
MINERVA
Winter School Challenge: Vision-based MLLM agent
Dott.ssa Sara Sarto

The Challenge

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Multimodal Large Language Models (MLLMs) are powerful, but they suffer from three major issues:

1. they don't know everything (lack of encyclopedic knowledge),
2. they tend to hallucinate (seeing things that aren't there),
3. they struggle to read dense, small text in documents.

The Challenge

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Goal:

You will need to create a pipeline where the model acts as the "Brain" (Router) to analyze the user's image and question, deciding which Tool to call to get the correct answer.

Different Tasks:

You will be evaluated on three very different tasks:

- Encyclopedic-VQA (EVQA): Questions requiring deep external knowledge (e.g., recognizing a specific monument, a rare species, an artwork).
- Amber (Discriminative): Questions aimed at testing hallucinations. They are all of the type *"Is there a [object] in the image?"* (Expected answer: Yes/No).
- Document VQA (DocVQA): Questions extracting specific data from dense scanned documents, invoices, and receipts.

The winning team will be the one that achieves the highest combined score across all test datasets!

Technical Details

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Your system must implement a **multi-tool** logic. You are required to build the following components:

1. **The Router Logic:** you must prompt Qwen2.5-VL-7B to classify the question intent (encyclopedic, hallucination check, or document reading) and output a structured command to trigger the correct tool.

2. **Tool #1: The Researcher** (Information Retrieval)
   - Trigger: EVQA questions.
   - Function: Takes the image and queries a pre-computed index to retrieve relevant documents from the provided Knowledge Base.
   - Usage: The retrieved documents must be injected as context into your MLLM prompt to generate the final, informed answer.

Technical Details

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

3. Tool #2: The Eagle Eye (Object Detection)
   - Trigger: Amber Discriminative questions (*"Is there a...?"*).
   - Function: Runs a targeted Object Detection task on the specific object requested in the question to prevent MLLM blind hallucinations.
   - Usage: The detection might help the model to answer the question correctly.

4. Tool #3: The Reader (OCR)
   - Trigger: DocVQA questions.
   - Function: Because MLLMs drop details on dense documents when resizing images, this tool uses an external OCR system (e.g., EasyOCR) to extract all text from the image.
   - Usage: Pass this raw text back to the MLLM in the prompt so it can find the exact answer without hallucinating numbers or characters.

Provided Materials

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Test Datasets (The questions you must answer):

- EVQA Test Split: `data/evqa_test.csv`
- Amber Discriminative Test: `data/amber_disc.json`
- DocVQA Test: [INSERT_PATH_TO_DOCVQA_HERE]

Knowledge Base & Indices (For Tool #1):

- EVQA Knowledge Base: `data/encyclopedic_kb_wiki.json`
- Pre-computed Retrieved Entities: `data/lens_entities.csv`
- Retrieval Note: In `data/lens_entities.csv` you will find which documents Google Lens selected to answer the question on the image.
  - Hint: Use the `dataset_image_id` field of each sample to retrieve the correct Wikipedia URLs.

Models & Tools:

- `Qwen2.5-VL-7B`: Load the model using the cached file in `./hf_models`

Some Rules

1. **Zero-Shot Only:** Fine-tuning the model is strictly forbidden. The system must work "out-of-the-box" relying solely on prompt engineering, parsing, and proper tool usage.

2. **Multi-Tool Mandatory:** A system answering using *only* Qwen2.5-VL without calling the correct external tools will be heavily penalized or disqualified.

3. **Output Format:** Your final script must generate a JSON file with your predictions in the following format:
   - [{"data_id":"<dataset_name>_<idx>","prediction":"...","groun_truths":"..."},
     {"data_id""<dataset_name>_<idx>","prediction":"...","groun_truths":"..."}, ...]

Tips For Victory

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- **Router Prompting:** Ensure your initial prompt clearly explains to Qwen2.5-VL the difference between the types of questions and *how* to format the tool call (e.g., structured JSON output or special parsing tags).

- **Detector Thresholds:** For Tool 2, play smart with the bounding box confidence threshold to avoid false positives and false negatives.

- **RAG Context:** Format the retrieved documents cleanly before passing them to the model. Too much useless information (noise) will confuse Qwen and degrade its response.

Team Strategy: How to Divide and Conquer (5 Members)

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- Member 1: Router & Main LoopTask:
  - Sets up the main Python script, loads the offline Qwen2.5-VL model, and writes the complex "Router" prompt. Manages the main for loop that iterates over the datasets and calls the tools created by the others.
- Member 2: Tool 1 - RetrievalTask:
  - Takes ownership of EVQA. Writes the function to load the CSV/JSON data, maps the dataset_image_id to the Lens entities, and formats the retrieved text clearly for the prompt.
- Member 3: Tool 2 - Object DetectionTask:
  - Takes ownership of Amber. Builds an independent Python function that runs the object detection, tunes the bounding box confidence thresholds, and returns the logical output (Yes/No or the coordinates).
- Member 4: Tool 3 - OCRTask:
  - Takes ownership of DocVQA. Sets up the offline OCR (e.g., EasyOCR), writes the function to extract the raw text from the image, and formats it cleanly to be appended to the prompt.
- Member 5: Integration & EvaluationTask:
  - The most critical role in the last hour. Sets up a shared GitHub repo (or Live Share), writes the function to dump the final predictions into the required JSON format, handles error catching (try/except blocks so the code doesn't crash halfway), and runs small local tests to ensure the assembled pipeline actually works.

