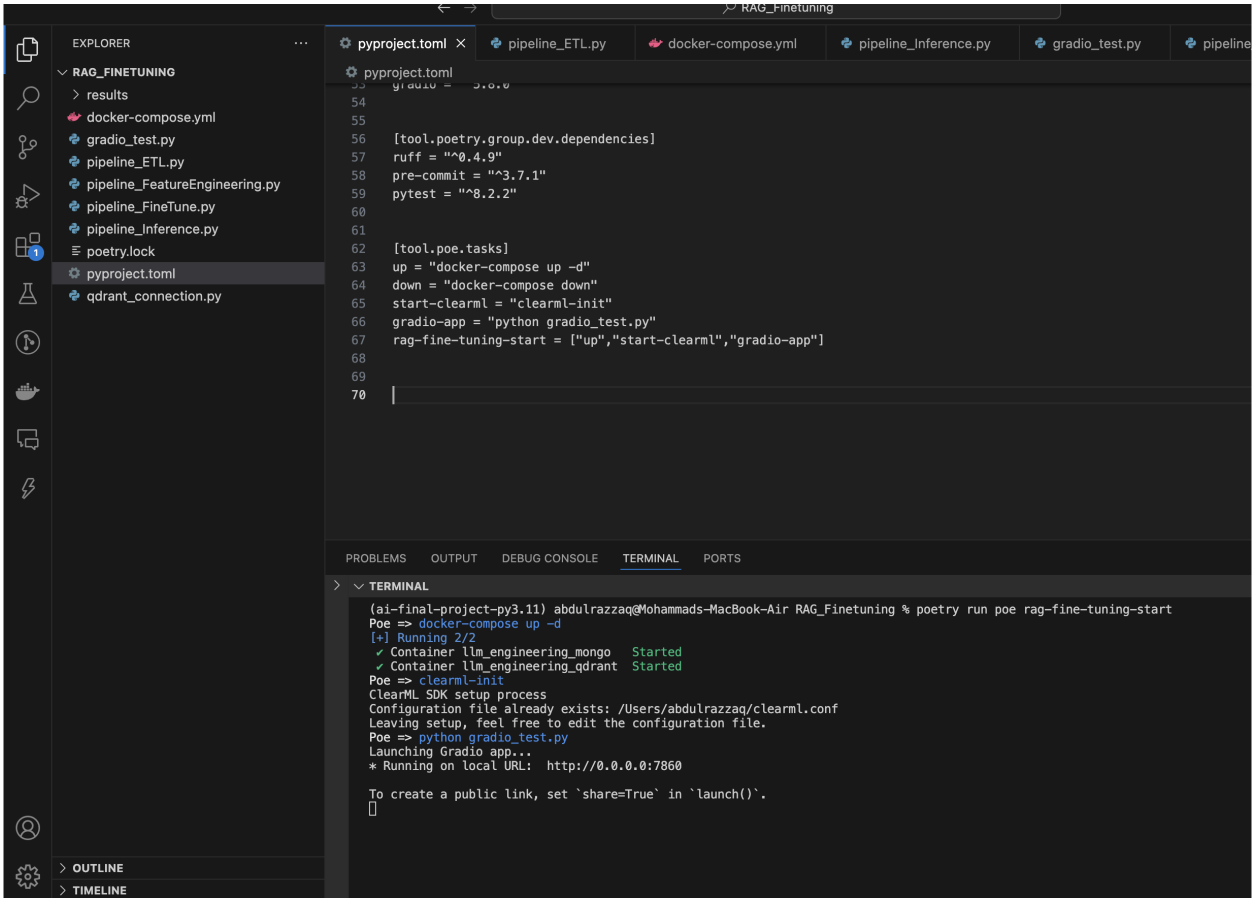
Finetuned RAG Systems Engineering Report

Step 1: Infrastucture

- **Goal**: Set up your environment, including Qdrant for vector storage, MongoDB for raw data storage, Gradio App , ClearML.
- **Completed Tasks**:
    - Infrastructure appears ready, including Qdrant and MongoDB.

