Alfresco Tech Talk Live #154
Agenda

• Community news
• Enhancing Document-Centric Features with On-Premise Generative AI for Alfresco Community
Happy New Year, humans

NightCafe

Human made
Meet a Hyland Developer Evangelist at FOSDEM!

Using Generative AI and Content Service Platforms together

Join us to learn from the integration of Generative AI (GenAI Stack) with a Content Service Platform ( Alfresco).

Leveraging Open-Source platforms, we will explore different applications of Generative AI within Document Management ecosystem. From enabling conversational searching to generating concise summaries and facilitating intelligent categorization, this session aims to showcase practical approaches that combines the potential of Generative AI and Content Service Platforms.

Live demo and sample source code will be provided!

Join us to discover how this dynamic collaboration enhances document processing, retrieval, and organization, opening new avenues for efficient and intelligent information management.

Speakers

Angel Boroy

Attachments

* Alfresco GenAI Stack (slides)
Meetups, meetups, meetups… and a DevCon?

Initiatives:
- Local User Groups with CSMs
- Meetups (Online, In Person)
- Chapters on events.hyland.com

We want to check if it makes sense to organize DevCon again.
Resources

Alfresco

Deploying Alfresco Outlook Transform Engine with Docker Compose @ GitHub

Script Object for RenditionService2 @ GitHub and thanks to @hi-ko

Alfresco GenAI @ GitHub

Resources to come

• Adapting your logging configuration to log4jv2
• How to set up messages for increased reliability
• Upgrading your addons to Jakarta EE 10 and Spring 6
• Upgrading to Apache Tomcat 10
• Using Control Center App with Community Edition
• Deploying Alfresco with Helm in Kubernetes
Collaboration

Blog posts

• Summarization of textual content in Alfresco repository with Amazon Bedrock by @abhinavmishra14

Contributions

• https://github.com/aborroy/alfresco-dockerx-builder/issues/11 by @uvukasinovic
• https://github.com/Alfresco/alfresco-docker-installer/issues/185 by @luca86r
• https://github.com/Alfresco/alfresco-docker-installer/issues/186 by @N00BTellaBrot

Conferences

• FOSDEM 2024, 3 & 4 February, Brussels
TTL Speakers wanted!

• Take the opportunity to showcase your work with the community
• About Alfresco, Nuxeo, and associated technologies
• Best practices, integration, scaling, cloud, …
• In your native language
Today’s talk
Enhancing Document-Centric Features with On-Premise Generative AI for Alfresco Community

Angel Borroy
Tech Evangelist at Hyland
Enhancing Document-Centric Features with On-Premise Generative AI for Alfresco Community

Alfresco Tech Talk Live #154

Angel Borroy
Developer Evangelist
Agenda

- Docker GenAI Stack
- Alfresco Integration
- Document Features
  - Existing Content
  - New Content
- What Else?
Docker GenAI Stack

Components

**ollama**
- Local management of *open source* LLMs
- Catalog of preconfigured LLMs, such as Llama2 or Mistral

**neo4j**
- Graph and native vector search capabilities
- Ground LLMs for more precise GenAI predictions and outcomes

**LangChain**
- Communication between the LLM, your application, and the database
- Python framework for developing applications powered by LLMs

[https://github.com/docker/genai-stack](https://github.com/docker/genai-stack)
Docker GenAI

Deployment

[Diagram showing Docker deployment with pull-model, ollama, neo4j, streamlit, LangChain, and REST API connected with arrows.]

