


AMOS WS23/24 Project 5

Pipeline Configuration Chat AI

Project name	Pipeline Configuration Chat AI
Project mission	In collaboration with Shell and under the AMOS framework, is to develop a cutting-edge tool, user-friendly, that seamlessly blends chat AI technology with the robust data processing capabilities of RTDIP. The project is dedicated to creating a platform where users can effortlessly configure data pipelines using a conversational interface.
Industry partner	Shell
Team logo	
Project summary	<p>RTDIP is an open-source framework built on Apache Spark for real-time data ingestion. Data engineers use it to construct pipelines daily for extraction, transformation, and loading (ETL) tasks.</p> <p>To enhance this process, we developed Chatbot, integrating technologies such as LangChain, Streamlit, ChromaDB, and OpenAI. These are the main features of the ChatRTDIP:</p> <ol style="list-style-type: none">1. <u>Query Processing</u>: Queries are converted into embeddings for semantic search in the content hub.2. <u>Semantic Search</u>: The content hub is searched for relevant files, which are then marked for later retrieval.3. <u>Response Generation</u>: A Language Model (LLM) uses both the original query and retrieved file information to generate a response.4. <u>API Key Management</u>: The system ensures the validity of API keys. A dynamic API key selection component allows users to input or select existing keys, new keys can be saved for future use.5. <u>Model Update Mechanism</u>: The chatbot includes an efficient update mechanism to keep the model current. By clicking a button, the system connects to the RTDIP GitHub repository, selectively, updating content with the latest sources, destinations, and transformers. This process is designed to be swift and non-disruptive, ensuring the platform remains responsive and knowledgeable. <p>Using an update approach, the content store is refreshed with the newest sources, destinations, and transformers from the core repository. This method involves creating a temporary workspace, identifying necessary directories, and executing a checkout to retrieve only relevant parts of the codebase without unnecessary files. Once retrieved, the content seamlessly integrates into the local system, ensuring the model's knowledge remains up-to-date. This update process is designed to be swift and non-disruptive, preserving the platform's responsiveness and knowledgeability.</p> <p>The Content Store Extractor is key in the Retrieval Augmented Generation (RAG) system, helping in combining knowledge and generating accurate responses. After each bot-generated message, users have access to a "New Conversation" button, allowing them to clear their screen and start all over again.</p>

Project illustration

```
I would like to use RTDIP components to read from SparkDeltaSource, transform using
PandasToPySparkTransformer, then write to SparkEventhubDestination, return python code

Here is the python code to use RTDIP components to read from SparkDeltaSource, transform
using PandasToPySparkTransformer, and then write to SparkEventhubDestination:

from rtdip_sdk.pipelines.sources import SparkDeltaSource
from rtdip_sdk.pipelines.destinations import SparkEventhubDestination
from rtdip_sdk.pipelines.transformers import PandasToPySparkTransformer
from rtdip_sdk.pipelines.utilities import SparkSessionUtility

# Not required if using Databricks
spark = SparkSessionUtility(config={}).execute()

# Read from SparkDeltaSource
delta_source = SparkDeltaSource(
    spark=spark,
    path="DELTA-TABLE-PATH",
    version=None
)

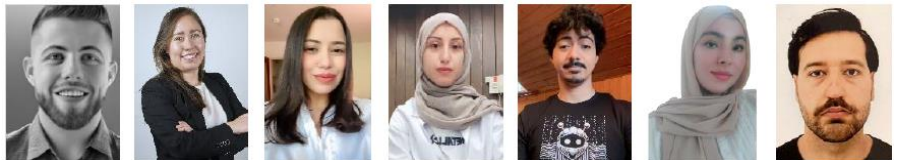
df = delta_source.read_batch()

# Transform using PandasToPySparkTransformer
transformer = PandasToPySparkTransformer(df)

df_transformed = transformer.transform()

# Write to SparkEventhubDestination
eventhub_destination = SparkEventhubDestination(
    spark=spark,
    data=df_transformed,
    options={
        "file_format": "parquet",
        "compression": "gzip",
        "overwrite": True
    }
)
```

Team photo



Project repository

<https://github.com/amosproj/amos2023ws05-pipeline-config-chat-ai>