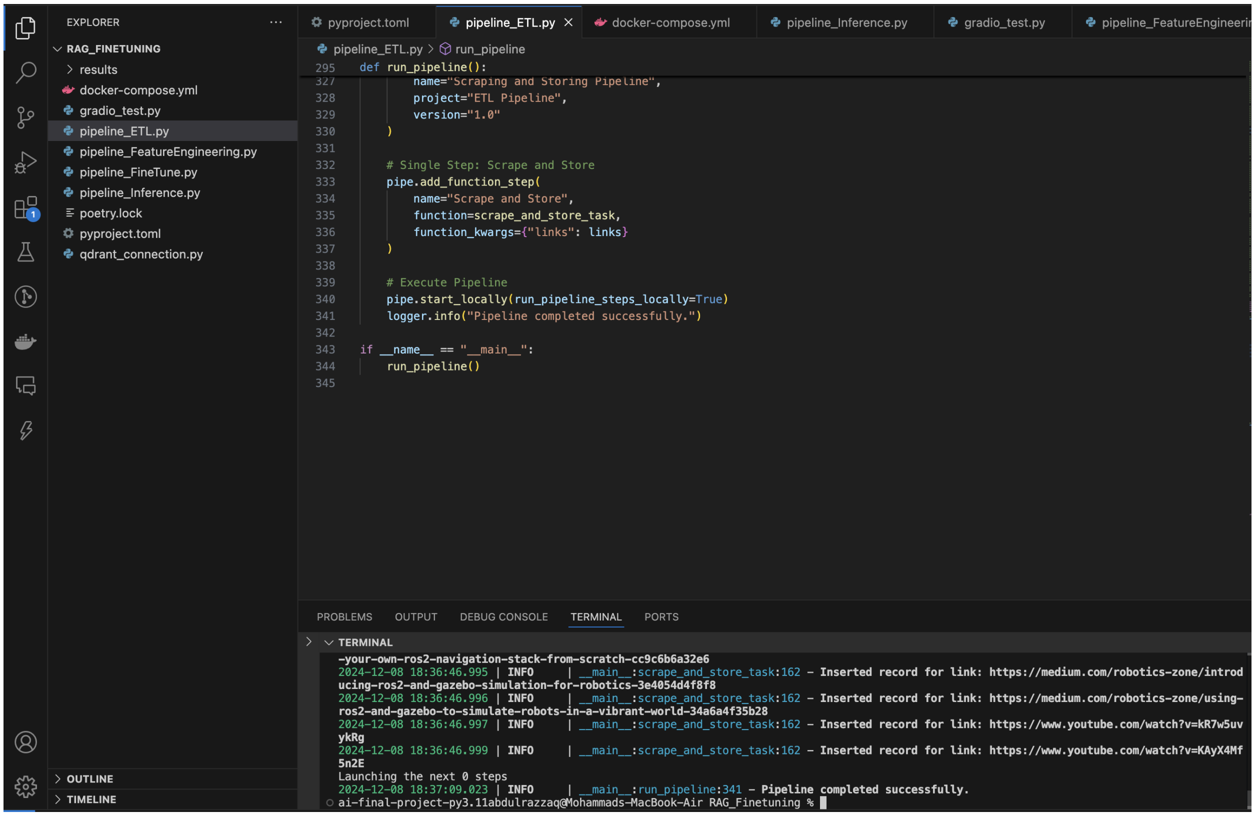


```
gradio = "5.8.0"


[tool.poetry.group.dev.dependencies]
ruff = "^0.4.9"
pre-commit = "^3.7.1"
pytest = "^8.2.2"


[tool.poe.tasks]
up = "docker-compose up -d"
down = "docker-compose down"
start-clearml = "clearml-init"
gradio-app = "python gradio_test.py"
rag-fine-tuning-start = ["up","start-clearml","gradio-app"]
```

```
(ai-final-project-py3.11) abdulrazzaq@Mohammads-MacBook-Air RAG_Finetuning % poetry run poe rag-fine-tuning-start
Poe => docker-compose up -d
[+] Running 2/2
 ✔ Container llm_engineering_mongo   Started
 ✔ Container llm_engineering_qdrant  Started
Poe => clearml-init
ClearML SDK setup process
Configuration file already exists: /Users/abdulrazzaq/clearml.conf
Leaving setup, feel free to edit the configuration file.
Poe => python gradio_test.py
Launching Gradio app...
* Running on local URL:  http://0.0.0.0:7860

To create a public link, set `share=True` in `launch()`.
```

