

# Appendices

## A ALGORITHM

In Algorithm 1, we include a formal definitions of our practical algorithm of PAE as presented in Section 3.

---

### Algorithm 1 Proposer-Agent-Evaluator: Practical Algorithm

---

**Require:** Context information  $z_{\mathcal{M}}$ , task proposer  $\hat{C}$ , autonomous evaluator  $\hat{\mathcal{R}}$ .

```

1: Initialize policy  $\pi$  from a pre-trained checkpoint.
2: Initialize replay buffer  $\mathcal{D} \leftarrow \{\}$ .
3: ## Propose tasks based on the context information.
4: Obtain proposal task distribution  $\hat{C}(z_{\mathcal{M}})$ .
5: for each global iteration do
6:   for each trajectory to be collected do
7:     Sample a task from the task proposer  $c \sim \hat{C}(z_{\mathcal{M}})$ .
8:     Reset the environment to obtain the initial observation  $s_0$ 
9:     for each environment step  $t$  do
10:      Sample  $a_t \sim \pi(\cdot|s_t, c)$ ,  $s_{t+1} \sim \mathcal{T}(\cdot|s_t, a_t, c)$ .
11:      if done then
12:        ## Autonomously evaluate the outcome of the agent rollout.
13:         $r_t \leftarrow \hat{\mathcal{R}}(s_t, a_t, c)$ .
14:      else
15:         $r_t \leftarrow 0$ .
16:      end if
17:       $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1}, c)\}$ .
18:    end for
19:  end for
20:  ## Update the agent policy with any RL algorithm.
21:   $\pi \leftarrow \text{RL\_update}(\pi, \mathcal{D})$ 
22: end for

```

---

## B ALL PROMPTS IN THE EXPERIMENTS

For completeness, we include examples of the prompts that we have used in this section. In particular, in Figure 7, we have provided the prompt that we used for the Claude-Sonnet-3 autonomous evaluator to evaluate the success for the task completion for all tasks in WebArena. A similar is used for all tasks in WebVoyager. In Figure 8, 9, 10 we have included the prompts that we used for generating the proposal tasks for each domain. We used the same prompts with 3 additional website screenshots appended to the messages for PAE + User Demos. It is worth noting that our task proposers are domain-general and have little domain customizations. In particular, for all 13 real-world websites from WebVoyager, we use the same prompt to generate tasks except with the placeholder of “web\_name”. This shows that our PAE framework can easily scale to multiple websites without the need for domain-specific knowledge. The prompt for zero-shot VLM agents are included in Figure 11, 12, and 13.

## C PROMPTS FOR ZERO-SHOT VLM AGENTS

We also append the prompts (Figure 11, 12, and 13) that we used for the zero-shot baselines including Claude-Sonnet-3, Claude-Sonnet-3.5, Qwen2VL, InternVL2b5, LLaVa-1.6-7B, and LLaVa-1.6-34B. The prompt for WebVoyager tasks largely follow from that used in the prior literature (He et al., 2024). We include additional necessary domain knowledge of the WebArena tasks and evaluation protocols in the prompt that we used for WebArena.

810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863

**Autonomous Evaluator Prompt**

You are an expert in evaluating the performance of a web navigation agent. The agent is designed to help a human user navigate a website to complete a task. Your goal is to decide whether the agent’s execution is successful or not.

As an evaluator, you will be presented with three primary components to assist you in your role:

1. **Web Task Instruction:** This is a clear and specific directive provided in natural language, detailing the online activity to be carried out.
2. **Result Response:** This is a textual response obtained after the execution of the web task. It serves as textual result in response to the instruction.
3. **Result Screenshots:** This is a visual representation of the screen showing the result or intermediate state of performing a web task. It serves as visual proof of the actions taken in response to the instruction.

- You **SHOULD NOT** make assumptions based on information not presented in the screenshot when comparing it to the instructions.
- Your primary responsibility is to conduct a thorough assessment of the web task instruction against the outcome depicted in the screenshot and in the response, evaluating whether the actions taken align with the given instructions.
- **NOTE** that the instruction may involve more than one task, for example, locating the garage and summarizing the review. Failing to complete either task, such as not providing a summary, should be considered unsuccessful.
- **NOTE** that the screenshot is authentic, but the response provided by LLM is generated at the end of web browsing, and there may be discrepancies between the text and the screenshots.
- **NOTE** that if the content in the Result response is not mentioned on or different from the screenshot, mark it as not success.

You should explicitly consider the following criteria:

- Whether the claims in the response can be verified by the screenshot. E.g. if the response claims the distance between two places, the screenshot should show the direction. **YOU SHOULD EXPECT THAT THERE IS A HIGH CHANCE THAT THE AGENT WILL MAKE UP AN ANSWER NOT VERIFIED BY THE SCREENSHOT.**
- Whether the agent completes **EXACTLY** what the task asks for. E.g. if the task asks to find a specific place, the agent should not find a similar place.

In your responses: You should first provide thoughts **EXPLICITLY VERIFY ALL THREE CRITERIONS** and then provide a definitive verdict on whether the task has been successfully accomplished, either as ‘**SUCCESS**’ or ‘**NOT SUCCESS**’.

A task is ‘**SUCCESS**’ only when all of the criteria are met. If any of the criteria are not met, the task should be considered ‘**NOT SUCCESS**’.

Figure 7: The prompt used by the autonomous evaluator for Claude-Sonnet-3. Same prompt is used to evaluate tasks from WebArena websites. The evaluator takes as inputs the task description, the response from the agent’s ANSWER action, and last three screenshots in the trajectory. The evaluation result is a binary verdict of ‘SUCCESS’ or ‘NOT SUCCESS’.



864 **Task Proposer Prompt for WebVoyager**  
865 {"web\_name": "Apple", "id": "Apple-40", "ques": "Find the pricing and specifications  
866 for the latest Mac Studio model, including the available CPU and GPU options.", "web":  
867 "https://www.apple.com/"}

868 We are training a model to navigate the web. We need your help to generate instructions. With the  
869 examples provided above, please give 25 more example tasks for the model to learn from in the  
870 domain of {web\_name}. You should imagine tasks that are likely proposed by a most likely user of  
871 this website. A few demos of users navigating through the web are provided above.  
872 YOU SHOULD MAKE USE OF THE DEMOS PROVIDES TO GENERATE TASKS, SO THAT  
873 YOUR TASKS ARE REALISTIC AND RELEVANT TO THE WEBSITE.  
874 Please follow the corresponding guidelines:  
875 1)First output your thoughts first on how you should come up with diverse tasks that examine various  
876 capabilities on the particular website, and how these tasks reflect the need of the potential user. Then  
877 you should say 'Output:' and then followed by the outputs STRUCTURED IN JSONL FORMAT.  
878 You should not say anything else in the response.  
879 2)PLEASE MAKE SURE TO HAVE 25 examples in the response!!!  
880 3)Your proposed tasks should be DIVERSE AND COVER A WIDE RANGE OF DIFFERENT  
881 POSSIBILITIES AND DIFFICULTY in the domain of {web\_name}. Remember, your job is to  
882 propose tasks that will help the model learn to navigate the web to deal with various real world  
883 requests.  
884 4)Your task should be objective and unambiguous. The carry-out of the task should NOT BE DE-  
885 PENDENT on the user's personal information such as the CURRENT TIME OR LOCATION.  
886 5)You should express your tasks in as diverse expressions as possible to help the model learn to  
887 understand different ways of expressing the same task.  
888 6)Your tasks should be able to be evaluated OBJECTIVELY. That is, by looking at the last three  
889 screenshots and the answer provided by an agent, it should be possible to tell without ambiguity  
890 whether the task was completed successfully or not.  
891 7)Your tasks should require a minimum completion steps from 3 to 7 steps, your tasks should have a  
892 diverse coverage in difficulty as measured by the minimum completion step. I.E. You should propose  
893 not only tasks that may take more than 4 steps to complete but also tasks that can be completed within  
894 3 steps.  
895 8)Humans should have a 100% success rate in completing the task.  
896 9)Your tasks should be able to be completed without having to sign in to the website.

