fellowship.ai

Demo and Status Report: Autonomous Web Agent

- 28.02.2025

Overview

Recent progress

- After testing both models, we finalized UI-TARS over Midscene since it handled tasks better, and started integrating it for financial data scraping.
- We ran benchmark tests to see how UI-TARS stacks up against other models and made adjustments to improve its ability to process financial data.
- Since UI-TARS is resource-heavy, we analyzed its computational costs and runtime to see if it can work efficiently as a lightweight automation model.

Midscene Testing – What Happened?

Go to Amazon, search for "Dell Laptop," select the first suggested search term, choose the first product in search results, and extract only its price. Ignore all other elements.

Results

- Midscene successfully navigated Amazon but failed before extracting the price.
- It processed too much data and hit token limits before completing the task.
- Eventually, Amazon blocked further requests due to multiple attempts (429 rate limit error).

Midscene Token Limit Adjustments

*MIDSCENE_OPENAI_INIT_CONFIG_JSON='{"defaultheaders": {"Authorization": ""}, "max_new_tokens": **1000**, "max_input_tokens": **4000**}'*

1) Set max_input_tokens = 4000, max_new_tokens = 1000.
   Result - Failed – The token limit was exceeded.
2) Adjusted to max_input_tokens = 2000, max_new_tokens = 1000.
   Result - Failed – Still exceeded the limit.
3) Reduced further: max_input_tokens = 1000, max_new_tokens = 500.
   Result - Failed – The issue persisted due to the dynamic content and large data being processed.

UI-TARS Desktop – A Better Alternative

Since Midscene struggled, we decided to test UI-TARS Desktop instead. This time, we used a real-world scenario -

Open the HDFC Bank website, search for "Personal Loan EMI Calculator," enter a random loan amount and tenure, calculate the EMI, and return the result.

Results

- UI-TARS navigated the site correctly and located the calculator.
- It successfully entered the loan details and adjusted the sliders without issues.
- The EMI was calculated and displayed as expected.

UI-TARS - Web Scraper (Under Testing)

We've been focused on refining the scraping agent by updating Iremide's tools. The "key" challenge remains adjusting the UI-TARS model's prompting mechanism, as the prompt and next action are currently being returned together. While model integration is still being fine-tuned, we're testing scraping actions on the Bankrate website, drawing from the previous cohort's code for consistency.

Next Steps

- Capture Website URL (Bankrate) & Screenshot (Playwright)
- OCR Analysis & Action Prediction (UI-TARS Model)

Benchmarking Results & Insights

We Used Benchmark Tests for QA(Squad), Reasoning(Hellswag), Math(GSM8K) and Summarization(CNN Daily) to test UI Tars performance against Llama 3.2 1B and 3B. Although the results were far below the standard, we realised that UI TARS is a vlm model and it needs multimodal inputs so just testing on textual inputs is not the correct way.

We then ran the benchmarking on VQA(squad), Image Captioning (COCO dataset) and Text-Image Reasoning (ScienceQA)

Fine-Tuning Progress

For Fine Tuning we used FinLang/investopedia-instruction-tuning-dataset. Here's the LoRa Config Updates and Training Arguments Improvement

LoRA Config Updates

- Increased rank (r=64) → Allows more trainable parameters.
- Higher LoRA alpha (128) → Strengthens adaptation.
- Increased dropout (0.1) → Helps prevent overfitting.

Training Arguments Improvements

- Higher learning rate (5e-5) → Allows better updates.
- More epochs (5) → Provides sufficient training time.
- Gradient accumulation (8) → Helps with limited memory.
- FP16 enabled → Faster training with reduced memory usage.

Why Is Fine-Tuning Not Improving the Output?

Even with these improvements, the fact that the before and after fine-tuning responses are identical suggests that the model is not learning useful patterns from the dataset.

1. We were using "Context" column in the dataset [Dataset column names: ['Topic', 'Title', 'Context', 'Question-Answer', 'Question', 'Answer', 'bge-large-en-v1.5-correlation']]
   We need to use "Question" and "Answer" instead for better instruction-based learning.
2. Your current batch size: [per_device_train_batch_size=4] we can increase it to 8 if the vram allows
3. Verify If Fine-Tuning Changed Model Weights. Before reloading, manually compare pre-trained and fine-tuned models

Computational Costs & Runtime Evaluation

Today, we discussed the performance differences between the UI-TARS 7B model in Midscene and the UI-TARS Desktop agent, as well as the apparently high computational costs of UI-TARS, which may make it harder for it to function as a lightweight, web-integrated agent. In agreement with Feras, we will now focus on evaluating the performance, computational costs, and runtime for

- The UI-TARS model.
- The *proxy-lite-3b* model, another model for UI navigation tasks suggested by Kukesh, which we will explore.
- The agent built by the previous cohort.

Once the review is over we will discuss which path we should pursue with Feras or Kukesh.

Goals for next meeting

- Finalize UI-TARS integration for financial scraping tasks and ensure stable execution.
- Refine model prompting to improve accuracy and reduce unnecessary outputs.
- Optimize fine-tuning configurations to enhance model performance on financial data.