https://github.com/docker/genai-stack
Docker GenAI

LLM 3 Billion Parameters

<table>
<thead>
<tr>
<th>Model</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>phi</td>
<td>MIT License</td>
</tr>
<tr>
<td>dolphin-phi</td>
<td>MIT License</td>
</tr>
<tr>
<td>orca-mini</td>
<td>cc-by-nc-sa-4.0</td>
</tr>
<tr>
<td>deepseek-coder</td>
<td>DEEPSEEK LICENSE AGREEMENT</td>
</tr>
<tr>
<td>Model</td>
<td>License</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>llama2</td>
<td>LLAMA2 COMMUNITY LICENSE AGREEMENT</td>
</tr>
<tr>
<td>codellama</td>
<td>LLAMA2 COMMUNITY LICENSE AGREEMENT</td>
</tr>
<tr>
<td>vicuna</td>
<td>LLAMA2 COMMUNITY LICENSE AGREEMENT</td>
</tr>
<tr>
<td>mistral</td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>mistral-openorca</td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>llava</td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>orca-mini</td>
<td>cc-by-nc-sa-4.0</td>
</tr>
<tr>
<td>deepeasek-coder</td>
<td>DEEPSEEK LICENSE AGREEMENT</td>
</tr>
<tr>
<td>orca2</td>
<td>MICROSOFT RESEARCH LICENSE</td>
</tr>
<tr>
<td>falcon</td>
<td>FALCON 180B TII LICENSE 1.0</td>
</tr>
</tbody>
</table>

RAM: 8 GB

Size: 8 GB
LLM 13 Billion Parameters

- llama2: LLAMA2 COMMUNITY LICENSE AGREEMENT
- codellama: LLAMA2 COMMUNITY LICENSE AGREEMENT
- vicuna: LLAMA2 COMMUNITY LICENSE AGREEMENT
- orca-mini: cc-by-nc-sa-4.0
- llava: Apache License 2.0
- orca2: MICROSOFT RESEARCH LICENSE
## LLM 13+ Billion Parameters

<table>
<thead>
<tr>
<th>Model</th>
<th>License Agreement</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>llama2</td>
<td>LLAMA2 COMMUNITY LICENSE AGREEMENT</td>
<td>70B</td>
</tr>
<tr>
<td>codellama</td>
<td>LLAMA2 COMMUNITY LICENSE AGREEMENT</td>
<td>34B</td>
</tr>
<tr>
<td>mixtral</td>
<td>Apache License 2.0</td>
<td>56B</td>
</tr>
<tr>
<td>deepseek-coder</td>
<td>DEEPSEEK LICENSE AGREEMENT</td>
<td>33B</td>
</tr>
<tr>
<td>vicuna</td>
<td>LLAMA2 COMMUNITY LICENSE AGREEMENT</td>
<td>33B</td>
</tr>
<tr>
<td>falcon</td>
<td>FALCON 180B TII LICENSE 1.0</td>
<td>40B / 180B</td>
</tr>
</tbody>
</table>
Creating a new REST API

```python
import streamlit as st
from langchain.chains import RetrievalQA
from langchain.vectorstores.neo4j_vector import Neo4jVector
from fastapi import FastAPI, UploadFile

app = FastAPI()

@app.post("/document")
    vector_store = Neo4jVector.from_texts(...)
    qa = RetrievalQA.from_chain_type(
        llm=llm, chain_type="stuff", retriever=vector_store.as_retriever()
    )
    stream_handler = StreamHandler(st.empty())
    term = qa.run("Hello", callbacks=[stream_handler])
```
Alfresco Integration
Alfresco Integration

Deployment

https://github.com/aborroy/alfresco-genai

ALFRESCO
- model-repo
- model-share

GENAI STACK
- llm
- neo4j
- langchain

Compose

Docker Network

app
alfresco-ai-applier
service
alfresco-ai-listener

http://alfresco:8080
http://genai:8506/summary
http://genai:8506/classify
http://genai:8506/prompt

tcp://activemq:61616

https://github.com/aborroy/alfresco-genai
Alfresco Integration

Alfresco Aspects

**summarizable**
- summary text
- tags text
- llmSummary text

**promptable**
- question text
- answer text
- llmPrompt text

**classifiable**
- terms text, multiple

- term text
- llmClassify text

**tags** are identified by LLM without restrictions
**term** is selected by LLM from a list of **terms**
llm* includes the LLM name used
Alfresco Integration