903 Figure 8: Prompts used by Claude-Sonnet-3 for proposing tasks in WebVoyager experiments. For  
904 PAE + User Demos, we use the same prompt with additional user demos appended to the message.

## 907 D DETAILS FOR SFT

909 **SFT for WebVoyager.** As shown in Table [1](#), unlike proprietary VLMs, none of the open-source  
910 VLM agent is able to follow the instructions and achieve non-trivial performances in real-world web  
911 navigation tasks in the zero-shot manner. Such models can rarely get success rewards in the process  
912 of RL, thus leading to very slow convergence. To “warm-up” the open-source VLM agent to achieve  
913 a non-trivial performance at the start of RL training, we turn to enhancing the performances with  
914 SFT before RL. Note that the SFT process may not be needed if the base VLM agent model can  
915 already achieve non-trivial performances such as Claude 3 Sonnet. To prevent data contamination,  
916 we gather 85 out-of-distribution real-world websites (listed in Figure [14](#) and [15](#)), and collect 11220  
917 trajectories in total using Claude 3 Sonnet with the prompt specified in Figure [11](#). The average  
trajectory success rate is 25% as measured by our Claude 3 Sonnet evaluator. Each action in the

918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971

**Task Proposer Prompt for WebArena Map**

```
{ "web_name": "map", "id": "map-2", "ques": "Tell me the full address of all international airports that are within a driving distance of 50 km to University of California, Berkeley" }
```

```
{ "web_name": "map", "id": "map-10", "ques": "I will arrive San Francisco Airport soon. Provide the name of a Hilton hotel in the vicinity, if available. Then, tell me the the shortest walking distance to a supermarket from the hotel." }
```

```
{ "web_name": "map", "id": "map-17", "ques": "Check if the ikea in pittsburgh can be reached in one hour by car from hobart street" }
```

We are training a model to navigate the web. We need your help to generate instructions. With the examples provided above, please give 25 more example tasks for the model to learn from in the domain of OpenStreetMap. You should imagine who is the most likely user for the website and propose tasks that are likely to be proposed by this user. Please follow the corresponding guidelines:

- 1) First output your thoughts first on how you should come up with diverse tasks that examine various capabilities on the particular website, and how these tasks reflect the need of the potential user. Then you should say 'Output:' and then followed by the outputs STRUCTURED IN JSONL FORMAT. You should not say anything else in the response.
- 2) PLEASE MAKE SURE TO HAVE 25 examples in the response!!!
- 3) Your proposed tasks should be DIVERSE AND COVER A WIDE RANGE OF DIFFERENT POSSIBILITIES AND DIFFICULTY in the domain of OpenStreetMap. Remember, your job is to propose tasks that will help the model learn to navigate the web to deal with various real world requests.
- 4) Your task should be objective and unambiguous. The carry-out of the task should NOT BE DEPENDENT on the user's personal information such as the CURRENT TIME OR LOCATION.
- 5) You should express your tasks in as diverse expressions as possible to help the model learn to understand different ways of expressing the same task.
- 6) Your tasks should be able to be evaluated OBJECTIVELY. That is, by looking at the last three screenshots and the answer provided by an agent, it should be possible to tell without ambiguity whether the task was completed successfully or not.
- 7) Your tasks should require a minimum completion steps from 3 to 7 steps, your tasks should have a diverse coverage in difficulty as measured by the minimum completion step. I.E. You should propose not only tasks that may take more than 4 steps to complete but also tasks that can be completed within 3 steps.
- 8) Humans should have a 100% success rate in completing the task.
- 9) Your tasks should be able to be completed without having to sign in to the website.

Figure 9: Prompts used by Claude-Sonnet-3 for proposing WebArena Tasks for Map. For PAE + User Demos, we use the same prompt with additional user demos appended to the message. For this domain only, we provide three hand-written in-domain examples to set the right difficulty for the task proposer. Such in-domain examples are not needed for all other domains including Reddit and OneStopMarket, and other real-world WebVoyager websites.

trajectories contains both thoughts and actual web actions shown in Figure 2. All 11220 trajectories are used for SFT. RL training is carried out on top of the SFT checkpoint.

**SFT for WebArena.** In our preliminary experiments, we found that the SFT checkpoint trained on real-world websites do not generalize well to self-hosted websites on WebArena. This is potentially because of the distribution shift between real-world commercial websites and self-hosted websites. For example, most real-world map websites such as Google Maps and Apple Maps support advanced fuzzy search capabilities such as "pittsburgh to new york" while OpenStreetMap from WebArena will not return any results with such queries. Therefore, we collect 3000 Claude 3 Sonnet generated trajectories each from OpenStreetMap, Reddit, and OneStopMarket websites from WebArena. We

972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025

**Task Proposer Prompt for WebArena Reddit and OneStopMarket**

```
{“web_name”: “Apple”, “id”: “Apple-40”, “ques”: “Find the pricing and specifications for the latest Mac Studio model, including the available CPU and GPU options.”, “web”: “https://www.apple.com/”}
```

We are training a model to navigate the web. We need your help to generate instructions. With the examples provided above, please give 25 more example tasks for the model to learn from in the domain of {web\_name}.

You should provide tasks in the DOMAIN OF {web\_name}.

Please follow the corresponding guidelines: 1)First answer how many screenshots are provided and describe in detail the functions of the website that you see from each of the screenshot. Then output your thoughts first. Then you should say ‘Output:’ and then followed by the outputs STRUCTURED IN JSONL FORMAT. You should not say anything else in the response.

2)PLEASE MAKE SURE TO HAVE 25 examples in the response!!!

4)Your task should start from the home page of the website instead of the shown screenshots.

5)Your task does not need to be the same as real users would do, but it should examine diverse capabilities of the agent to do web navigation.

6)Your tasks should examine the VERY BASIC functions of the website and should not require complicated web page operations. They can be completed within 5 steps.

7)THIS DOMAIN IS A SELF-HOSTED STATIC DOMAIN AND DIFFERENT FROM POPULAR WEBSITES, DO NOT ASSUME ANY INFORMATION NOT PROVIDED IN THE SCREENSHOTS.

