fellowship.ai

Demo and Status Report: Autonomous Web Agent

- 14.03.2025

Overview

Recent progress

- Progress with the agents
- Benchmarking

What agents are we working on?

As of this week, we have been working on three agents which are the following:

1. Web Automation agent: the one responsible for interpretation as well as completing tasks
2. Visual data scraper: This agent responsible for picking up visual data and sending it to the first agent.
3. Textual data scraper: this agent is responsible for collecting data from the html code of a website and sending it to the agent.

Why Two Scrapers?

Currently our issue currently lies in how Proxy-lite 3b works as a VLM. As a VLM it should be able to process visual inputs, but when we try to input a screenshot of the website it doesn't work. So as of right now we are trying to check if it can get visual context from screenshot based OCR techniques; and if the model is not trained on detecting and labeling web elements it would lead to the scarper failing to perform any actions.

As a back-up plan we have decided that using a textual scraper that collects the HTML elements of the website to get information like button placement and data displayed would be a good move.

Benchmarking

For benchmarking we thought of trying to apply the benchmark notebook that the previous Cohort had used for their agent to save ourselves some effort. However, we noticed that their benchmarking code is not directly applicable to the proxy-lite webagent, so we decided to use their examples without using the notebook itself.

We have also integrated logging functionalities into the proxy-lite agent so that it would store its actions for us to compare its actions to the desired actions.

Benchmarking

However, we are still facing problems when it comes to benchmarking. For example when an agent makes a mistake in one of the steps all the actions that follow it will be different from what is expected from the agent even if it was correct.

This then leads us to evaluating it based on each step rather than the run as a whole. Which caused us to run into some errors at first but as of today we managed to get it to work even on simple tasks. (video)
However, the agent does still run into the cookie banner and not be able to get across it.

Goals for next meeting

- Get the visual scraper to accept visual input
- Start documenting the code