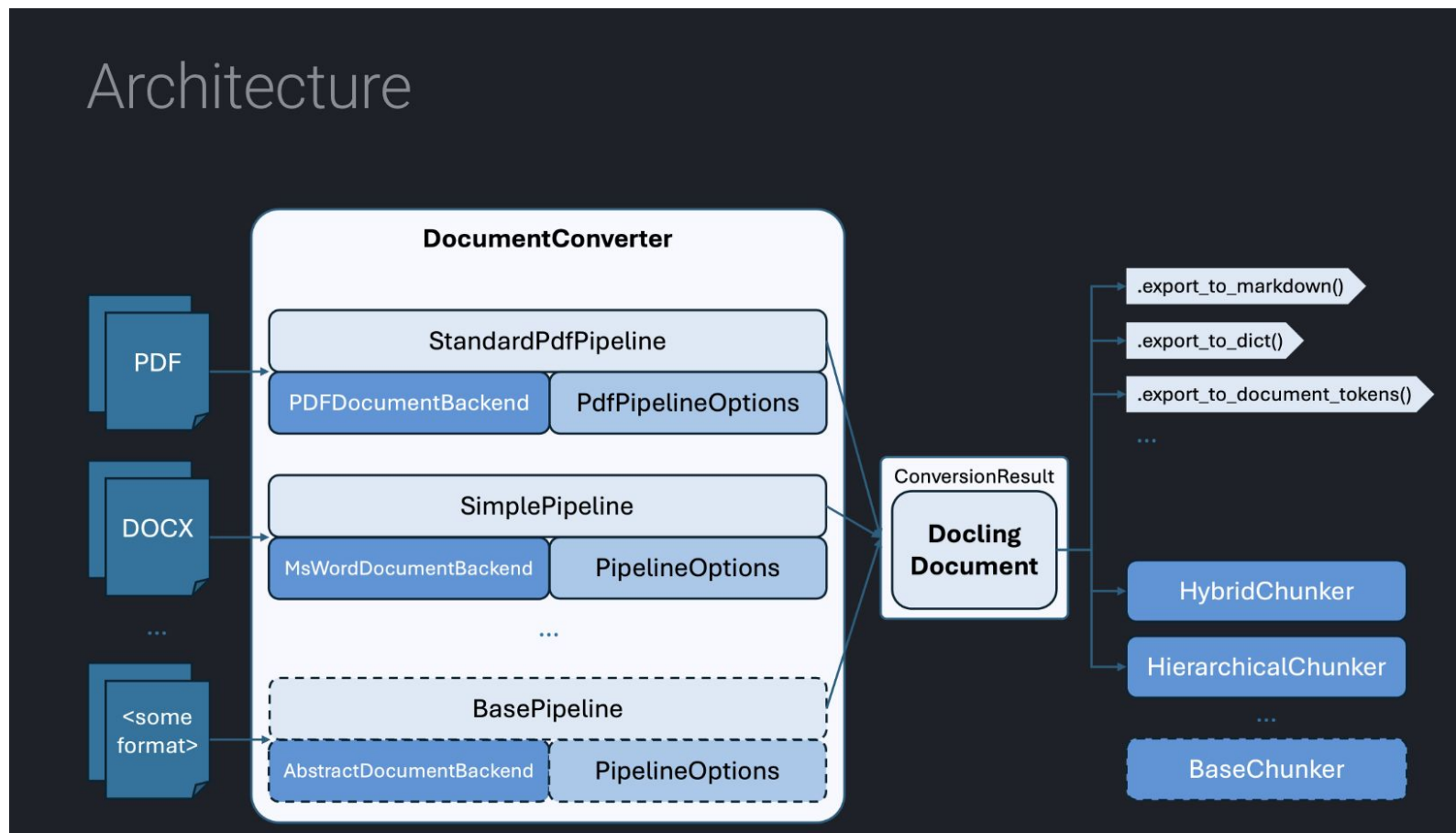
[Legal Notice](#)

Abstract

This document provides introductory information for Red Hat Enterprise Linux AI. This includes an