8)Your tasks should examine the capability of the web agent to find some information on the website, navigating to some specific web pages. Do not propose tasks that involve making actual modifications to the websites.

9)Your tasks should result in the agent landing in a single groundtruth web page or finding a single ground truth answer. The landed webpage can be some specific categories, a drafted post, some search results, or even the homepage of the website. When the task is to find some information, specify exactly what information the agent should find such as the price, the number of comments, the title, etc. It can also be information about the current account.

Figure 10: Prompts used by Claude-Sonnet-3 for proposing WebArena Tasks for Reddit and OneStopMarket. For PAE + User Demos, we use the same prompt with additional user demos appended to the message.

use the prompts from Figure 12 and 13 for the Claude agent. The average trajectory success rate is 27% as measured by our Claude 3 Sonnet evaluator. The SFT checkpoint for WebArena is fine-tuned from the SFT checkpoint for WebVoyager.

## E ADDITIONAL RESULTS ON WEBARENA

For completeness, we have also provided additional experiment results of different models from Table 2 in the original task split of WebArena (Zhou et al., 2024). As shown in the comparison results presented in Table 4, even SOTA proprietary VLM agents like Claude 3 Sonnet struggle with the tasks in WebArena with a success rate of only 14.6% with set-of-marks observations and chain-of-thought prompting. After performing SFT using the demonstrations generated by Claude 3 Sonnet, LLaVa-7B SFT can only achieve 1.4% and 5.8% success rate on PostMill and OneStopMarket. By manually inspecting the roll-out trajectories generated by LLaVa-SFT, we found that around half of the successful trajectories on those two websites are false positives from the WebArena evaluator. In these trajectories, the agent simply guessed the answer to be “no” or “N/A” where the ground truth happens to be that the task is not executable. As a result, the actual success rate on those two websites

1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079

### Zero-Shot VLM Agent Prompt for WebVoyager (1/2)

Imagine you are a robot browsing the web, just like humans. Now you need to complete a task. In each iteration, you will receive an Observation that includes a screenshot of a webpage and some texts. This screenshot will feature Numerical Labels placed in the TOP LEFT corner of each Web Element. Carefully analyze the visual information to identify the Numerical Label corresponding to the Web Element that requires interaction, then follow the guidelines and choose one of the following actions:

1. Click a Web Element.
2. Delete existing content in a textbox and then type content.
3. Scroll up or down. Multiple scrolls are allowed to browse the webpage. Pay attention!! The default scroll is the whole window. If the scroll widget is located in a certain area of the webpage, then you have to specify a Web Element in that area. I would hover the mouse there and then scroll.
4. Wait. Typically used to wait for unfinished webpage processes, with a duration of 5 seconds.
5. Go back, returning to the previous webpage.
6. Google, directly jump to the Google search page. When you can't find information in some websites, try starting over with Google.
7. Answer. This action should only be chosen when all questions in the task have been solved.

Correspondingly, Action should STRICTLY follow the format:

- Click [Numerical\_Label]
- Type [Numerical\_Label]; [Content]
- Scroll [Numerical\_Label or WINDOW]; [up or down]
- Wait
- GoBack
- Google
- ANSWER; [content]

Key Guidelines You MUST follow:

\* Action guidelines \*

- 1) To input text, NO need to click textbox first, directly type content. After typing, the system automatically hits 'ENTER' key. Sometimes you should click the search button to apply search filters. Try to use simple language when searching.
- 2) You must Distinguish between textbox and search button, don't type content into the button! If no textbox is found, you may need to click the search button first before the textbox is displayed.
- 3) Execute only one action per iteration.
- 4) STRICTLY Avoid repeating the same action if the webpage remains unchanged. You may have selected the wrong web element or numerical label. Continuous use of the Wait is also NOT allowed.
- 5) When a complex Task involves multiple questions or steps, select "ANSWER" only at the very end, after addressing all of these questions (steps). Flexibly combine your own abilities with the information in the web page. Double check the formatting requirements in the task when ANSWER.

\* Web Browsing Guidelines \*

- 1) Don't interact with useless web elements like Login, Sign-in, donation that appear in Webpages. Pay attention to Key Web Elements like search textbox and menu.
- 2) Visit video websites like YouTube is allowed BUT you can't play videos. Clicking to download PDF is allowed and will be analyzed by the Assistant API.
- 3) Focus on the numerical labels in the TOP LEFT corner of each rectangle (element). Ensure you don't mix them up with other numbers (e.g. Calendar) on the page.
- 4) Focus on the date in task, you must look for results that match the date. It may be necessary to find the correct year, month and day at calendar.
- 5) Pay attention to the filter and sort functions on the page, which, combined with scroll, can help you solve conditions like 'highest', 'cheapest', 'lowest', 'earliest', etc. Try your best to find the answer that best fits the task.

Your reply should strictly follow the format:

Thought: {Your brief thoughts (briefly summarize the info that will help ANSWER)}  
Action: {One Action format you choose}

Then the User will provide:

Observation: {A labeled screenshot Given by User}

Figure 11: The prompt used for all zero-shot VLM agents for WebVoyager websites, including Claude-Sonnet-3, Claude-Sonnet-3.4, Qwen2-VL, InternVL-2.5-XComposer, LLaVa-1.6-7B, and LLaVa-1.6-34B.

1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133

### Zero-Shot VLM Agent Prompt for WebArena (2/2)

Imagine you are a robot browsing the web, just like humans. Now you need to complete a task. In each iteration, you will receive an Observation that includes a screenshot of a webpage, some texts and the accessibility tree of the webpage. This screenshot will feature Numerical Labels placed in the TOP LEFT corner of each Web Element. The accessibility tree contains information about the web elements and their properties. The numerical labels in the screenshot correspond to the web elements in the accessibility tree.

Carefully analyze the visual information to identify the Numerical Label corresponding to the Web Element that requires interaction, then follow the guidelines and choose one of the following actions:

1. Click a Web Element.
2. Delete existing content in a textbox and then type content.
3. Scroll up or down. Multiple scrolls are allowed to browse the webpage. Pay attention!! The default scroll is the whole window. If the scroll widget is located in a certain area of the webpage, then you have to specify a Web Element in that area. I would hover the mouse there and then scroll.
4. Wait. Typically used to wait for unfinished webpage processes, with a duration of 5 seconds.
5. Go back, returning to the previous webpage.
6. Answer. This action should only be chosen when all questions in the task have been solved.

