🅭🅯 2024. This work is licensed under a Creative Commons "Attribution 4.0 International" license.

Evaluating Language Models for Open-Ended Tasks in Real Computer Environments

Dongchan Shin

The University of Hong Kong

1 Introduction

Language agents exhibit the capability to utilize natural language for a variety of complex tasks across different environments, particularly when built on large language models (LLMs). Notably, language model agents that can effectively execute intricate computer tasks introduces new research area for autonomous agents and also have the potential to revolutionize human-computer interaction. This report aims to assess language model agents by evaluating their performance on diverse and complex tasks within real computer environments.

2 Experiment

2.1 Settings

In this experiment, models and datasets from Hugging Face are utilized. The Llama-3.2-1B-Instruct model serves as the baseline, while both Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct are fine-tuned using the Mind2Web dataset (Deng et al., 2023). The GPU used for fine-tuning is the P100 available on Kaggle notebooks. Due to limited GPU resources, 8-bit quantization is applied during fine-tuning. The fine-tuned models are accessible at `https://huggingface.co/ShinDC/llama_finetune_mind2web` for the 3B model and `https://huggingface.co/ShinDC/llama_finetune_mind2web_1B`

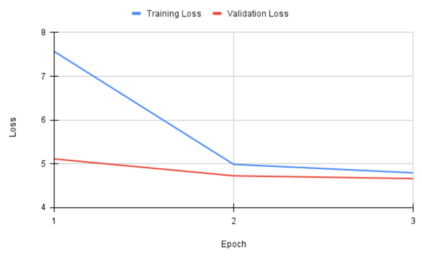


Figure 1: Loss curve for 1B model

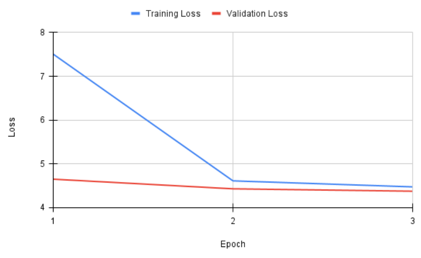


Figure 2: Loss curve for 3B model.

for the 1B model. Figure 1 and 2 show both training and validation loss for Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct respectively. All curves show decreasing trend and loss values of 3B model are smaller than that of 1B model.

Table 1: Fine-tuning Hyperparameters

Hyperparameter	Value
Learning rate	1e-05
Train batch size	2
Validation batch size	2
Seed	42
Optimizer	Adam with betas=(0.9,0.999) and epsilon=1e-08
Loss function	Cross-Entropy Loss
LR scheduler	Linear
Number of epochs	3
Gradient Accumulation Step	1
Weight decay	0.01
temperature	1.0
top p	0.9

Table 1 shows hyperparameters used for fine-tuning. Due to limited GPU memory, batch size of 2 is used for both training and validation and 3 for number of epochs. For learning rate, 1e-05 is applied and Adam optimizer is used for optimization.

Table 2: Experiment Hyperparameters

Hyperparameter	Value
Action space	pyautogui
Observation type	Screenshot
Max steps	15
Max trajectory length	3
Temperature	1.0
Top p	0.9

Main experiment hyperparameters from table 2 include 1.0 for temperature and 0.9 for top p, which are commonly used by many models from leaderboard.

Table 3: Text Input and 3B Fine-tuned Model Output Examples

Text Input$_1$	Task: Find a latest post with more than 10k upvotes in r/announcements community and upvote it.
Model Output$_1$	Task: Find a latest post with more than 10k upvotes in r/announcements community and upvote it.[button] [button] [button][button] ->][button] ->][button] ->][button] ->] ->[button] ->] ->
Text Input$_2$	Task: Join a fitness community.
Model Output$_2$	Task: Join a fitness community. CLICK
Text Input$_3$	Task: Get the weekend wedding first dance lessons in my home at zip 10005 for the beginner in Latin dance for a group of 3-4 people with no choreography or song preferences, and contact for prices on my phone number 5214586525.
Model Output$_3$	Task: Get the weekend wedding first dance lessons in my home at zip 10005 for the beginner in Latin dance for a group of 3-4 people with no choreography or song preferences, and contact for prices on my phone number 5214586525.[button] ->[button]

Three example pairs of text input from test_domain (Cross Domain) data and 3B fine-tuned model output are shown in table2. It is observed that fine-tuned model outputs contain keywords such as CLICK and [button].

2.2 Benchmark

The evaluation employs the OSworld benchmark (Xie et al., 2024) to assess both baseline and fine-tuned models within computer environments. Due to time-limit error, a subset of 312 out of 369 tasks was used.