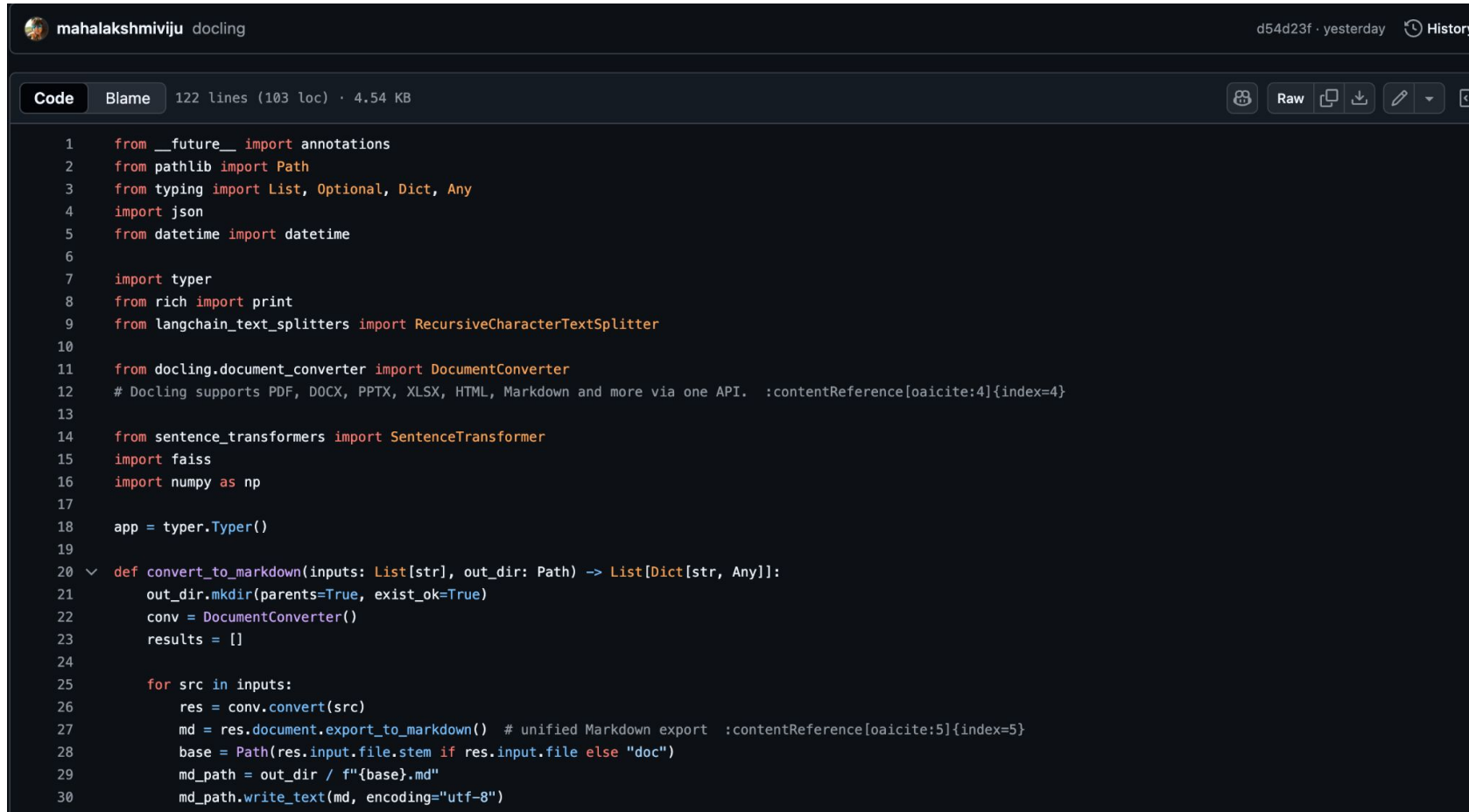
Collecting Knowledge Base

Web scraping using Docling



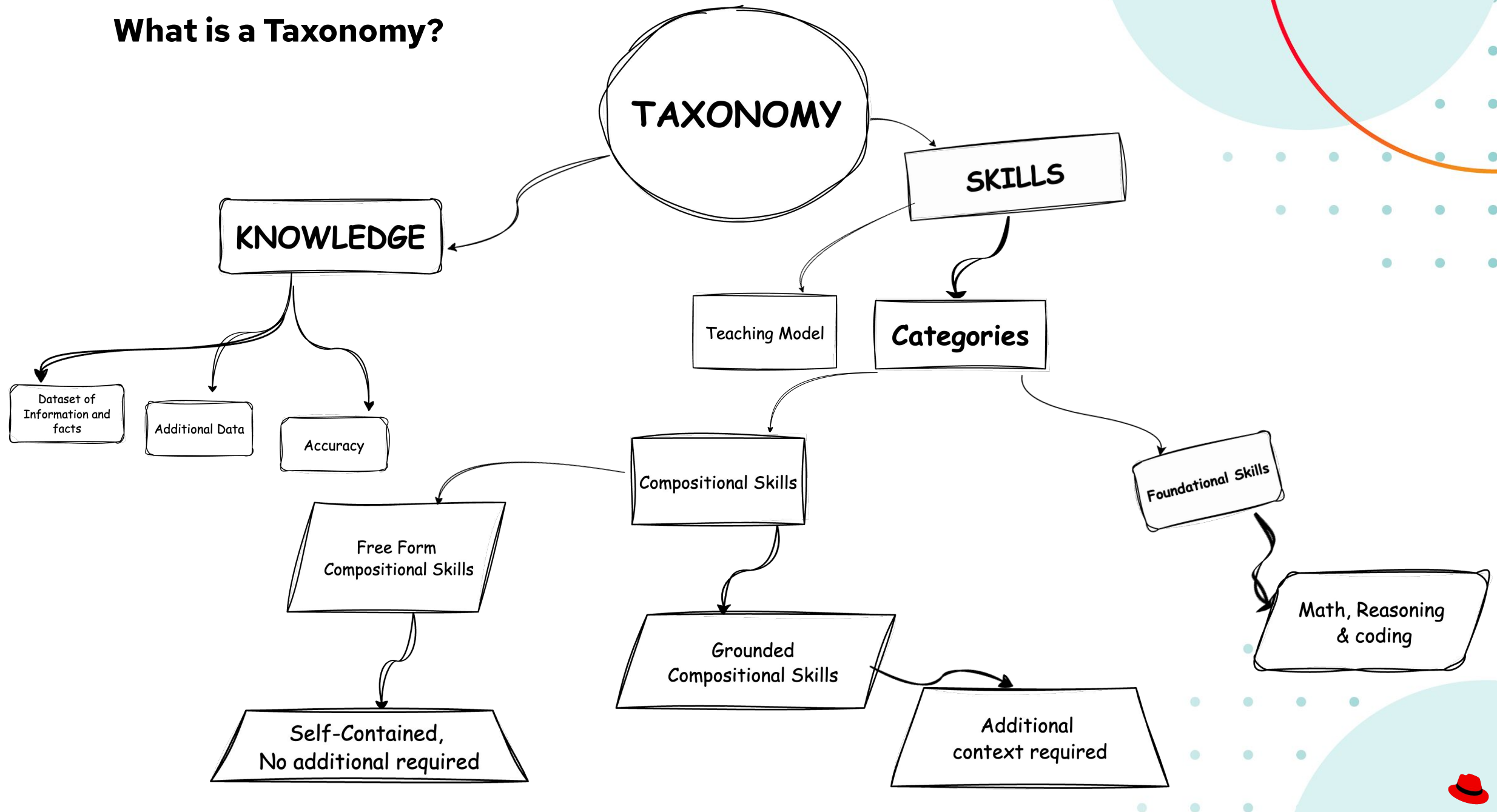
Collecting Knowledge Base

Web scraping using Docling



```
1  from __future__ import annotations
2  from pathlib import Path
3  from typing import List, Optional, Dict, Any
4  import json
5  from datetime import datetime
6
7  import typer
8  from rich import print
9  from langchain_text_splitters import RecursiveCharacterTextSplitter
10
11  from docling.document_converter import DocumentConverter
12  # Docling supports PDF, DOCX, PPTX, XLSX, HTML, Markdown and more via one API. :contentReference[oaicite:4]{index=4}
13
14  from sentence_transformers import SentenceTransformer
15  import faiss
16  import numpy as np
17
18  app = typer.Typer()
19
20  def convert_to_markdown(inputs: List[str], out_dir: Path) -> List[Dict[str, Any]]:
21      out_dir.mkdir(parents=True, exist_ok=True)
22      conv = DocumentConverter()
23      results = []
24
25      for src in inputs:
26          res = conv.convert(src)
27          md = res.document.export_to_markdown() # unified Markdown export :contentReference[oaicite:5]{index=5}