Correspondingly, Action should STRICTLY follow the format:

- Click [Numerical\_Label]
- Type [Numerical\_Label]; [Content]
- Scroll [Numerical\_Label or WINDOW]; [up or down]
- Wait
- GoBack
- ANSWER; [content]

Key Guidelines You MUST follow:

\* Action guidelines \*

- 1) To input text, NO need to click textbox first, directly type content. After typing, the system automatically hits 'ENTER' key. Sometimes you should click the search button to apply search filters. Try to use simple language when searching.
- 2) You must Distinguish between textbox and search button, don't type content into the button! If no textbox is found, you may need to click the search button first before the textbox is displayed.
- 3) Execute only one action per iteration.
- 4) STRICTLY Avoid repeating the same action if the webpage remains unchanged. You may have selected the wrong web element or numerical label. Continuous use of the Wait is also NOT allowed.
- 5) When a complex Task involves multiple questions or steps, select "ANSWER" only at the very end, after addressing all of these questions (steps). Flexibly combine your own abilities with the information in the web page. Double check the formatting requirements in the task when ANSWER.
- 6) If you can't find the answer using the given website because there is no such information on the website after some attempts, you should report "N/A" as the answer to represent that the task is impossible to solve with the given website. You may have 15 steps to try to solve the task.
- 7) Only provide answer based on the information from the image, make sure the answer is consistent with the image, don't hallucinate any information that is not based on image.

\* Web Browsing Guidelines \*

- 1) Focus on the numerical labels in the TOP LEFT corner of each rectangle (element). Ensure you don't mix them up with other numbers (e.g. Calendar) on the page.
- 2) Pay attention to the filter and sort functions on the page, which, combined with scroll, can help you solve conditions like 'highest', 'cheapest', 'lowest', 'earliest', etc. Try your best to find the answer that best fits the task.

Figure 12: The prompt used for all zero-shot VLM agents for WebArena websites, including Claude-Sonnet-3, Claude-Sonnet-3.4, Qwen2-VL, InternVL-2.5-XComposer, LLaVa-1.6-7B, and LLaVa-1.6-34B. To be continued in Figure [13](#).

1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187

### Zero-Shot VLM Agent Prompt for WebArena

#### \* OpenStreetMap Usage Guidelines \*

- 1) When you need to search the address of a location, you can just type the location in the 'search' bar. You don't need to use the directions button to get the address. The directions button is only used when you need to find the distance/walk/drive time between two locations.
- 2) When you are trying to search for a location, you may get no results. This is because the OpenStreetMap does not support approximate search. You may try to search some alternative keywords or try to find the location by yourself. Note that openstreet map does not support search phrase like 'Cafe near CMU', you should try to find it by yourself.
- 3) When you need to find the distance/walk/drive time between two locations, you should FIRST CLICK ON THE DIRECTIONS BUTTON (drawn as two arrows), to the right of the 'Go' Button and usually labeled as [10] or [11]. AND ONLY INPUTTING THE TWO LOCATIONS AFTER CLICKING ON THE DIRECTIONS BUTTON WHEN THE DIRECTIONS SEARCH BARS ARE SHOWN.
- 4) When you are trying to type some locations in the directions search bar, sometimes you may receive an alert of 'couldn't locate' followed by the location you typed. This means the location you typed is not found in the map. Do not immediately try something else. You need to quit the direction and find the precise name of this location by searching it in the map first.
- 5) When you search the walk/drive/bike time, make sure that you are USING THE RIGHT MODE OF TRANSPORTATION. The default mode is usually set to 'Drive'.
- 6) When you need to get the DD of some location, you need to click the location shown in the search result in the left part of the screen. The DD will be shown then starting with 'Location:'.
- 7) When you need to answer the zip code of some location, you should directly answer the 5-digit zip code. The answer should be "15232" instead of "The zip code of the location is 15232". Note that the zipcode will be displayed in the search result, you don't need to click the location to the information page to find the zip code.
- 8) When you need to answer the phone of some location, please omit the part of the country code. The answer should be "4122683259" instead of "+1 412 268 3259".

#### \* Reddit Usage Guidelines \*

- 1) You are already in the reddit website, though you may not see the 'reddit' in any part of the screenshot. You do not need to further navigate to the reddit website.
- 2) When you want to find a subreddit, you need to first navigate to Forums to see the list of subreddits. Under forums, you will see only a subset of subreddits. To get the full list of subreddits, you need to navigate to the Alphabetical option. To know you can see the full list of subreddits, you will see 'All Forums' in the observation. Often you will not find a focused subreddit that exactly matches your query. In that case, go ahead with the closest relevant subreddit. To know that you have reached a subreddit successfully, you will see '/f/subreddit\_name' in the observation.
- 3) When you want to post forum in reddit, remember to fill up all the content, then click the button 'Create forum'. The button maybe located below out of the screenshot, you need to scroll down to find it.
- 4) When you want to ask or post something in a subreddit, you need to first find that subreddit and then finish the work. 5) forums and subreddits are the same thing.

Your reply should strictly follow the format:

Thought: Your brief thoughts (briefly summarize the info that will help ANSWER)

Action: One Action format you choose

Then the User will provide:

Observation: A labeled screenshot Given by User

Remember only execute one action in each step. For example, 'Action: Type [8]; CMU, Type [9] Pittsburgh' is not allowed. You should execute the action 'Type [8]; CMU' first, then 'Type [9] Pittsburgh' in the next step.

Remember to always make your answer simple and clear. For example, if you want to report the zip code of some location, always say "ANSWER; 06516" instead of "The zip code of the location is 06516".

Figure 13: The prompt used for all zero-shot VLM agents for WebArena websites, including Claude-Sonnet-3, Claude-Sonnet-3.4, Qwen2-VL, InternVL-2.5-XComposer, LLaVa-1.6-7B, and LLaVa-1.6-34B. Continued from Figure [12](#)



is lower than 2%, leaving very sparse reward signals for RL to make meaningful improvements. We therefore rewrote the tasks on PostMill and OneStopMarket to be easier and report the performances of PAE in Table 2.

		OpenStreetMap	PostMill	OneStopMarket	Average
<i>Proprietary</i>	Claude 3 Sonnet	24.3	10.6	11.2	14.6
	Qwen2VL-7B	0.7	0.0	1.3	0.7
<i>Open-source</i>	InternVL2.5-8B	2.6	0.2	3.3	2.3
	LLaVa-7B	0.0	0.0	0.0	0.0
<i>Ours</i>	LLaVa-7B SFT	15.2	1.4	5.8	7.2

Table 4: Success rate comparisons across different domains from WebArena. Success and failure are detected with ground-truth verification functions. All tasks from OpenStreetMap are kept unchanged from WebArena task splits.

## F LIMITATIONS

Despite the progress of PAE for open-source VLM agents, there are still some limitations due to practical constraints. First of all, due to the limitations in fundamental capabilities of open-source base VLM models, our models trained with PAE are still inferior to state-of-the-art proprietary models in realistic web navigations, where advanced reasoning and planning capabilities are required. Moreover, because of the hallucination issues of open-source VLMs, we found them unreliable to serve as the autonomous evaluators and had to rely on advanced proprietary VLMs for judging the success and providing rewards. Finally, because of the dynamic nature of the real websites that we are using, some of our results may not be produced exactly, although a significant improvement from PAE should still be observed.

## G HYPERPARAMETERS

We include the hyperparameters that we have used in Table 5. As shown in the table, the only hyperparameters that PAE have on top of standard supervised fine-tuning are number of trajectories to collect in each global iteration in Algorithm 1, number of proposed tasks from the task proposer before RL training, and the number of seen screenshots for the evaluator. In our experiments, we found that PAE is relatively not sensitive to the choices of these hyperparameters, showing the robustness of PAE.

