fellowship.ai

Demo and Status Report: Autonomous Web Agent

- 07.02.2025

Overview

Recent progress

- Researched about UI TARS Model and went through its documentation to understand how it works.

Currently Working

- Running the UI TARS Desktop model

Goals for the next week

Trying out Different Models

We started this week by implementing different models to see how compatible and accurate they were. We experiment with Deepseek R1 WebGPU Model but quickly realised that to navigate across different functions will have to build frameworks with different functions and that will make our code less complex.

Then we started building a scraping tool which will use LLM to scrape the data from the web instead of the traditional tools which require hardcoded URL Links. And Learnt how LLM is used for scrapping, How Langchain works and What are the suitable Sentence Embedding models that we can use for Scrapping.

Why UI TARS?

Previous Approach:

1. Used **GPT-4.0 and LLaMA** for core tasks, requiring additional models for specific functions.
2. **YOLO** for object detection, **EasyOCR** for text extraction, **BLIP-2** for image captioning.
3. **Image Hash** for detecting differences between consecutive images.
4. Dependency on multiple models made the system **complex and resource-intensive**.

Why UI-TARS is Better for Autonomous Web Agents:

1. **Unified Model** – Handles perception, reasoning, and action without external dependencies.
2. **Simplifies Codebase** – Eliminates the need for separate models, reducing complexity.
3. **Improved Context Understanding** – Recognizes UI elements, captions, and interactions in a single step.
4. **Efficient Multi-Step Execution** – Uses **System-2 reasoning** for complex workflows.
5. **Scalability & Adaptability** – Learns iteratively, making it more robust in dynamic web environments

How UI TARS work

- **End-to-End GUI Interaction** – Operates using only screenshots, mimicking human-like interactions.
- **Enhanced Perception & Action Modeling** – Uses large-scale data for accurate GUI understanding and cross-platform standardization.
- **System-2 Reasoning** – Implements task decomposition, reflection, and milestone recognition for complex tasks.
- **Iterative Learning** – Continuously refines actions through self-correction and experience-based adaptation.
- **Benchmark Leader** – Outperforms GPT-4o and Claude in GUI task execution

Our Current Focus

We are trying to setup the UI TARS Desktop Application which is an **AI-powered assistant** designed for automation, data retrieval, and interaction with external sources. It functions similarly to how our WebAgent project is intended to work but in a **desktop environment**.

We are in process of setting that up locally and running to see how that model performs and also going through its source files to see how it uses scrapping functions, LLM functions and Parsing Functions

Issues we are facing

1. The Source code is very extensive so its taking a long time to figure out the modules that are relevant to us and how they are integrated into an application
2. The UI TARS Desktop Application is showing a lot of errors when we are trying to set it up

- Got an error "no module named resource" and apparently this missing module is not available in Windows.
- Tried everything again in Ubuntu instead of Windows, but ran into trouble when setting up vllm, because here we have to slightly modify the commands since we don't have cuda and then it doesn't work without gpu

Next Step

1. We have to create a basic web agent using ui tars and see how it works and get a benchmark test to compare its performance with the previous model.

2. We also are getting to learn LoRa to fine tune our model for different use cases

Goals for next meeting

1. A Basic Web agent capable to using a prompt and hardcoded link of a url to extract and summarize the result.