28          base = Path(res.input.file.stem if res.input.file else "doc")
29          md_path = out_dir / f"{base}.md"
30          md_path.write_text(md, encoding="utf-8")
```

What is a Taxonomy?



How the Fine-tuning process flow works?



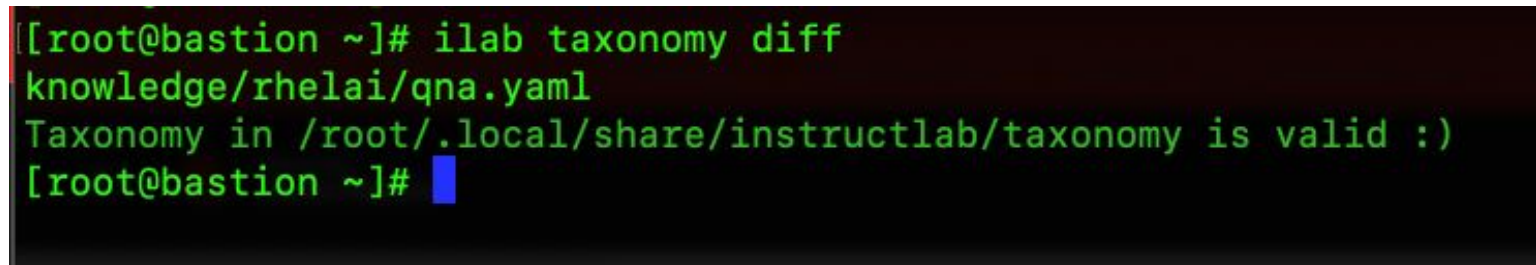
Validating using Taxonomy



mahalakshmiyiju Update qna.yaml - RHEL AI 24b012b · 2 days ago History

Code Blame 169 lines (169 loc) · 19.1 KB

```
1 version: 3
2 domain: technology
3 document_outline: |
4   This document provides introductory information for Red Hat Enterprise Linux AI. This includes an overview of RHEL AI and the product architecture
5 created_by: maha
6 seed_examples:
7   - context: |
8     Red Hat Enterprise Linux AI (RHEL AI) is a specialized platform designed to facilitate the development of enterprise applications using open-source Large Language Models (LLMs).
9     questions_and_answers:
10      - question: |
11        What is Red Hat Enterprise Linux AI (RHEL AI)?
12        answer: |
13          RHEL AI (Red Hat Enterprise Linux AI) is a platform designed for developing enterprise applications using open-source Large Language Models (LLMs). It's built u
14      - question: |
15        What are the main capabilities of RHEL AI?
16        answer: |
17          RHEL AI offers several key capabilities: it allows you to host an LLM and interact with the open-source Granite family of LLMs; it enables the creation and addi
18      - question: |
19        How does RHEL AI relate to open-source LLMs?
20        answer: |
21          RHEL AI is fundamentally designed to work with open-source LLMs, allowing users to host and interact with models from families like Granite. It emphasizes an op
22      - question: |
23        What is the significance of the "LAB method" in RHEL AI?
24        answer: |
25          The LAB method is a key feature that simplifies the process of adding custom knowledge or skills data. It allows users to contribute their own data through a G
26      - question: |
27        Can users with limited machine learning experience utilize RHEL AI?
28        answer: |
```



```
[root@bastion ~]# ilab taxonomy diff
knowledge/rhelai/qna.yaml
Taxonomy in /root/.local/share/instructlab/taxonomy is valid :)
[root@bastion ~]#
```