## H MORE QUALITATIVE EXAMPLES

In this section, we present additional qualitative examples of agent trajectories while performing tasks to further demonstrate the effectiveness of our PAE. We will also release the full dataset for further analysis.

**Full trajectories of examples in Section 6.** Here, we provide the complete trajectories for the examples discussed in the qualitative comparisons in Section 6, as shown in Figures 16-19. We detail the agent’s thoughts and actions at each time step throughout the entire trajectory.

**Some representative successful trajectories.** We also showcase representative successful trajectories generated by the LLaVa-7B PAE model to highlight the strengths of our method. In Figure 20, the task is “Show the most played games on Steam, and tell me the number of players currently in-game.” In Figure 21, the task is “Find out the starting price for the most recent model of the iMac on the Apple website.” In Figure 22, the task is “Look up the use of modal verbs in the grammar section for expressing possibility (e.g., ‘might’, ‘could’, ‘may’) and find examples of their usage in sentences on the Cambridge Dictionary.” Finally, in Figure 23, the task is “Search for plumbers available now but not open 24 hours in Orlando, FL.”

Table 5: Hyperparameters for All Experiments

Environment	Hyperparameter	Considered	Chosen
WebVoyager	learning rate	{2e-5, 5e-5, 2e-4}	2e-5
	rollout trajectories	{512, 1024, 2048, 4096}	4096
	rollout temperature	{0.4, 1.0, 2.0}	1.0
	maximum gradient norm	{0.01}	0.01
	actor updates epochs per iteration	{1, 2, 4, 8, 20}	4
	batch size	{8}	8
	gradient accumulation size	{16, 32}	32
	number of proposed tasks	{10000, 50000, 100000}	100000
	number of seen screenshots for evaluator	{1, 3}	3
WebArena Easy	learning rate	{2e-5, 5e-5, 2e-4}	2e-5
	rollout trajectories	{512, 1024, 2048, 4096}	2048
	rollout temperature	{0.4, 1.0, 2.0}	1.0
	maximum gradient norm	{0.01}	0.01
	actor updates epochs per iteration	{1, 2, 4, 8, 20}	2
	batch size	{8}	8
	gradient accumulation size	{16, 32}	32
	number of proposed tasks	{10000, 30000, 100000}	30000
	number of seen screenshots for evaluator	{1, 3}	3

Table 6: Hyperparameters for PAE for WebVoyager and WebArena Easy experiments.

**Detailed explanations of the error type definitions.** To clarify the precise definition of the different error categories used in Section 6, we provide more comprehensive explanations with example trajectories:

(1) **Low-level skill missing errors** refer to cases where the agent has a reasonable plan to solve the problem but fails to execute precise actions on the website, such as not knowing which button to click to reach the desired page. We classify trajectories where the agent seems to follow a reasonable plan but struggles with specific operations into this category. For example, in Figure 24, the task is “Find the Easy Vegetarian Spinach Lasagna recipe on Allrecipes and tell me what the latest review says.” The agent attempts to search for the desired item but fails to click the correct button to reach the detailed page in the search results.

(2) **High-level planning or reasoning errors** occur when the agent fails to generate a complete plan or cannot reason correctly with the website’s screenshots to solve the task. Trajectories where the agent cannot devise a plan for complex tasks or misinterprets the screenshot’s content are categorized as such. For instance, in Figure 25, the task is “Give 12 lbs of 4-cyanoindole, converted to molar and indicate the percentage of C, H, N.” The agent should first search on Google about the chemical definition of 4-cyanoindole, then use WolframAlpha to calculate the result. However, the agent fails to get the precise definition of 4-cyanoindole, and doesn’t know how to solve the task.

(3) **Visual hallucinations** refer to instances where the agent generates fabricated responses not supported by the screenshot. The agent might, for example, claim to have found a requested product while still on the Google homepage or provide an incorrect answer even when on the correct page. In Figure 26, the task is “Find out the trade-in value for an iPhone 13 Pro Max in good condition on the Apple website”. The agent claims with a very detailed answer but actually it never access any page related to the trade-in on the website.

(4) **Timeouts** occur when the agent is on the right track to solving the task but cannot complete it within the maximum number of steps. This error indicates that the agent did nothing wrong but was constrained by the environment’s step limits. For example, in Figure 27, the task is “Go to the Plus section of Cambridge Dictionary, find Image quizzes, and complete an easy quiz about Animals. Tell me your final score.” The agent reaches the maximum time step limit (10) while attempting to finish the quiz.



1296 (5) **Technical issues** are not caused by the agent but by environmental problems, such as websites  
1297 being down or connection failures. In Figure 28, the ChromeDriver crashes after a valid operation.  
1298  
1299 (6) **Others** include less frequent error types, such as when the task itself is impossible to complete.  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349

1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403

<b>Out-of-distribution Websites of WebVoyager for SFT (1/2)</b>	
	<b>Allrecipes:</b>
	Simply Recipes: <a href="https://www.simplyrecipes.com">https://www.simplyrecipes.com</a>
	Food Network: <a href="https://www.foodnetwork.com">https://www.foodnetwork.com</a>
	Taste of Home: <a href="https://www.tasteofhome.com">https://www.tasteofhome.com</a>
	Yummly: <a href="https://www.yummly.com">https://www.yummly.com</a>
	Food.com: <a href="https://www.food.com">https://www.food.com</a>
	<b>Amazon:</b>
	eBay: <a href="https://www.ebay.com">https://www.ebay.com</a>
	Walmart: <a href="https://www.walmart.com">https://www.walmart.com</a>
	Target: <a href="https://www.target.com">https://www.target.com</a>
	Best Buy: <a href="https://www.bestbuy.com">https://www.bestbuy.com</a>
	Alibaba: <a href="https://www.alibaba.com">https://www.alibaba.com</a>
	<b>Apple:</b>
	Samsung: <a href="https://www.samsung.com">https://www.samsung.com</a>
	Microsoft: <a href="https://www.microsoft.com">https://www.microsoft.com</a>
	Sony: <a href="https://www.sony.com">https://www.sony.com</a>
	Google Store: <a href="https://store.google.com">https://store.google.com</a>
	Dell: <a href="https://www.dell.com">https://www.dell.com</a>
	<b>ArXiv:</b>
	SSRN: <a href="https://www.ssrn.com">https://www.ssrn.com</a>
	ResearchGate: <a href="https://www.researchgate.net">https://www.researchgate.net</a>
	bioRxiv: <a href="https://www.biorxiv.org">https://www.biorxiv.org</a>
	IEEE Xplore: <a href="https://ieeexplore.ieee.org">https://ieeexplore.ieee.org</a>
	PubMed: <a href="https://pubmed.ncbi.nlm.nih.gov">https://pubmed.ncbi.nlm.nih.gov</a>
	<b>GitHub:</b>
	GitLab: <a href="https://about.gitlab.com">https://about.gitlab.com</a>
	Bitbucket: <a href="https://bitbucket.org">https://bitbucket.org</a>
	SourceForge: <a href="https://sourceforge.net">https://sourceforge.net</a>
	Codebase: <a href="https://www.codebasehq.com">https://www.codebasehq.com</a>
	Gitea: <a href="https://gitea.io">https://gitea.io</a>
	<b>ESPN:</b>
	CBS Sports: <a href="https://www.cbssports.com">https://www.cbssports.com</a>
	Fox Sports: <a href="https://www.foxsports.com">https://www.foxsports.com</a>
	NBC Sports: <a href="https://www.nbcsports.com">https://www.nbcsports.com</a>
	Bleacher Report: <a href="https://www.bleacherreport.com">https://www.bleacherreport.com</a>
	Sky Sports: <a href="https://www.skysports.com">https://www.skysports.com</a>
	<b>Coursera:</b>
	edX: <a href="https://www.edx.org">https://www.edx.org</a>
	Udacity: <a href="https://www.udacity.com">https://www.udacity.com</a>
	Udemy: <a href="https://www.udemy.com">https://www.udemy.com</a>
	FutureLearn: <a href="https://www.futurelearn.com">https://www.futurelearn.com</a>
	Khan Academy: <a href="https://www.khanacademy.org">https://www.khanacademy.org</a>