2.3 Metric

The evaluation metric, drawn from OSWorld (Xie et al., 2024), is based on the success rate for each domain and the overall average.

3 Results

In this section, we present the performance of each model on digital agent tasks.

Table 3: Success rates of baseline model and fine-tuned models in each domain (Input format: Screenshot)

	Success Rate			
Model Domain	Llama-3.2-1B- Instruct(Baseline)	Llama-3.2-1B- Instruct(Finetune)	Llama-3.2-3B- Instruct(Finetune)	Human
Calc	0.0%	0.0%	0.0%	61.7%
Imp	6.52%	2.17%	2.17%	80.85%
Writer	4.35%	4.35%	4.35%	73.91%
VLC	6.53%	6.53%	6.53%	70.59%
VSCode	0.0%	0.0%	0.0%	73.91%
OS	4.17%	4.17%	4.17%	75.0%
ThunBird	0.0%	0.0%	0.0%	46.67%
GIMP	0.0%	0.0%	0.0%	73.08%
Chrome	7.14%	7.14%	7.14%	78.26%
multiapps	2.12%	0.83%	2.12%	73.27%
Overall	2.80%	1.84%	2.16%	72.36%

3.1 Success rate

According to Table 3, the baseline model achieved the highest overall success rate at 2.80%, followed by the 3B fine-tuned model at 2.16% and the 1B fine-tuned model at 1.84%. The majority of the performance gap between the baseline and fine-tuned models is attributed to differences on LibreOffice Impress. However, when compared to human performance, all models exhibit significantly lower success rates, with an approximate 70% absolute difference.
Across domains, models struggled notably, particularly on LibreOffice Calc, VSCode, Thunderbird, and GIMP, where all models achieved a 0.0% success rate. Baseline and fine-tuned models performed similarly on most tasks, indicating a limited effect from fine-tuning.The highest success rate among all models was 7.14% on Chrome, while the lowest human success rate was 46.67% on Thunderbird, still far exceeding the models' performance. The highest human success rate was 80.85% on LibreOffice Impress.

4 Analysis

In this section, we analyze factors that influence the performance of language models in computer agent tasks.

4.1 Computing resources

Given the resource constraints of this experiment, available computing resources, such as GPUs, are insufficient for fine-tuning large language models. Therefore, relatively smaller models (1B and 3B parameters) were selected. To optimize memory efficiency during fine-tuning, adjustments were made, including reducing per-device batch size, shortening context length, and applying 8-bit quantization. Although quantization is commonly used to reduce model size, it has notable drawbacks, such as a loss of precision and information, which can significantly impact model performance.

4.2 Dataset used for fine-tuning

Based on Table 3, the fine-tuned models show lower performance compared to the baseline model, with absolute differences of 0.64% for the 3B model and 0.96% for the

1B model. This suggests that fine-tuning with the Mind2Web dataset and its relatively simple preprocessing approach does not enhance the performance of language models in computer-based tasks. The Mind2Web dataset is specialized for the web domain, primarily designed to follow natural language instructions for completing complex tasks on various websites (Deng et al., 2023). As such, it may not be well-suited for tasks from applications that make up the majority of OSWorld.

4.3 Time-limit error

As mentioned in Section 2.2, this experiment encountered time-limit errors for several tasks. Out of the 369 tasks, only 312 were completed, with the majority coming from Chrome (14 out of 46 executed) and multi-app tasks (77 out of 101 executed). This incomplete execution reduces the credibility of the results, as the analysis lacks full representation across all tasks. Consequently, any insights derived may not accurately reflect the performance or characteristics of the system under different conditions, potentially skewing conclusions.

5 Limitations

5.1 Language model

This experiment is limited in scope by its exclusive use of Llama models. According to the OSWorld leaderboard, models such as Claude, GPT-4, and Gemini demonstrate superior performance compared to Llama models, particularly in processing screenshot inputs. Assessing these alternative models under the same experimental conditions could yield valuable insights and provide a more comprehensive understanding of how different architectures handle similar tasks. Such comparisons may also highlight strengths and weaknesses specific to each model, contributing to a deeper evaluation of their suitability for applications requiring screenshot-based inputs.

5.2 Input method

This experiment has solely focused on screenshot input for evaluating model performance. OSWorld offers four distinct types of observations: A11y tree, screenshot, Screenshot+A11y tree, and Set-of-Mark (Xie et al., 2024). According to the OSWorld leaderboard, Llama-3-70B, which is a significantly larger model than the one used in this experiment, achieved a score of 1.61 on the A11y tree input. Given that this experiment utilized only one of the four observation types, there is potential to expand future experiments to include the remaining observation types. Such an expansion would allow for a more comprehensive evaluation of the model's versatility and its ability to process and interpret different types of input, potentially improving performance across the broader set of observations.