AI Applier

Run action on a Repository Folder

- Summary
- Classify
  - List of terms required as parameter

$ java -jar alfresco-ai-applier-0.8.0.jar \
applier.root.folder=/app:company_home/app:shared \
applier.action=SUMMARY

$ java -jar alfresco-ai-applier-0.8.0.jar \
applier.root.folder=/app:company_home/app:shared \
applier.action.classify.term.list=English,Spanish \
applier.action=CLASSIFY
Alfresco Integration

AI Listener
Listen to ActiveMQ for aspect settings
- Summary
- Classification
- Prompting

```
alfresco-ai-listener:
    image: alfresco-ai-listener
    environment:
        CONTENT_SERVICE_URL: "http://alfresco:8080"
        SPRING_ACTIVEMQ_BROKERURL: "tcp://activemq:61616"
        GENAI_URL: "http://genai:8506"
```
Alfresco Integration

**GenAI App**

Python application that provides REST API endpoints

---

**Document Loader**
Split document into smaller chunks

**Store in VectorDB**
Embed and save chunks

**Create QA Chain**
With the ability to lookup Vector DB

---

**LLM Embeddings**

**LLM**

---

**SUMMARY_LANGUAGE=English**
**SUMMARY_SIZE=120**
**TAGS_NUMBER=3**
curl --location 'http://localhost:8506/summary' --form 'file=@"./file.pdf"'
{
    "summary": "The text discusses...",
    "tags": "Golang, Merkle, Difficulty",
    "model": "mistral"
}

SUMMARY Write a short summary of the text in 120 words only in ENGLISH

TAGS Provide 3 words to categorize the document in language ENGLISH in a single line. Use only language ENGLISH for these 3 words in the answer. Don't add any explanation for the words in the answer. Don't add any note after the list of words in the answer. Don't use bullets or numbers to list the words in the answer. Don't add in the answer the translation of the words in a different language after the list of words. Give the answer exactly as a list of 3 words in language ENGLISH separated with comma and without ending dot.
TERM

Pick one of the following list of categories: Japanese, Spanish, Vietnamese.
Write the answer only in ENGLISH language.
Don't add any explanation for the choice in the answer.
Don't add any note after the word in the answer.
Don't add any space before the word in the answer.
Don't add in the answer the translation of the word in a different language after chosen word.
Give the answer exactly as a single word from the list.

curl --location
--form 'file=./file.pdf'
{
    "term": "English",
    "model": "mistral"
}
curl --location 'http://localhost:8506/prompt?prompt="What is this text about?"' \
--form 'file=./file.pdf'
{
    "answer": "Yes, it is difficult to find childcare in Tokyo.",
    "model": "mistral"
}
Document Features
Existing Content

- Summary
- Classify

REST API

alfresco-ai-applier

Scan a SOURCE FOLDER

MISTRAL AI_

Apache License 2.0

https://github.com/aborroy/alfresco-genai
Demo Time
New Content

```java
document.move(
    document.parent.childByNamePath(
        document.properties["genai:term"]
    )
);
```

Apache License 2.0

https://github.com/aborroy/alfresco-genai
Demo Time

c5a.8xlarge
32 vCPUs
64 GiB RAM
GPU
What Else?
Find your way

Docker AI/ML Hack-a-thon

- Readme AI
- Techdocs
- Docker Image Analyzer
- Docker Log Sentiment Analyzer
- GitChats AI

ollama alternatives

- [https://gpt4all.io/index.html](https://gpt4all.io/index.html)
- [https://www.secondstate.io/run-llm](https://www.secondstate.io/run-llm)
- [https://huggingface.co/docs/hub/spaces-sdks-docker-first-demo](https://huggingface.co/docs/hub/spaces-sdks-docker-first-demo)
Thanks!