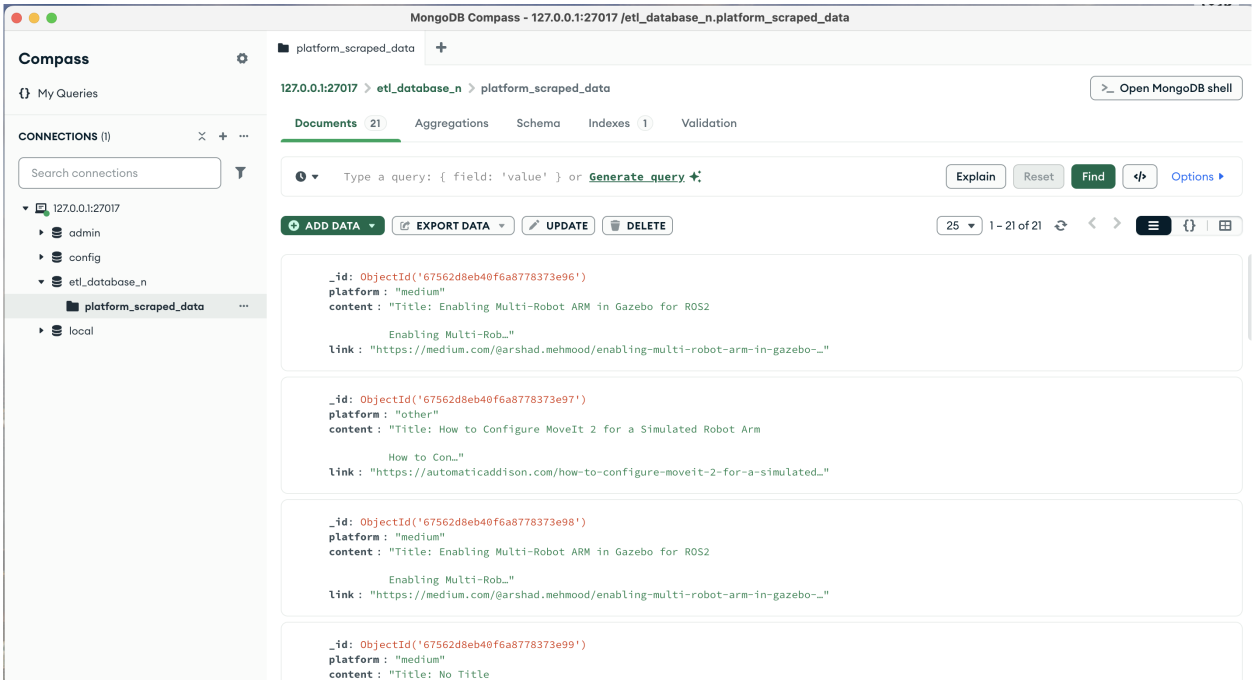
Step 2: ETL Pipeline

- **Goal**: Collect high-quality Q&A data, store it in MongoDB, and prepare it for feature engineering.
- **Completed Tasks**:
    - Blogs and other data sources like Medium articles have been ingested.
    - Data is formatted into a Q&A structure.



```python
def run_pipeline():
        name="Scraping and Storing Pipeline",
        project="ETL Pipeline",
        version="1.0"
    )

    # Single Step: Scrape and Store
    pipe.add_function_step(
        name="Scrape and Store",
        function=scrape_and_store_task,
        function_kwargs={"links": links}
    )

    # Execute Pipeline
    pipe.start_locally(run_pipeline_steps_locally=True)
    logger.info("Pipeline completed successfully.")

if __name__ == "__main__":
    run_pipeline()
```

```
-your-own-ros2-navigation-stack-from-scratch-cc9c6b6a32e6
2024-12-08 18:36:46.995 | INFO     | __main__:scrape_and_store_task:162 - Inserted record for link: https://medium.com/robotics-zone/introd
ucing-ros2-and-gazebo-simulation-for-robotics-3e4054d4f8f8
2024-12-08 18:36:46.996 | INFO     | __main__:scrape_and_store_task:162 - Inserted record for link: https://medium.com/robotics-zone/using-
ros2-and-gazebo-to-simulate-robots-in-a-vibrant-world-34a6a4f35b28
2024-12-08 18:36:46.997 | INFO     | __main__:scrape_and_store_task:162 - Inserted record for link: https://www.youtube.com/watch?v=kR7w5uv
ykRg
2024-12-08 18:36:46.999 | INFO     | __main__:scrape_and_store_task:162 - Inserted record for link: https://www.youtube.com/watch?v=KAyX4Mf
5n2E
Launching the next 0 steps
2024-12-08 18:37:09.023 | INFO     | __main__:run_pipeline:341 - Pipeline completed successfully.
○ ai-final-project-py3.11abdulrazzaq@Mohammads-MacBook-Air RAG_Finetuning %
```

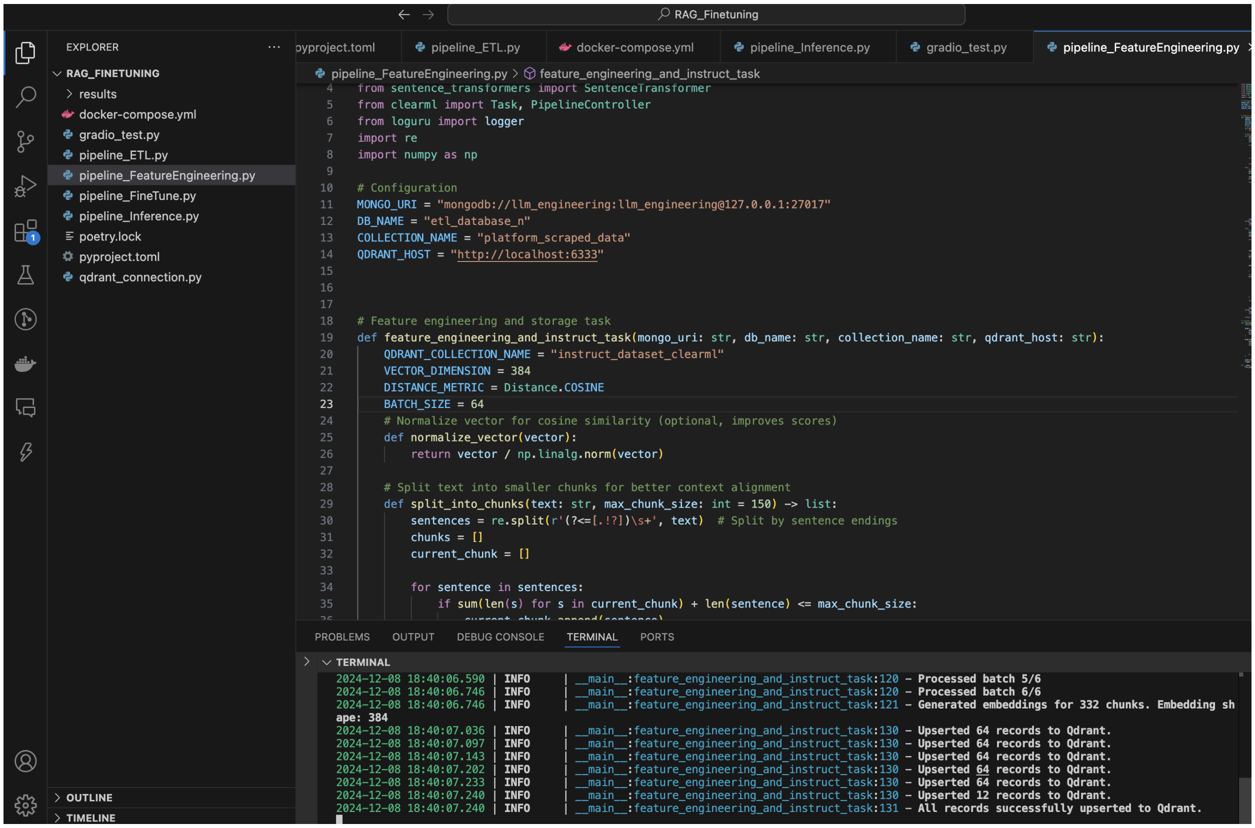


```
_id: ObjectId('67562d8eb40f6a8778373e96')
platform : "medium"
content : "Title: Enabling Multi-Robot ARM in Gazebo for ROS2

         Enabling Multi-Rob…"
link : "https://medium.com/@arshad.mehmood/enabling-multi-robot-arm-in-gazebo-…"

_id: ObjectId('67562d8eb40f6a8778373e97')
platform : "other"
content : "Title: How to Configure MoveIt 2 for a Simulated Robot Arm

         How to Con…"
link : "https://automaticaddison.com/how-to-configure-moveit-2-for-a-simulated…"

_id: ObjectId('67562d8eb40f6a8778373e98')
platform : "medium"
content : "Title: Enabling Multi-Robot ARM in Gazebo for ROS2

         Enabling Multi-Rob…"
link : "https://medium.com/@arshad.mehmood/enabling-multi-robot-arm-in-gazebo-…"

_id: ObjectId('67562d8eb40f6a8778373e99')
platform : "medium"
content : "Title: No Title
```

