Amazon ML Challenge 2024

**Team Name: DALL - E**

**Introduction**

This project aims to develop an advanced Vision Language Model (VLLM) capable of extracting entity values from product images. Our approach involves processing a dataset containing image URLs and associated product information to build a sophisticated machine learning model for entity value prediction. The VLLM takes two key inputs: the URL of the product image and a prompt containing the entity name asked. Based on these inputs, the model generates the corresponding entity value along with its appropriate unit of measurement.

**Methodology**

Our approach to solving the entity value extraction task involves leveraging and fine-tuning a state-of-the-art Vision Language Model (VLM). The methodology can be broken down into the following key steps:

**1. Model Selection**

We chose the Qwen2-VL-2B-Instruct (Top-5 in OCRBench) model as our base architecture. This model is part of the Qwen (Qwen-VL) family, developed by Alibaba Cloud, and is specifically designed for vision-language tasks. Key features of Qwen2-VL-2B-Instruct:

* Multimodal capabilities, allowing it to process both images and text
* Pre-trained on a large corpus of image-text pairs
* Designed for instruction-following tasks, making it suitable for our specific use case

**2. Fine-tuning Process**

We fine-tuned the Qwen-VL-2B-Instruct model using a structured pipeline. Jupyter notebooks (.ipynb) for both training and inference are provided for simplicity. The process involves:

**3. Training:**

**a. Environment Setup:**

* Install required libraries (ensure latest version of transformers)
* Choose between custom training script or bundled version using swift library
* Note: Custom script may require >40GB RAM due to Deepspeed zero3 config

**b. Prompt Engineering:**

We designed a specialized prompt to guide the model in extracting entity values. The query prompt is constructed dynamically based on the entity name and image path. The following prompt structure was used:

{

"system": "You are an entity extractor OCR model. Given the entity, you can extract the text from the image denoting the entity value. Entity value is always a number followed by its unit like {entity\_unit\_map[row['entity\_name']]}.",

"query": "<image> What is the value of the {row['entity\_name']} of the item shown? Give me the numerical value and unit as seen in the image. Example Input: What is the value of the volume of the item shown, Output: 1 cup. Adhere to instructions.",

"response": "{row['entity\_value'].split()}",

"images": ["{image\_path}"]

}

**c. Data Preparation:**

* Create a stratified training sample maintaining distribution of entity\_names
* Download images using image\_downloader function (chunking and multiprocessing for efficiency)
* Prepare dataset as a json object for model input

**d. Model Saving:**

* Use PeFT (via swift) and LoRA for fine-tuning
* Save lightweight checkpoint (~200-300MB) with LoRA adapters and optimizer

**Inference:**

The test data is prepared by downloading the necessary images and creating a test.json file that includes system and user prompts along with image paths. During inference, GPTQ and 4-bit quantization are employed to optimize computational efficiency. Finally, the output in JSONL format is converted to the required CSV format. This fine-tuning process enables effective adaptation of the Qwen-VL model to the task while optimizing resource usage.

**Experiments:**

**Overall Performance**

The Vision Language Model (VLLM) demonstrated strong performance across a wide range of entity types and product categories. It successfully extracted values for attributes such as: Weight, Volume, Voltage, Wattage. The model's ability to understand context and relate visual information to textual prompts allowed it to accurately identify and extract these entity

**Challenges with Dimensional Measurements**

Despite its overall strong performance, we observed that the VLLM struggled with accurately capturing information for height, width, and depth measurements. This challenge was particularly evident in cases where:

* The image perspective made it difficult to distinguish between different dimensions.
* Multiple measurements were present in the image without clear labeling.
* The scale or reference point for measurements was not obvious from the image alone.

**Innovative Solution: Directional Annotations**

We developed an image preprocessing step that adds directional annotations to the product images. We overlaid simple, non-intrusive directional indicators on the images, such as:"H" for height, "W" for width, "D" for depth

* These indicators were strategically placed near the edges of the image, corresponding to the respective dimensions.
* Color Coding: We used a color scheme that contrasts with the original image to ensure visibility
* Training Data Augmentation: We applied this preprocessing to our training dataset and fine-tuned the model

**Results of Directional Annotations**

* Substantial improvement in accuracy for height, width, and depth measurements.
* Enhanced ability to distinguish between similar dimensions.
* Improved performance on test images without annotations, indicating better spatial understanding.
* No negative impact on the model's performance for other entity types.

**Conclusion**

Our project successfully leveraged the Qwen2-VL-2B-Instruct model for entity value extraction from product images. Key achievements include:

* Effective fine-tuning on 3,000 images
* Innovative use of directional annotations for improved dimensional measurements
* Enhanced spatial understanding, generalizing to unannotated images
* Potential applications in e-commerce, healthcare, and content moderation

This solution promises to streamline product information systems, enhancing efficiency in digital marketplaces.