6 Conclusion

In this experiment, we evaluated the performance of the Llama-3.2-1B-Instruct baseline model alongside two variants of the same model fine-tuned on the Mind2Web dataset: Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct. The overall success rates for the baseline model and the fine-tuned models were 2.80%, 1.84%, and 2.16%, respectively. These success rates are significantly lower than the human performance, which

achieved an overall success rate of 72.36%. This stark difference highlights the current limitations of language models and indicates that they are still far from reaching the level of fully autonomous digital agents in complex computer environments.
Despite the relatively low performance in this experiment, there remains considerable room for improvement. For instance, leveraging more powerful computing resources could potentially enhance the fine-tuning process and model performance. Additionally, exploring more suitable and specialized datasets for fine-tuning could lead to better adaptation to specific tasks and environments. These factors present exciting opportunities for future research and development in the quest to improve the capabilities of language models and move closer to the goal of creating highly autonomous digital agents.

References

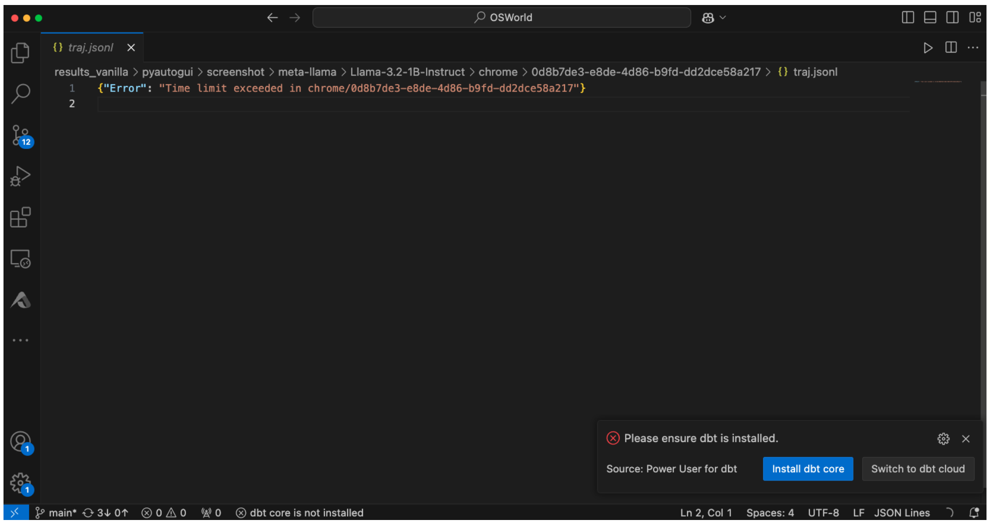
Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2Web: Towards a Generalist Agent for the Web. CoRR, abs/2306.06070, 2023. URL https://doi.org/10.48550/arXiv.2306.06070.

Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T. J., Cheng, Z., Shin, D., Lei, F., Liu, Y., Xu, Y., Zhou, S., Savarese, S., Xiong, C., Zhong, V., and Yu, T. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024.

Appendix

Table 3: Number of completed tasks, total tasks, and execution percentage for each domain

Domain	Completed	Total	Execution percentage
Calc	47	47	100%
Impress	46	47	97.87%
Writer	23	23	100%
VLC	17	17	100%
VSCode	23	23	100%
OS	24	24	100%
ThunBird	15	15	100%
GIMP	26	26	100%
Chrome	14	46	30.43%
Multiapps	77	101	76.24%
Total	312	369	84.55%