Added question-and-answer style blog posts to Medium to create a Q&A dataset

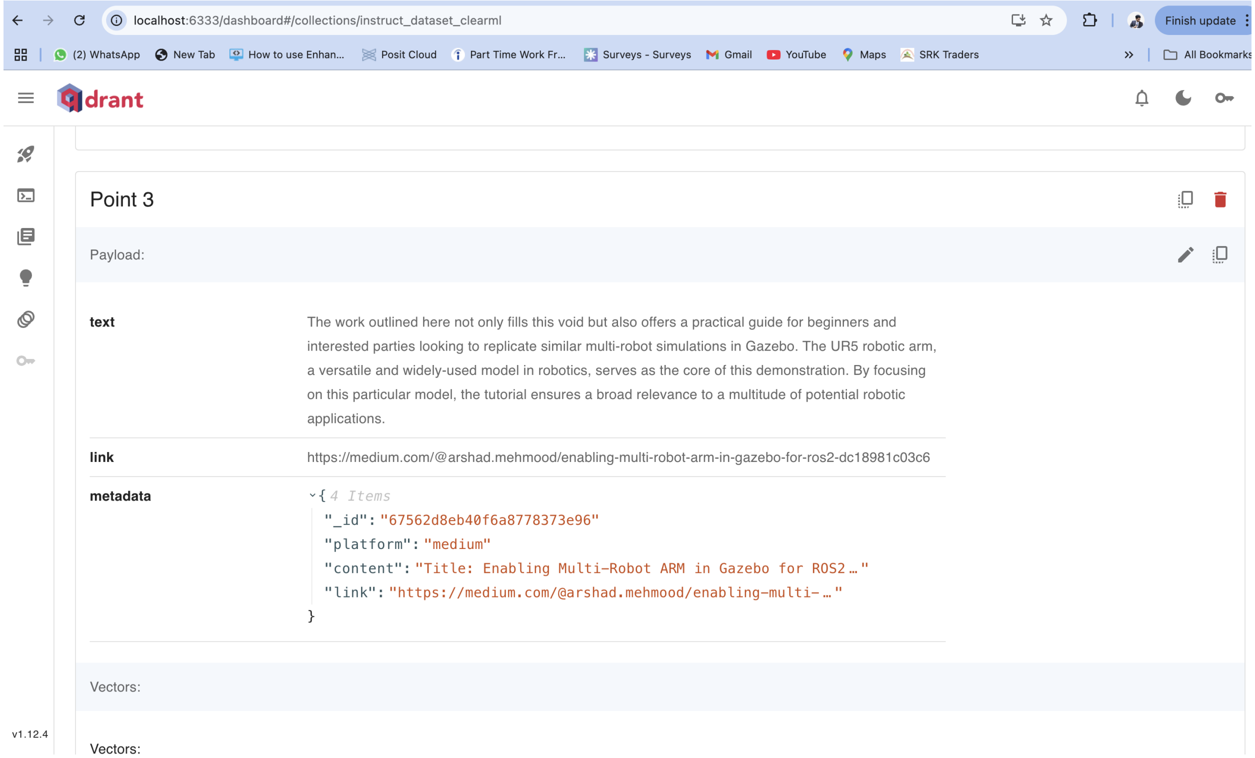
https://abdulrazzaq0902.medium.com/introduction-to-ros2-robot-operating-system-9fa9e9367263

