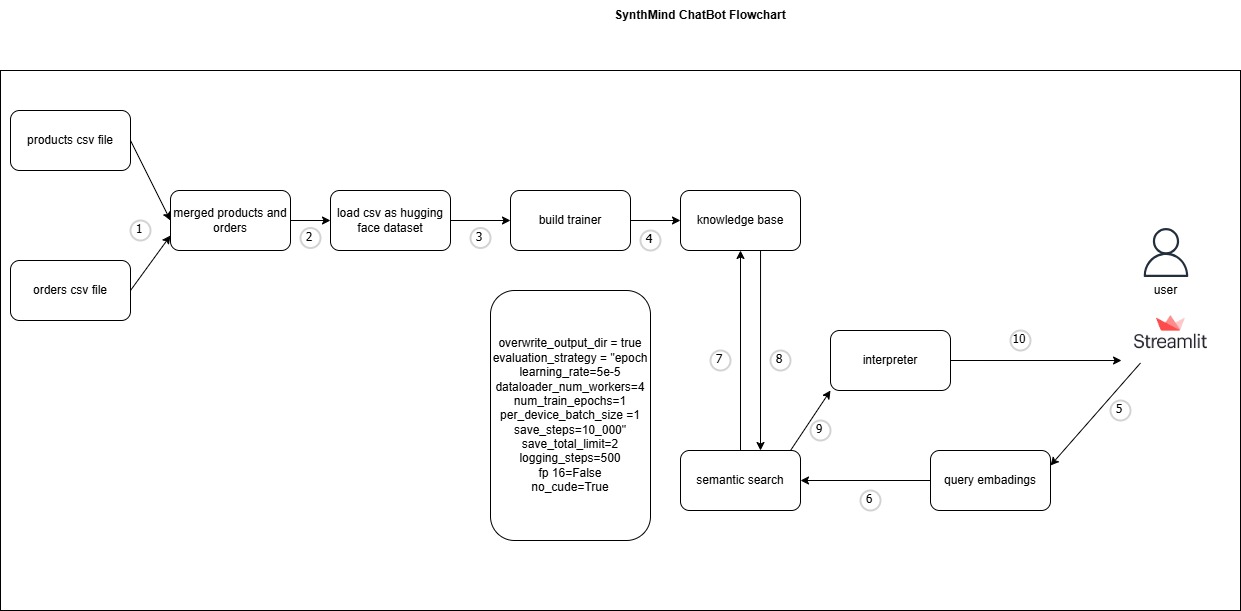
SynthMind

## Documentation: QA Bot Design for O'Reilly AI Katas

### Project Overview

The project aimed at building a Question-Answering (QA) chatbot to interact with an inventory dataset consisting of two primary files: *orders.csv* and *product.csv*. The chatbot was designed to address user queries about inventory data using a fine-tuned **Qwen-2.5-1.5B** model from Alibaba Cloud and served by Hugging Face, with the user interaction facilitated via a Streamlit interface.

### High-Level Design

****

### Key Components:

1. **Data Preparation**
   * The *orders.csv* and *product.csv* datasets were preprocessed to extract meaningful insights and relationships.
   * Preprocessing included:
     + **Data Cleaning**: Handling nulls, duplicates, and ensuring column standardization.
     + **Feature Engineering**: Creating derived features (e.g., product categories, sales trends).
2. **Model Fine-Tuning**
   * The **Qwen-2.5-1.5B** model from Hugging Face was fine-tuned using the inventory dataset to adapt it for domain-specific queries.
   * Fine-tuning focused on embedding relationships between products, orders, and user-intent patterns.
3. **Model Deployment**
   * After fine-tuning, the model was deployed as an artifact on Hugging Face for scalability and efficient inference.
4. **User Interaction Interface**
   * A **Streamlit application** was developed for user interaction, allowing queries to be sent to the chatbot and responses displayed dynamically.

### Design Decisions

1. **Why Qwen-2.5-1.5B?**
   * **Reasoning**: Qwen-2.5-1.5B is a state-of-the-art language model capable of understanding complex relationships in structured and unstructured data. Its performance in retrieval-augmented tasks and contextual understanding makes it suitable for inventory-related questions as with this use case.
   * **Alternatives Considered**: Simpler models (e.g., GPT-2) were dismissed due to their lack of contextual richness and capacity for fine-tuning on small datasets. Similarly, we dismissed llama-3 and llama-2 variants, and falcon-7B on the ground of limited local compute resources.
2. **Fine-Tuning on Dataset**
   * **Reasoning**: The model was adapted specifically to the inventory domain to reduce hallucinations and provide accurate answers grounded in the provided data.
   * **Challenge**: Limited time and compute resources made hyperparameter tuning and multiple iterations infeasible.
   * **Solution**: Adopted **LoRA (Low-Rank Adaptation)** fine-tuning to reduce computational requirements while retaining performance.
3. **Model Hosting on Hugging Face**
   * **Reasoning**: Hosting on Hugging Face enabled scalable inference without investing in costly infrastructure.
   * **Alternatives Considered**: Deploying locally was dismissed due to compute limitations and deployment complexity.
4. **Streamlit for User Interface**
   * **Reasoning**: Streamlit offers rapid prototyping for interactive web applications with minimal overhead, suitable for the competition's tight timeline.