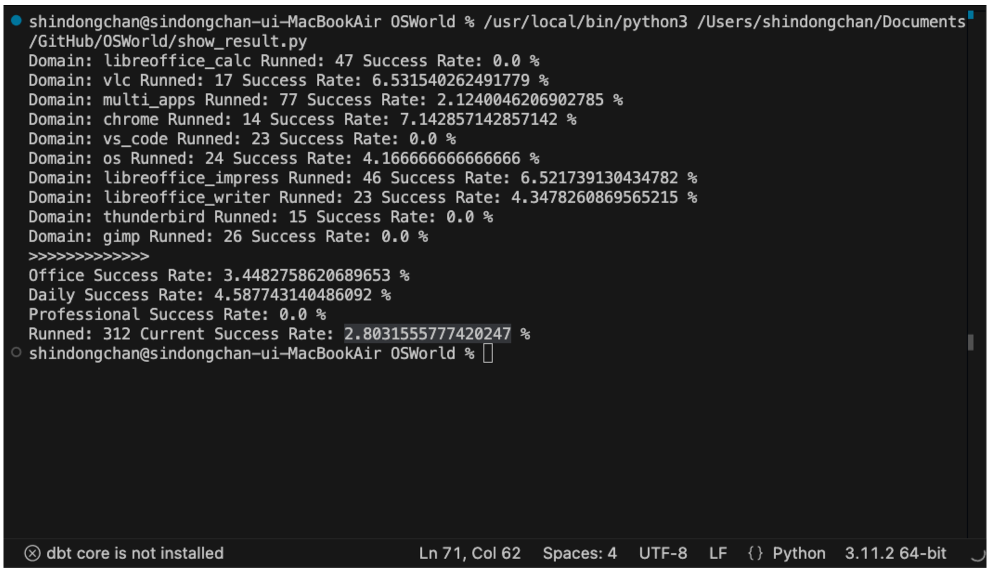

OSWorld
traj.jsonl
results_vanilla > pyautogui > screenshot > meta-llama > Llama-3.2-1B-Instruct > chrome > 0d8b7de3-e8de-4d86-b9fd-dd2dce58a217 > traj.jsonl

```
{"Error": "Time limit exceeded in chrome/0d8b7de3-e8de-4d86-b9fd-dd2dce58a217"}
```

Please ensure dbt is installed.
Source: Power User for dbt

Install dbt core
Switch to dbt cloud
main* 3↓ 0↑ 0 0 0 dbt core is not installed
Ln 2, Col 1 Spaces: 4 UTF-8 LF JSON Lines

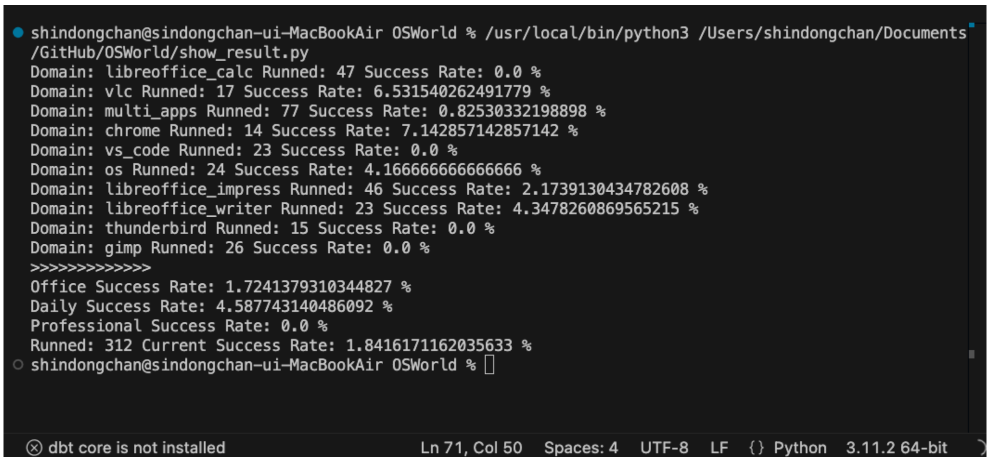
Figure 3: Time-limit error from Chrome



```
shindongchan@sindongchan-ui-MacBookAir OSWorld % /usr/local/bin/python3 /Users/shindongchan/Documents
/GitHub/OSWorld/show_result.py
Domain: libreoffice_calc Runned: 47 Success Rate: 0.0 %
Domain: vlc Runned: 17 Success Rate: 6.531540262491779 %
Domain: multi_apps Runned: 77 Success Rate: 2.1240046206902785 %
Domain: chrome Runned: 14 Success Rate: 7.142857142857142 %
Domain: vs_code Runned: 23 Success Rate: 0.0 %
Domain: os Runned: 24 Success Rate: 4.166666666666666 %
Domain: libreoffice_impress Runned: 46 Success Rate: 6.521739130434782 %
Domain: libreoffice_writer Runned: 23 Success Rate: 4.3478260869565215 %
Domain: thunderbird Runned: 15 Success Rate: 0.0 %
Domain: gimp Runned: 26 Success Rate: 0.0 %
>>>>>>>>>>>>>
Office Success Rate: 3.4482758620689653 %
Daily Success Rate: 4.587743140486092 %
Professional Success Rate: 0.0 %
Runned: 312 Current Success Rate: 2.8031555777420247 %
shindongchan@sindongchan-ui-MacBookAir OSWorld %
```