Step 3: Feature Engineering



```python
from sentence_transformers import SentenceTransformer
from clearml import Task, PipelineController
from loguru import logger
import re
import numpy as np

# Configuration
MONGO_URI = "mongodb://llm_engineering:llm_engineering@127.0.0.1:27017"
DB_NAME = "etl_database_n"
COLLECTION_NAME = "platform_scraped_data"
QDRANT_HOST = "http://localhost:6333"



# Feature engineering and storage task
def feature_engineering_and_instruct_task(mongo_uri: str, db_name: str, collection_name: str, qdrant_host: str):
    QDRANT_COLLECTION_NAME = "instruct_dataset_clearml"
    VECTOR_DIMENSION = 384
    DISTANCE_METRIC = Distance.COSINE
    BATCH_SIZE = 64
    # Normalize vector for cosine similarity (optional, improves scores)
    def normalize_vector(vector):
        return vector / np.linalg.norm(vector)

    # Split text into smaller chunks for better context alignment
    def split_into_chunks(text: str, max_chunk_size: int = 150) -> list:
        sentences = re.split(r'(?<=[.!?])\s+', text)  # Split by sentence endings
        chunks = []
        current_chunk = []

        for sentence in sentences:
            if sum(len(s) for s in current_chunk) + len(sentence) <= max_chunk_size:
                current_chunk.append(sentence)
```

```
2024-12-08 18:40:06.590 | INFO     | __main__:feature_engineering_and_instruct_task:120 - Processed batch 5/6
2024-12-08 18:40:06.746 | INFO     | __main__:feature_engineering_and_instruct_task:120 - Processed batch 6/6
2024-12-08 18:40:06.746 | INFO     | __main__:feature_engineering_and_instruct_task:121 - Generated embeddings for 332 chunks. Embedding sh
ape: 384
2024-12-08 18:40:07.036 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
2024-12-08 18:40:07.097 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
2024-12-08 18:40:07.143 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
2024-12-08 18:40:07.202 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
2024-12-08 18:40:07.233 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
2024-12-08 18:40:07.240 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 12 records to Qdrant.
2024-12-08 18:40:07.240 | INFO     | __main__:feature_engineering_and_instruct_task:131 - All records successfully upserted to Qdrant.
```



Point 3

Payload:

text The work outlined here not only fills this void but also offers a practical guide for beginners and interested parties looking to replicate similar multi-robot simulations in Gazebo. The UR5 robotic arm, a versatile and widely-used model in robotics, serves as the core of this demonstration. By focusing on this particular model, the tutorial ensures a broad relevance to a multitude of potential robotic applications.

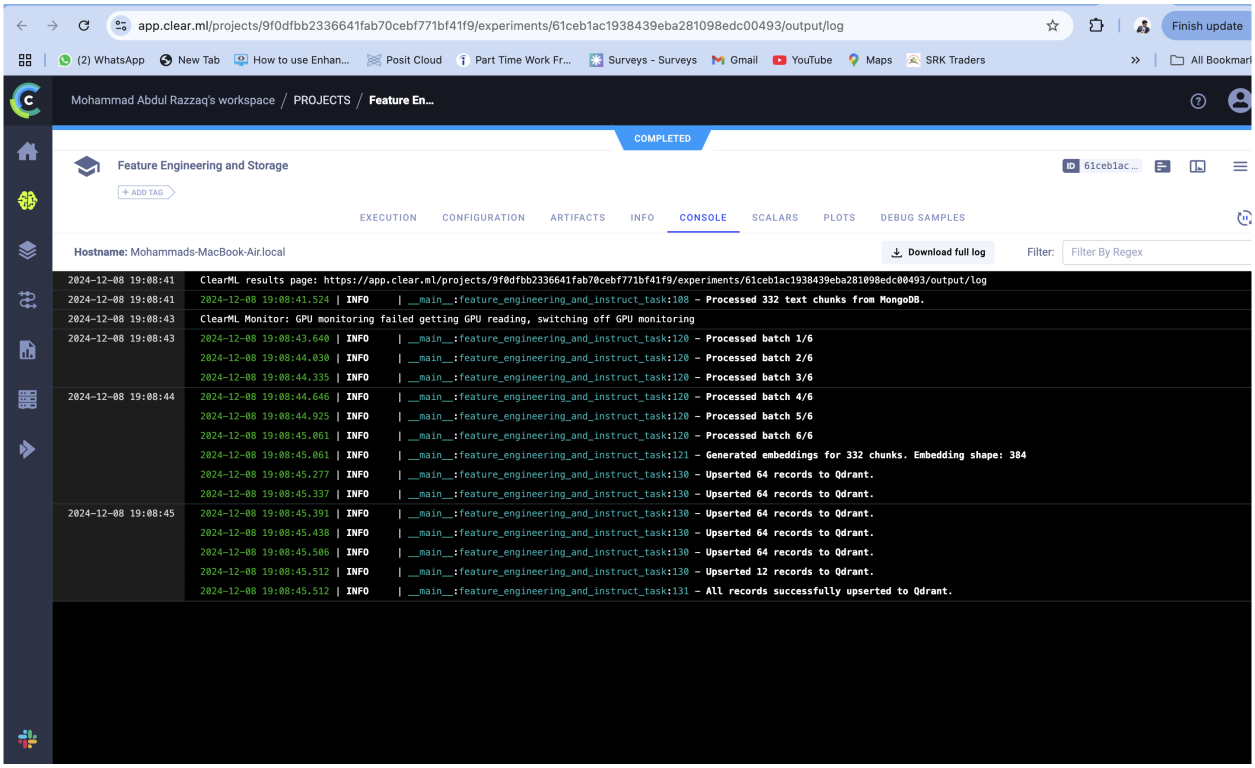
link https://medium.com/@arshad.mehmood/enabling-multi-robot-arm-in-gazebo-for-ros2-dc18981c03c6

metadata

```
{ 4 Items
  "_id":"67562d8eb40f6a8778373e96"
  "platform":"medium"
  "content":"Title: Enabling Multi-Robot ARM in Gazebo for ROS2 …"
  "link":"https://medium.com/@arshad.mehmood/enabling-multi-…"
}
```