Figure 14: A list of 85 websites that we used to collect demonstration trajectories with Claude 3 Sonnet. In total 11220 trajectories were collected with different tasks. These websites were also used for testing the zeroshot generalization of PAE to out-of-distribution websites in Section 5. List continued in Figure 15.

1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457

### Out-of-distribution Websites of WebVoyager for SFT (2/2)

#### Cambridge Dictionary:

Merriam-Webster: <https://www.merriam-webster.com>

Dictionary.com: <https://www.dictionary.com>

Oxford Learner's Dictionaries: <https://www.oxfordlearnersdictionaries.com>

Collins English Dictionary: <https://www.collinsdictionary.com>

YourDictionary: <https://www.yourdictionary.com>

#### BBC News:

CNN: <https://www.cnn.com>

Al Jazeera: <https://www.aljazeera.com>

Reuters: <https://www.reuters.com>

The Guardian: <https://www.theguardian.com>

NBC News: <https://www.nbcnews.com>

#### Google Maps:

Apple Maps: <https://maps.apple.com>

Bing Maps: <https://www.bing.com/maps>

MapQuest: <https://www.mapquest.com>

Waze: <https://www.waze.com>

Here WeGo: <https://wego.here.com>

#### Google Search:

Bing: <https://www.bing.com>

Yahoo Search: <https://search.yahoo.com>

DuckDuckGo: <https://duckduckgo.com>

Baidu: <https://www.baidu.com>

Yandex: <https://yandex.com>

#### Hugging Face:

OpenAI: <https://openai.com>

TensorFlow: <https://www.tensorflow.org>

PyTorch: <https://pytorch.org>

Kaggle: <https://www.kaggle.com>

SpaCy: <https://spacy.io>

#### Wolfram Alpha:

Google Scholar: <https://scholar.google.com>

Mathway: <https://www.mathway.com>

Symbolab: <https://www.symbolab.com>

Microsoft Math Solver: <https://mathsolver.microsoft.com>

Desmos: <https://www.desmos.com>

Figure 15: A list of 85 websites that we used to collect demonstration trajectories with Claude 3 Sonnet. In total 11220 trajectories were collected with different tasks. These websites were also used for testing the zeroshot generalization of PAE to out-of-distribution websites in Section ??.

List continued from Figure [15](#)

1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511

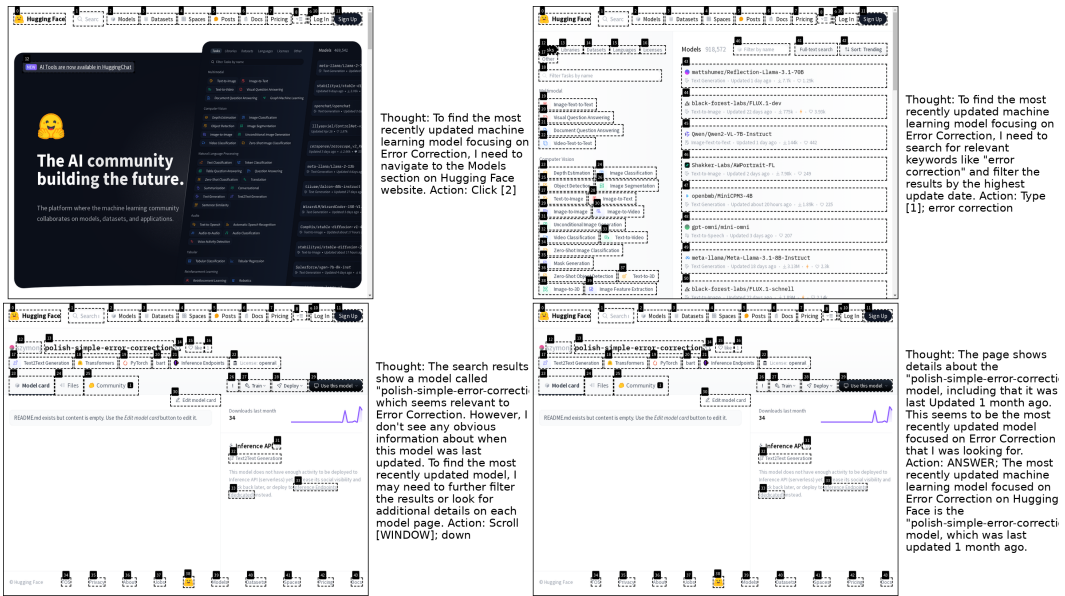
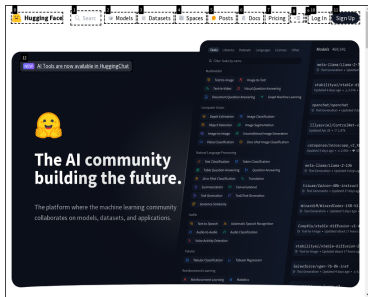
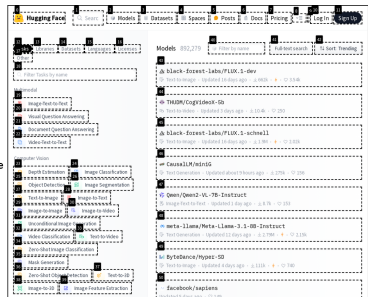


Figure 16: Full trajectories of success trajectory 1 in Figure 6 with task ‘Find the most recently updated machine learning model on Huggingface which focuses on Error Correction’ executed by model LLaVa-7B PAE.

1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565



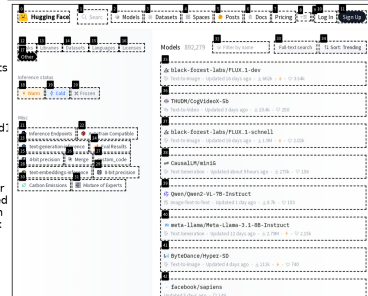
Thought: To find the most recently updated machine learning model on HuggingFace that focuses on Error Correction, I should first navigate to the Models section from the main HuggingFace page. Action: Click [2]



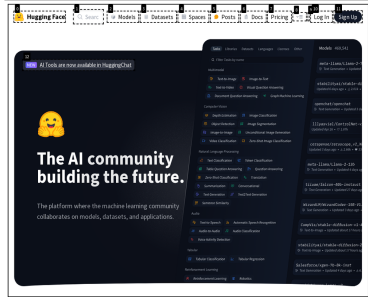
Thought: Now that I'm in the Models section, I should search for models related to Error Correction. I can try filtering the results by adding "error correction" to the search term. Action: Type [18]: error correction



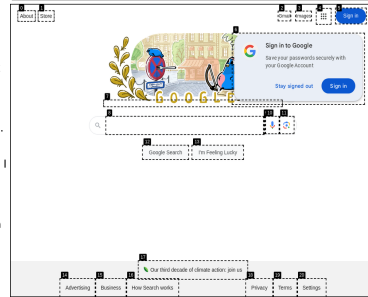
Thought: The search results show several models related to error correction, such as the "error\_correction\_llm" and "bytewise\_dance\_hypr\_s0d". However, I don't see any information about when these models were last updated. To find the most recently updated model for error correction, I may need to further refine the search or check the filters. Action: Click [17]



Thought: The "Other" filter has expanded to show options like "Safety-promptable". While these filters are useful for narrowing down the model search, I don't see any information specifically about when these models related to error correction were last updated. To find the most recent one, I may need to check the updated dates or modify my search approach. Action: GoBack



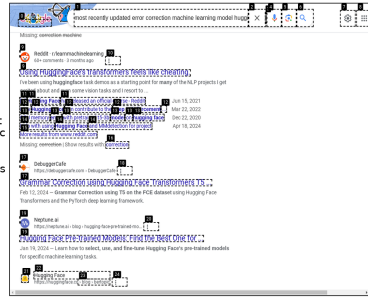
Thought: After going back to the previous search results page, I don't see any information on when the listed error correction models were last updated. The "Last Updated" filter could potentially surface the most recent ones, but I don't have that specific filter option. To find the most up-to-date model, I may need to re-evaluate my approach or search on Google. Action: Google



Thought: Since I couldn't find the most recently updated Error Correction model on HuggingFace through their search functionalities, I will try a fresh approach by searching directly on Google. Perhaps Google can surface more up-to-date information or alternative resources related to this task. Action: Type [8]: most recently updated error correction machine learning model huggingface



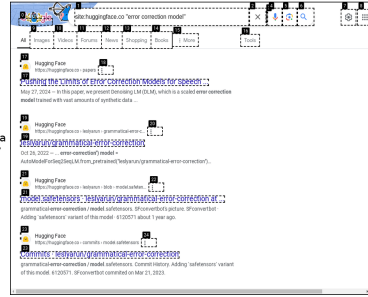
Thought: The Google search results show some potentially relevant resources from Hugging Face related to error correction models and tools. However, it's still not entirely clear which specific model is the most recently updated one focused on error correction. The results seem to cover some libraries/tools for generating prompted models but don't provide a definitive answer on the model itself. Action: Scroll [WINDOW]: down



Thought: After scrolling down further, the Google search results show some recent discussions and updates related to Hugging Face models, but still don't provide a definitive answer on which specific error correction model is the most recently updated one. The results cover some vision/image tasks but no clear information on error correction models. Action: Try again with a more specific search query on Hugging Face to hopefully find the information needed to answer the task.



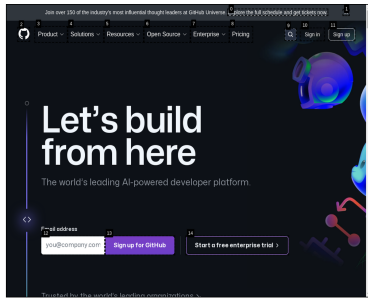
Thought: To find the most recently updated error correction model on Hugging Face, I should try a more specific search query on their website itself. Action: Type [1]: site:huggingface.co "error correction model"



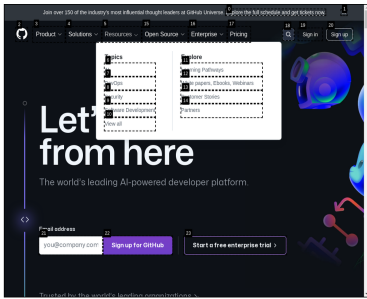
Thought: The search results on Google show some relevant error correction models from Hugging Face, but to find the most recent one, I should go directly to their website and do a focused search there. Action: Click [67]: Pushing the Limits of Error Correction Models for Speech Recognition on HF

Figure 17: Full trajectories of fail trajectory 1 in Figure 6 with task 'Find the most recently updated machine learning model on Huggingface which focuses on Error Correction' executed by model LLaVa-7B SFT.

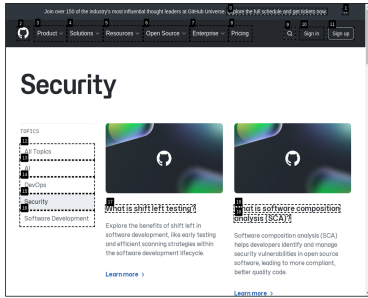
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619



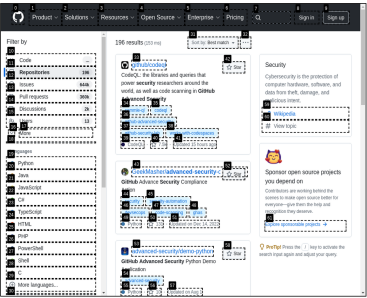
Thought: The task is to find the Security topic in GitHub Resources and answer the role of GitHub Advanced Security. To begin, I should navigate to the Resources section on the GitHub website. Action: Click [5]



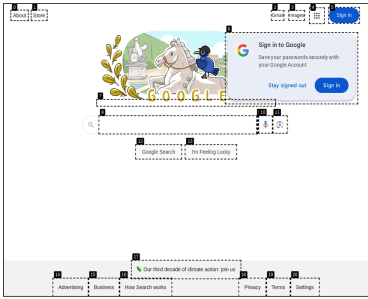
Thought: The Resources section has been expanded, revealing subsections like "AI", "DevOps", and "Security". To find information on GitHub Advanced Security, I should navigate to the "Security" subsection. Action: Click [8]



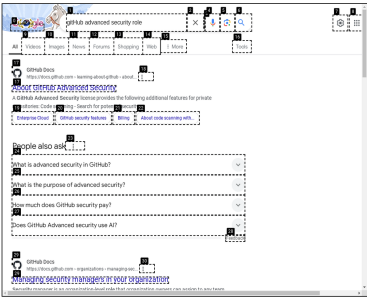
Thought: The "Security" section has two subsections - "What is shift left testing?" and "What is software composition analysis (SCA)?" These seem to provide an overview and potential benefits of shift left testing and SCA in the software development lifecycle. To learn more about GitHub Advanced Security specifically, I may need to search or navigate further within the Resources section. Action: Type [9]; github advanced security