dbt core is not installed
Ln 71, Col 62 Spaces: 4 UTF-8 LF Python 3.11.2 64-bit

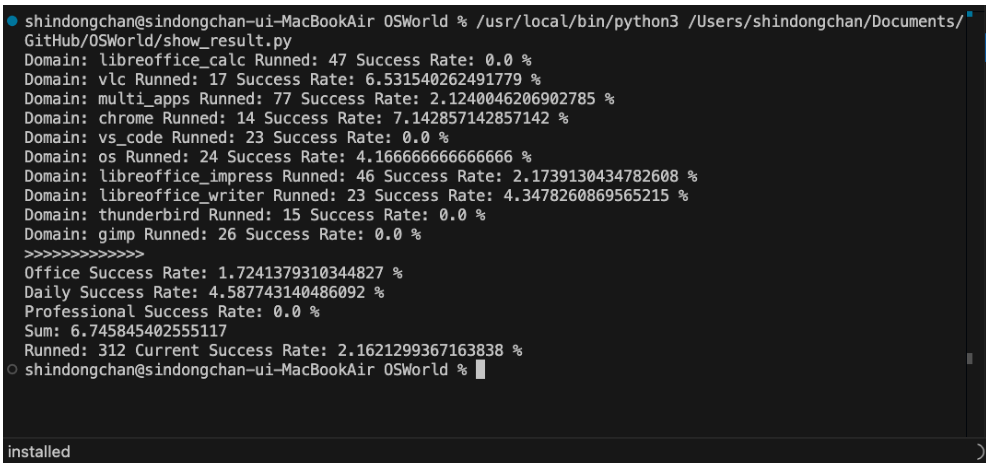
Figure 4: Baseline model results



```
shindongchan@sindongchan-ui-MacBookAir OSWorld % /usr/local/bin/python3 /Users/shindongchan/Documents
/GitHub/OSWorld/show_result.py
Domain: libreoffice_calc Runned: 47 Success Rate: 0.0 %
Domain: vlc Runned: 17 Success Rate: 6.531540262491779 %
Domain: multi_apps Runned: 77 Success Rate: 0.82530332198898 %
Domain: chrome Runned: 14 Success Rate: 7.142857142857142 %
Domain: vs_code Runned: 23 Success Rate: 0.0 %
Domain: os Runned: 24 Success Rate: 4.166666666666666 %
Domain: libreoffice_impress Runned: 46 Success Rate: 2.1739130434782608 %
Domain: libreoffice_writer Runned: 23 Success Rate: 4.3478260869565215 %
Domain: thunderbird Runned: 15 Success Rate: 0.0 %
Domain: gimp Runned: 26 Success Rate: 0.0 %
>>>>>>>>>>>>>
Office Success Rate: 1.7241379310344827 %
Daily Success Rate: 4.587743140486092 %
Professional Success Rate: 0.0 %
Runned: 312 Current Success Rate: 1.8416171162035633 %
shindongchan@sindongchan-ui-MacBookAir OSWorld %
```

dbt core is not installed
Ln 71, Col 50 Spaces: 4 UTF-8 LF Python 3.11.2 64-bit

Figure 5: 1B Fine-tuned model results



```
shindongchan@sindongchan-ui-MacBookAir OSWorld % /usr/local/bin/python3 /Users/shindongchan/Documents/
GitHub/OSWorld/show_result.py
Domain: libreoffice_calc Runned: 47 Success Rate: 0.0 %
Domain: vlc Runned: 17 Success Rate: 6.531540262491779 %
Domain: multi_apps Runned: 77 Success Rate: 2.1240046206902785 %
Domain: chrome Runned: 14 Success Rate: 7.142857142857142 %
Domain: vs_code Runned: 23 Success Rate: 0.0 %
Domain: os Runned: 24 Success Rate: 4.166666666666666 %
Domain: libreoffice_impress Runned: 46 Success Rate: 2.1739130434782608 %
Domain: libreoffice_writer Runned: 23 Success Rate: 4.3478260869565215 %
Domain: thunderbird Runned: 15 Success Rate: 0.0 %
Domain: gimp Runned: 26 Success Rate: 0.0 %
>>>>>>>>>>>>>
Office Success Rate: 1.7241379310344827 %
Daily Success Rate: 4.587743140486092 %
Professional Success Rate: 0.0 %
Sum: 6.745845402555117
Runned: 312 Current Success Rate: 2.1621299367163838 %
shindongchan@sindongchan-ui-MacBookAir OSWorld %
```

Figure 6: 3B Fine-tuned model results