Vectors:

Vectors:

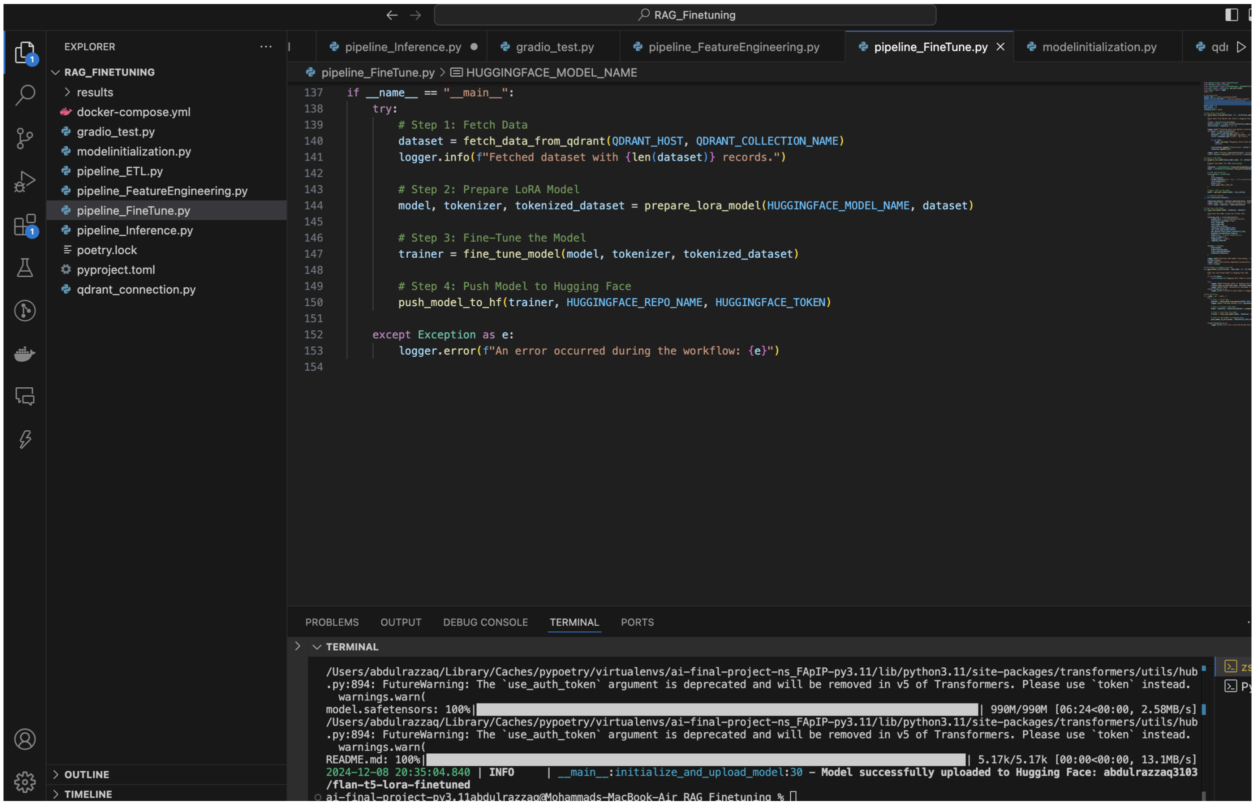


Hostname: Mohammads-MacBook-Air.local

```
2024-12-08 19:08:41   ClearML results page: https://app.clear.ml/projects/9f0dfbb2336641fab70cebf771bf41f9/experiments/61ceb1ac1938439eba281098edc00493/output/log
2024-12-08 19:08:41   2024-12-08 19:08:41.524 | INFO     | __main__:feature_engineering_and_instruct_task:108 - Processed 332 text chunks from MongoDB.
2024-12-08 19:08:43   ClearML Monitor: GPU monitoring failed getting GPU reading, switching off GPU monitoring
2024-12-08 19:08:43   2024-12-08 19:08:43.640 | INFO     | __main__:feature_engineering_and_instruct_task:120 - Processed batch 1/6
                      2024-12-08 19:08:44.030 | INFO     | __main__:feature_engineering_and_instruct_task:120 - Processed batch 2/6
                      2024-12-08 19:08:44.335 | INFO     | __main__:feature_engineering_and_instruct_task:120 - Processed batch 3/6
2024-12-08 19:08:44   2024-12-08 19:08:44.646 | INFO     | __main__:feature_engineering_and_instruct_task:120 - Processed batch 4/6
                      2024-12-08 19:08:44.925 | INFO     | __main__:feature_engineering_and_instruct_task:120 - Processed batch 5/6
                      2024-12-08 19:08:45.061 | INFO     | __main__:feature_engineering_and_instruct_task:120 - Processed batch 6/6
                      2024-12-08 19:08:45.061 | INFO     | __main__:feature_engineering_and_instruct_task:121 - Generated embeddings for 332 chunks. Embedding shape: 384
                      2024-12-08 19:08:45.277 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
                      2024-12-08 19:08:45.337 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
2024-12-08 19:08:45   2024-12-08 19:08:45.391 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
                      2024-12-08 19:08:45.438 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
                      2024-12-08 19:08:45.506 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 64 records to Qdrant.
                      2024-12-08 19:08:45.512 | INFO     | __main__:feature_engineering_and_instruct_task:130 - Upserted 12 records to Qdrant.
                      2024-12-08 19:08:45.512 | INFO     | __main__:feature_engineering_and_instruct_task:131 - All records successfully upserted to Qdrant.
```

- **Goal**: Extract and preprocess features, such as embeddings, for use in Qdrant.
- **Completed Tasks**:
  - Sentence embeddings generated using `all-MiniLM-L6-v2`.
  - Data has been chunked and upserted into Qdrant.