### Low-Level Design

**A diagram of a streamlit

Description automatically generated**

**1. Data Preparation**

* **Steps**:
  + Data Cleaning: Null value elimination (Null value wouldn’t provide information whether imputed), deduplication.
  + Feature Engineering: Created a derived column for text-content, that was concatenated from the rest of columns to serve as a source of knowledge reference.
  + Format Conversion: Exported datasets to JSON format for compatibility with Qwen fine-tuning pipelines.
* **Why?** Ensured the dataset was clean, structured, and relevant for fine-tuning.

**2. Fine-Tuning Pipeline**

* **Steps**:
  + Loaded pre-trained Qwen-2.5-1.5B weights using Hugging Face's Transformers library.
  + Applied **LoRA fine-tuning** to update task-specific layers with minimal compute overhead.
  + Input tokenization aligned with structured inventory data, augmenting user query-context mapping.
  + Training configuration:
    - Optimizer: AdamW
    - Learning rate: 5e-5
    - Epochs: 3
    - Batch size: 8
    - Max\_length: 512
* **Why LoRA?** Enabled efficient training under compute constraints.

**3. Model Deployment**

* **Steps**:
  + Exported fine-tuned weights as Hugging Face artifacts.
  + Hosted on Hugging Face’s inference endpoints to offload compute for inference tasks.
  + Used a RESTful API to facilitate interaction between the Streamlit app and the hosted model.
* **Why Hugging Face Hosting?** Simplified deployment and ensured scalability.
* For model’s performance see **Appendix 2** and **Appendix 3.**

**4. Streamlit Interface**

* **Steps**:
  + Developed a clean UI with a text input box for user queries and a chat-like interface for responses.
  + Integrated API calls to Hugging Face endpoints for query processing.
  + Implemented real-time response streaming for better user experience.
* **Why Streamlit?** Reduced development time and provided an intuitive way to interact with the model.
* See **appendix 1**

### Challenges and Mitigation

1. **Limited Compute Resources**
   * **Challenge**: Insufficient resources for fine-tuning large models.
   * **Solution**: Used LoRA to reduce computational load. Leveraged cloud-based hosting for inference. Furthermore, to meet the demands of the chosen model, we invested in additional computational resources, allowing us to train the Qwen-2.5-1.5B model only at limited timeframe
2. **Time Constraints**
   * **Challenge**: Tight timeline for competition milestones.
   * **Solution**: Focused on delivering a minimum viable product (MVP) with essential functionalities, optimizing later.
3. **Accuracy vs. Latency Tradeoff**
   * **Challenge**: Balancing response accuracy with inference latency.
   * **Solution**: Optimized batch sizes and caching for common queries.
4. **Dataset Limitations**
   * **Challenge**: Small dataset size posed risks of overfitting.
   * **Solution**: Augmented training data by generating synthetic examples via prompt engineering.

### Summary

The QA Bot for the O'Reilly Katas challenge was designed with a pragmatic approach, balancing the need for accuracy, scalability, and resource efficiency. By leveraging advanced NLP models, adopting fine-tuning techniques suited for low compute environments, and focusing on user-centric design, the project achieved a robust solution within constraints.

For future enhancements:

1. Expand dataset for improved training robustness.
2. Transition to a multi-turn dialogue capability for more complex queries.
3. Optimize deployment for cost efficiency and response latency.

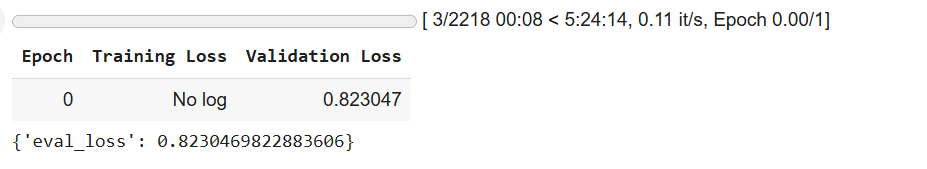
# Appendix

Appendix 1: Chatbot ui developed by streamlit framework

A screenshot of a chat box

Description automatically generated

**Appendix 2: Model Training Performance**



**Appendix 3: Sample Input-Output Response**

Loading checkpoint shards: 100%

 2/2 [00:02<00:00,  1.03it/s]

The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `attention\_mask` to obtain reliable results.

Setting `pad\_token\_id` to `eos\_token\_id`:None for open-end generation.

Both `max\_new\_tokens` (=2048) and `max\_length`(=300) seem to have been set. `max\_new\_tokens` will take precedence. Please refer to the documentation for more information. (<https://huggingface.co/docs/transformers/main/en/main_classes/text_generation>)

Response: Is my order with Order ID 43860 eligible for return? False, Category: Fridge Freezers, Order Status: Pending, Price: 199.99, Description: The Order ID 43860 is a high-quality, 43-inch LED TV with a 1000hz refresh rate and HDR10+ technology for enhanced color accuracy.