Generating Synthetic Data

```
$ ilab data generate --taxonomy-path /root/.local/share/instructlab/taxonomy/knowledge/rhelai/qna.yaml
```

```
INFO 2025-09-17 11:37:14,301 instructlab.sdg.datamixing:774: Knowledge detected to be less than 3.00% of skills (0.41%), upsampling to: 11824
Creating json from Arrow format: 100%|#####| 2/2 [00:00<00:00, 63.98ba/s]
INFO 2025-09-17 11:37:14,486 instructlab.sdg.datamixing:158: Loading dataset from /usr/share/instructlab/sgd/datasets/skills.jsonl ...
INFO 2025-09-17 11:37:30,655 instructlab.sdg.datamixing:160: Dataset columns: ['messages', 'metadata', 'id']
INFO 2025-09-17 11:37:30,655 instructlab.sdg.datamixing:161: Dataset loaded with 394141 samples
INFO 2025-09-17 11:37:55,576 instructlab.sdg.datamixing:158: Loading dataset from /root/.local/share/instructlab/datasets/2025-09-17_053215/node_dataset_2025-09-17T05_32_47/knowledge_rhelai_p10.jsonl ...
Generating train split: 1617 examples [00:00, 60151.92 examples/s]
INFO 2025-09-17 11:37:55,669 instructlab.sdg.datamixing:160: Dataset columns: ['messages', 'metadata', 'id', 'unmask']
INFO 2025-09-17 11:37:55,669 instructlab.sdg.datamixing:161: Dataset loaded with 1617 samples
INFO 2025-09-17 11:37:55,669 instructlab.sdg.datamixing:44: Rebalancing dataset to have 11824 samples ...
Map (num_proc=8): 100%|#####| 11824/11824 [00:05<00:00, 2240.17 examples/s]
Map (num_proc=8): 100%|#####| 405965/405965 [00:48<00:00, 8438.54 examples/s]
Creating json from Arrow format: 100%|#####| 406/406 [01:23<00:00, 4.85ba/s]
INFO 2025-09-17 11:40:14,381 instructlab.sdg.datamixing:235: Mixed Dataset saved to /root/.local/share/instructlab/datasets/2025-09-17_053215/skills_train_msgs_2025-09-17T05_32_47.jsonl
INFO 2025-09-17 11:40:14,600 instructlab.sdg.datamixing:158: Loading dataset from /root/.local/share/instructlab/datasets/2025-09-17_053215/node_dataset_2025-09-17T05_32_47/knowledge_rhelai_p07.jsonl ...
Generating train split: 823 examples [00:00, 24121.70 examples/s]
INFO 2025-09-17 11:40:14,710 instructlab.sdg.datamixing:160: Dataset columns: ['messages', 'metadata', 'id', 'unmask']
INFO 2025-09-17 11:40:14,710 instructlab.sdg.datamixing:161: Dataset loaded with 823 samples
Map (num_proc=8): 100%|#####| 823/823 [00:00<00:00, 3157.26 examples/s]
Map (num_proc=8): 100%|#####| 823/823 [00:00<00:00, 3224.53 examples/s]
Creating json from Arrow format: 100%|#####| 1/1 [00:00<00:00, 52.55ba/s]
INFO 2025-09-17 11:40:16,101 instructlab.sdg.datamixing:235: Mixed Dataset saved to /root/.local/share/instructlab/datasets/2025-09-17_053215/knowledge_train_msgs_2025-09-17T05_32_47.jsonl
INFO 2025-09-17 11:40:16,101 instructlab.sdg.generate_data:757: Generation took 22048.53s
C(0J6~) Data generate completed successfully! C(0J6~)
[root@bastion rhelai]#
```



Training

```
ilab model train --pipeline full --device cpu --data-path
~/.local/share/instructlab/datasets/2025-09-17_053215/skills_train_msgs_2025-0
9-17T05_32_47.jsonl
```

[illegible]