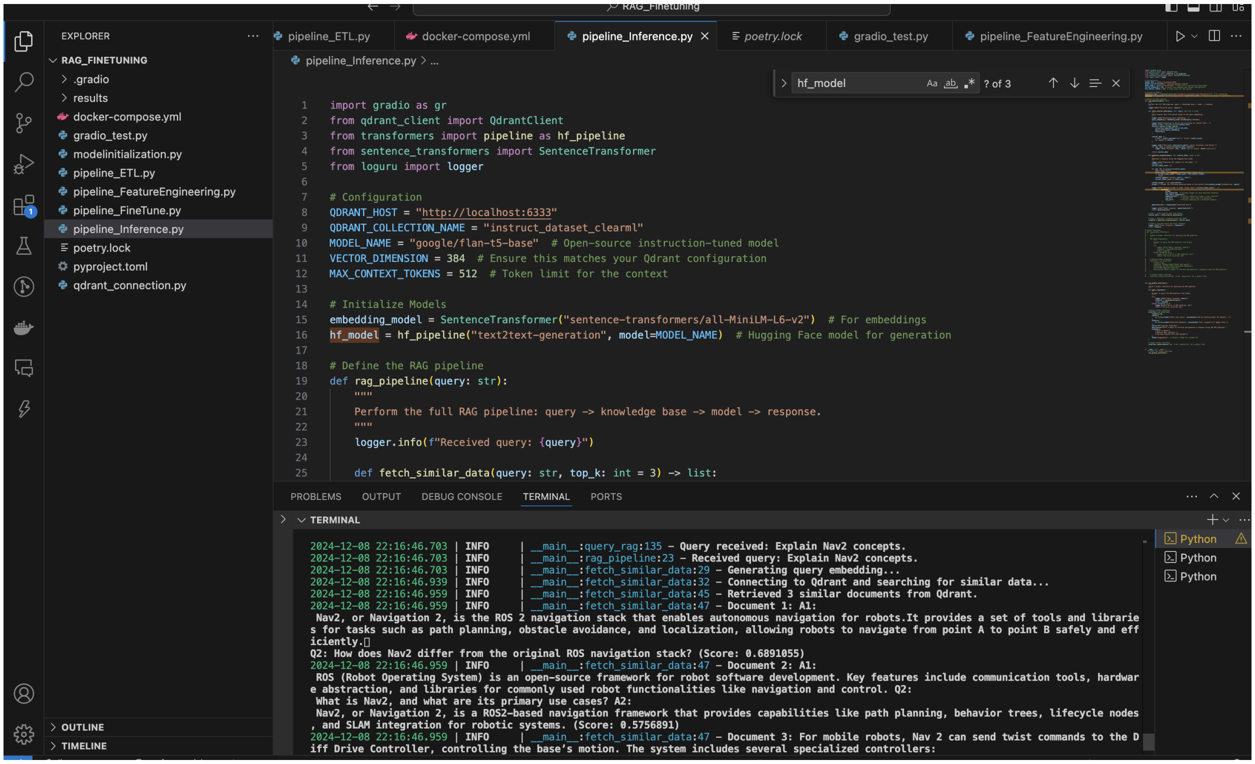
Step 4: Fine-Tuning



```python
if __name__ == "__main__":
    try:
        # Step 1: Fetch Data
        dataset = fetch_data_from_qdrant(QDRANT_HOST, QDRANT_COLLECTION_NAME)
        logger.info(f"Fetched dataset with {len(dataset)} records.")

        # Step 2: Prepare LoRA Model
        model, tokenizer, tokenized_dataset = prepare_lora_model(HUGGINGFACE_MODEL_NAME, dataset)

        # Step 3: Fine-Tune the Model
        trainer = fine_tune_model(model, tokenizer, tokenized_dataset)

        # Step 4: Push Model to Hugging Face
        push_model_to_hf(trainer, HUGGINGFACE_REPO_NAME, HUGGINGFACE_TOKEN)

    except Exception as e:
        logger.error(f"An error occurred during the workflow: {e}")
```