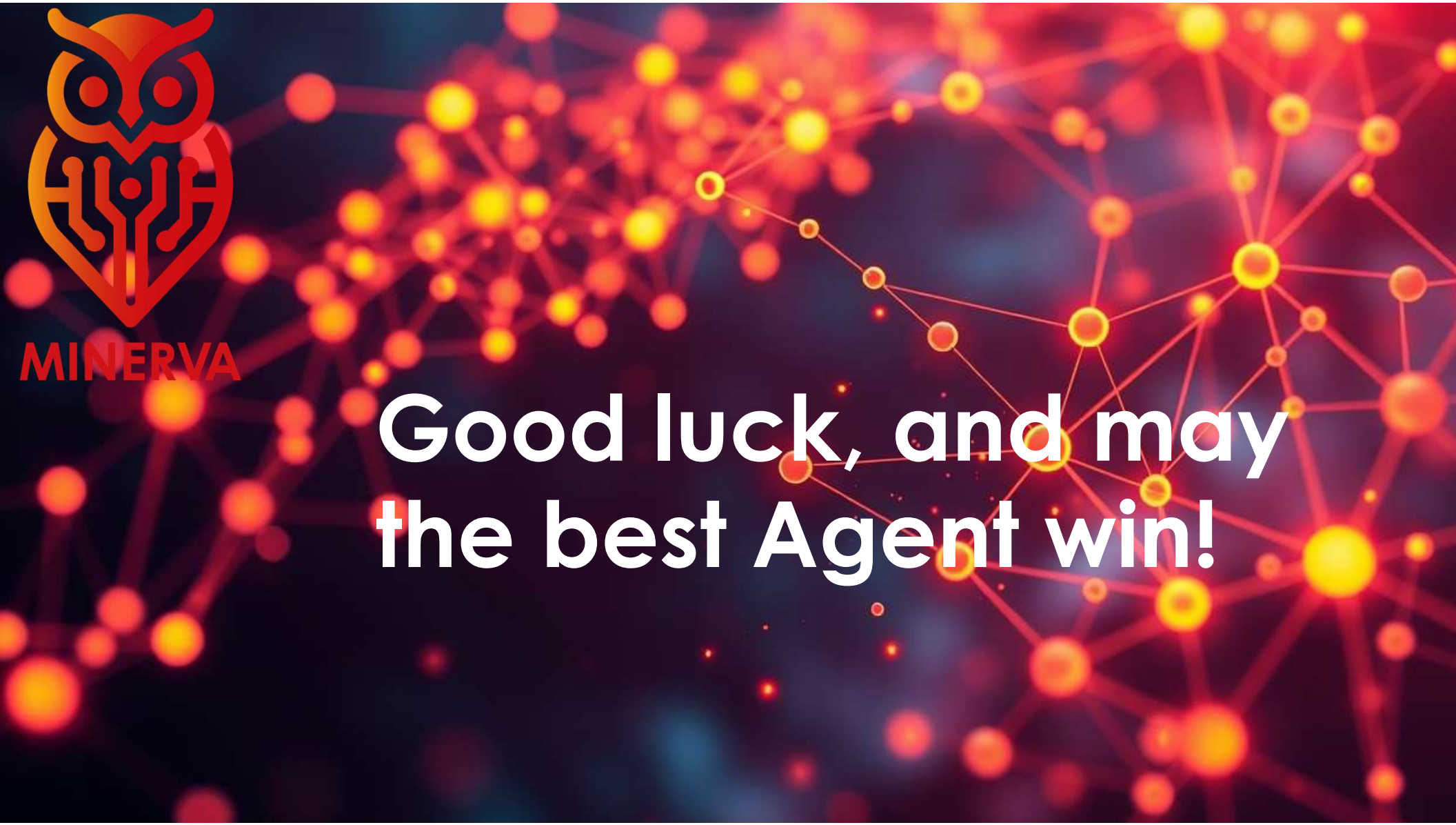
Score Calculation

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Use the evaluation script at `utils/final_score.py` to calculate the datasets scores. Provide the predictions file paths as args to the script.

Example:

```
python utils/final_score.py \
--input_path_evqa evqa_prdictions.json \
--input_path_docvqa docvqa_prdictions.json \
--input_path_amber_disc amber_disc_predictions.json
```


MINERVA
Good luck, and may the best Agent win!