Serving the new trained Model

```
[root@bastion checkpoints]# ilab model serve --model-path /root/.local/share/instructlab/checkpoints/ggml-model-f16.gguf
INFO 2025-09-24 13:53:39,766 instructlab.model.serve_backend:80: Setting backend_type in the serve config to llama-cpp
INFO 2025-09-24 13:53:39,782 instructlab.model.serve_backend:86: Using model '/root/.local/share/instructlab/checkpoints/ggml-model-f16.gguf' with -1 gpu-layers and 4096 max context size.
INFO 2025-09-24 13:54:40,467 instructlab.model.backends.llama_cpp:306: Replacing chat template:
    {% set eos_token = "<|endoftext|>" %}
    {% set bos_token = "<|begginingoftext|>" %}
    {% for message in messages %}{% if message['role'] == 'pretraining' %}{{'<|pretrain|>' + message['content'] + '<|endoftext|>' + '<|/pretrain|>' }}{% elif message['role'] == 'system' %}{{'<|system|>' + ' + message['content'] + ' }}{% elif message['role'] == 'user' %}{{'<|user|>' + ' + message['content'] + ' }}{% elif message['role'] == 'assistant' %}{{'<|assistant|>' + ' + message['content'] + '<|endoftext|>' + ('' if loop.last else ' ')}}{% endif %}{% if loop.last and add_generation_prompt %}{{ ' <|assistant|>' + ' ' }}{% endif %}{% endfor %}
INFO 2025-09-24 13:54:40,493 instructlab.model.backends.llama_cpp:233: Starting server process, press CTRL+C to shutdown server...
INFO 2025-09-24 13:54:40,493 instructlab.model.backends.llama_cpp:234: After application startup complete see http://127.0.0.1:8000/docs for API.
```


Chatting with the new trained model

```
[root@bastion ~]# ilab model chat -m /root/.local/share/instructlab/checkpoints/ggml-model-f16.gguf
```

```
system
Welcome to InstructLab Chat w/ GGML-MODEL-F16.GGUF (type /h for help)
```

```
>>> What is "serving a model" and "inference" in the context of LLMs?
```

```
[S][default]
```

```
ggml-model-f16.gguf
```

Serving a model refers to making an LLM available for use, while inference is the actual process of using the model to generate responses based on input data. In other words, serving is about making the model accessible, and inference is about utilizing it to produce outputs. Both serve different purposes but are closely related in the context of LLMs.

For example, when you build an LLM from scratch or fine-tune it using a teacher model and synthetic data, you serve the model by making it available for use. Once the model is serving, you can then perform inference to generate responses based on user-provided input data. The results of these inferences are known as "prompts" in Red Hat InstructLab.

In summary, serving a model involves making an LLM available for use, while inference is the actual process of using the model to generate responses based on input data. Both are important aspects of working with LLMs in the context of RHEL 9.4 and Red Hat InstructLab.

```
elapsed 11.704 seconds
```

```
>>> █
```

```
[S][default]
```


Demo time!



```
[root@bastion ~]#
```

```
I
```

Demo

Demo

After SDG - How data is trained

Docling code,
Qna and Md file
Taxonomy
SDG
Train
Chat



Concluding our session

RHEL AI

Taxonomy

**Data
collection**

**Synthetic
Data**

Fine-tuning



Concluding our session



Reduces time and
cost

Reduces Resource
usage

Simplifies Data
preparation

Reduces AI
Complexity

What is next?



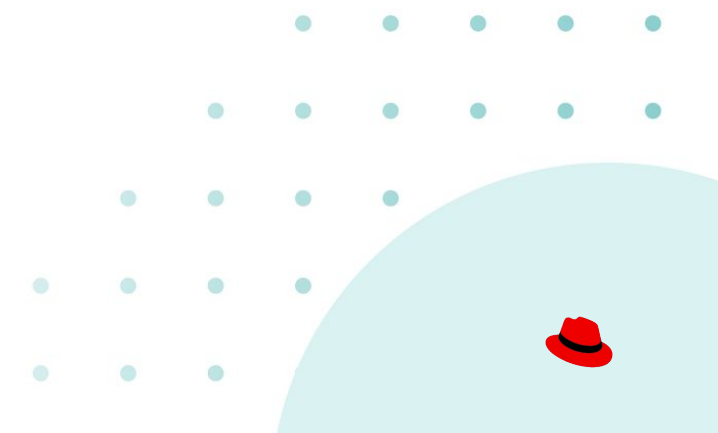
Red Hat
Enterprise Linux AI

+



Red Hat
OpenShift AI

Q & A





Connect

Thank you



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat