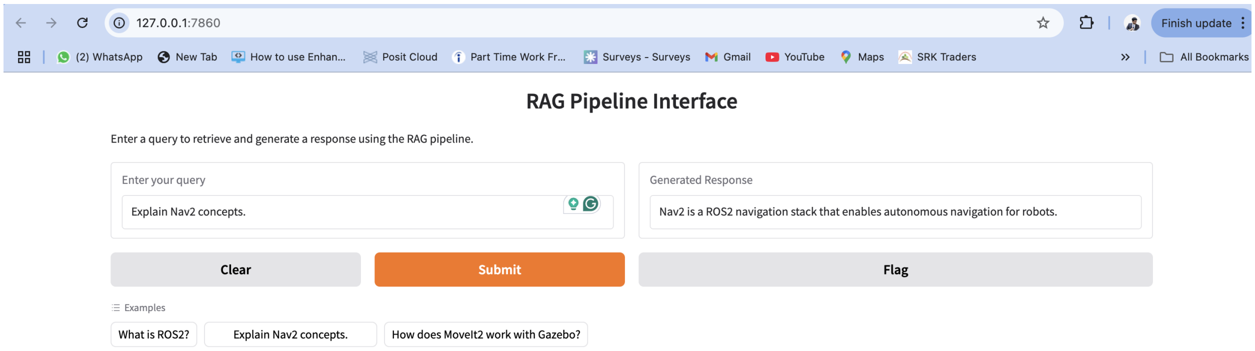
```
/Users/abdulrazzaq/Library/Caches/pypoetry/virtualenvs/ai-final-project-ns_FApIP-py3.11/lib/python3.11/site-packages/transformers/utils/hub
.py:894: FutureWarning: The `use_auth_token` argument is deprecated and will be removed in v5 of Transformers. Please use `token` instead.
  warnings.warn(
model.safetensors: 100%|████████████████████████████████████████| 990M/990M [06:24<00:00, 2.58MB/s]
/Users/abdulrazzaq/Library/Caches/pypoetry/virtualenvs/ai-final-project-ns_FApIP-py3.11/lib/python3.11/site-packages/transformers/utils/hub
.py:894: FutureWarning: The `use_auth_token` argument is deprecated and will be removed in v5 of Transformers. Please use `token` instead.
  warnings.warn(
README.md: 100%|███████████████████████████████████████████| 5.17k/5.17k [00:00<00:00, 13.1MB/s]
2024-12-08 20:35:04.840 | INFO     | __main__:initialize_and_upload_model:30 - Model successfully uploaded to Hugging Face: abdulrazzaq3103
/flan-t5-lora-finetuned
ai-final-project-py3.11abdulrazzaq@Mohammads-MacBook-Air RAG_Finetuning %
```

- **Goal**: Fine-tune the `Flan-T5` model using LoRA to adapt it to your specific Q&A dataset.
- **Completed Tasks**:
  - Initial fine-tuning setup with LoRA and Hugging Face libraries.



```python
import gradio as gr
from qdrant_client import QdrantClient
from transformers import pipeline as hf_pipeline
from sentence_transformers import SentenceTransformer
from loguru import logger

# Configuration
QDRANT_HOST = "http://localhost:6333"
QDRANT_COLLECTION_NAME = "instruct_dataset_clearml"
MODEL_NAME = "google/flan-t5-base"  # Open-source instruction-tuned model
VECTOR_DIMENSION = 384  # Ensure this matches your Qdrant configuration
MAX_CONTEXT_TOKENS = 512  # Token limit for the context

# Initialize Models
embedding_model = SentenceTransformer("sentence-transformers/all-MiniLM-L6-v2")  # For embeddings
hf_model = hf_pipeline("text2text-generation", model=MODEL_NAME)  # Hugging Face model for generation

# Define the RAG pipeline
def rag_pipeline(query: str):
    """
    Perform the full RAG pipeline: query -> knowledge base -> model -> response.
    """
    logger.info(f"Received query: {query}")

    def fetch_similar_data(query: str, top_k: int = 3) -> list:
```

```
2024-12-08 22:16:46.703 | INFO     | __main__:query_rag:135 - Query received: Explain Nav2 concepts.
2024-12-08 22:16:46.703 | INFO     | __main__:rag_pipeline:23 - Received query: Explain Nav2 concepts.
2024-12-08 22:16:46.703 | INFO     | __main__:fetch_similar_data:29 - Generating query embedding...
2024-12-08 22:16:46.939 | INFO     | __main__:fetch_similar_data:32 - Connecting to Qdrant and searching for similar data...
2024-12-08 22:16:46.959 | INFO     | __main__:fetch_similar_data:45 - Retrieved 3 similar documents from Qdrant.
2024-12-08 22:16:46.959 | INFO     | __main__:fetch_similar_data:47 - Document 1: A1:
 Nav2, or Navigation 2, is the ROS 2 navigation stack that enables autonomous navigation for robots.It provides a set of tools and librarie
s for tasks such as path planning, obstacle avoidance, and localization, allowing robots to navigate from point A to point B safely and eff
iciently.□
Q2: How does Nav2 differ from the original ROS navigation stack? (Score: 0.6891055)
2024-12-08 22:16:46.959 | INFO     | __main__:fetch_similar_data:47 - Document 2: A1:
 ROS (Robot Operating System) is an open-source framework for robot software development. Key features include communication tools, hardwar
e abstraction, and libraries for commonly used robot functionalities like navigation and control. Q2:
 What is Nav2, and what are its primary use cases? A2:
 Nav2, or Navigation 2, is a ROS2-based navigation framework that provides capabilities like path planning, behavior trees, lifecycle nodes
, and SLAM integration for robotic systems. (Score: 0.5756891)
2024-12-08 22:16:46.959 | INFO     | __main__:fetch_similar_data:47 - Document 3: For mobile robots, Nav 2 can send twist commands to the D
iff Drive Controller, controlling the base’s motion. The system includes several specialized controllers:
```



RAG Pipeline Interface

Enter a query to retrieve and generate a response using the RAG pipeline.

Enter your query: Explain Nav2 concepts.

Generated Response: Nav2 is a ROS2 navigation stack that enables autonomous navigation for robots.

Step 5: Contextual Retrieval and Inference

- **Goal**: Implement the end-to-end RAG pipeline for context-based response generation.
- **Completed Tasks**:
  - Query embeddings are generated, and Qdrant retrieves relevant context.

- Prompt structure integrates context with the user query.