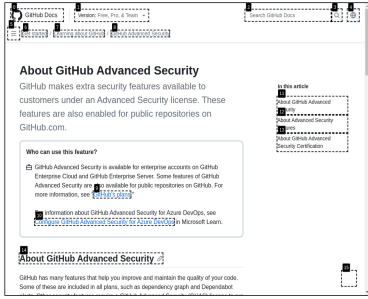
Thought: The search for "github advanced security" did not return any results. This implies that the current Resources section may not contain information specifically about GitHub Advanced Security. To find details on this topic, I may need to try a different approach like searching on Google or navigating to a different section of GitHub related to security and compliance. Action: Google



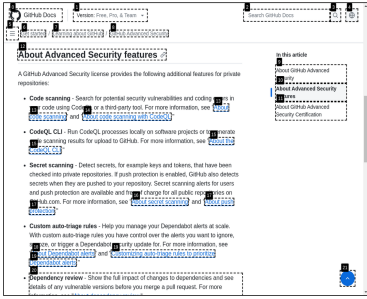
Thought: I am now on the Google search page. To find information on GitHub Advanced Security, I should enter a relevant search query in the textbox. Action: Type [8]; github advanced security role



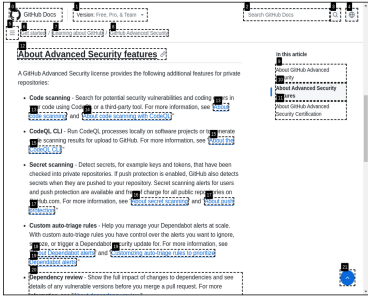
Thought: The search results provide some relevant information about GitHub Advanced Security. The top result seems to be the official documentation from GitHub on this topic. Reading through this page should give me a good understanding of the role and features included in GitHub Advanced Security. Action: Click [17]



Thought: The current page appears to be the GitHub Docs documentation on Advanced Security. It provides an overview and links to more specific information about the features included. To fully answer the role of GitHub Advanced Security, I should review the related documentation sections. Action: Click [12]



Thought: The "About Advanced Security features" section provides an overview of the vulnerability scanning, secret secrets, and secure coding features included in GitHub Advanced Security. This appears to directly address the role and capabilities of the advanced security offering. Action: Click [12]



Thought: The "About Advanced Security features" section provides an overview of the key advanced security features and capabilities, including vulnerability scanning, secured secrets detection, and secure coding rules. This directly addresses the role and capabilities of the GitHub Advanced Security offering. Action: ANSWER: GitHub Advanced Security provides enhanced security features and capabilities beyond the core GitHub platform. Key features include vulnerability scanning for open-source dependencies, secured secrets detection across projects, and secure coding rules to protect against...

Figure 18: Full trajectories of success trajectory 2 in Figure 6 with task 'Find the Security topic in GitHub Resources and answer the role of GitHub Advanced Security' executed by model LLaVa-7B PAE.



1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673

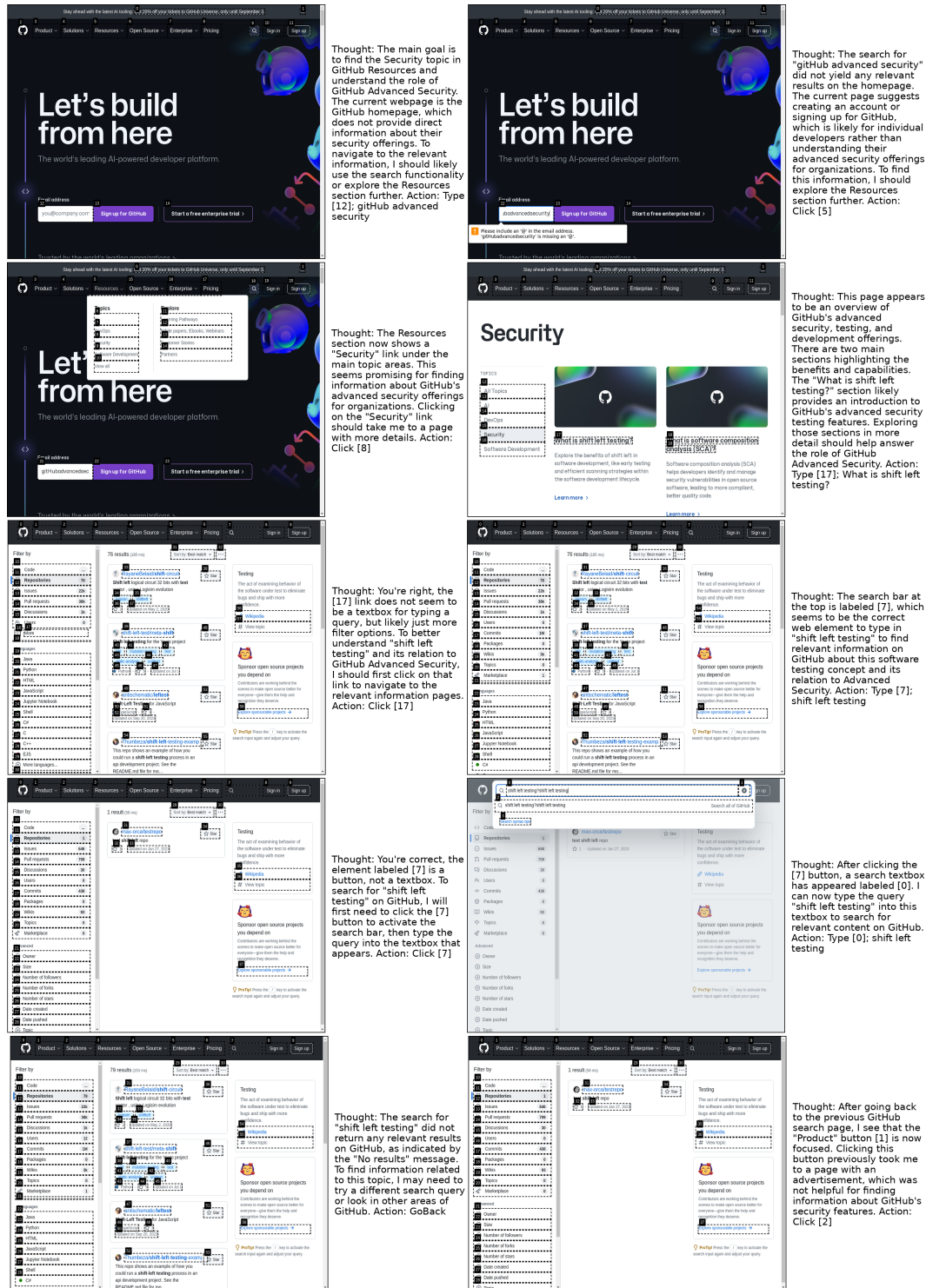


Figure 19: Full trajectories of fail trajectory 2 in Figure 6 with task 'Find the Security topic in GitHub Resources and answer the role of GitHub Advanced Security' executed by model LLaVa-7B SFT.

1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727

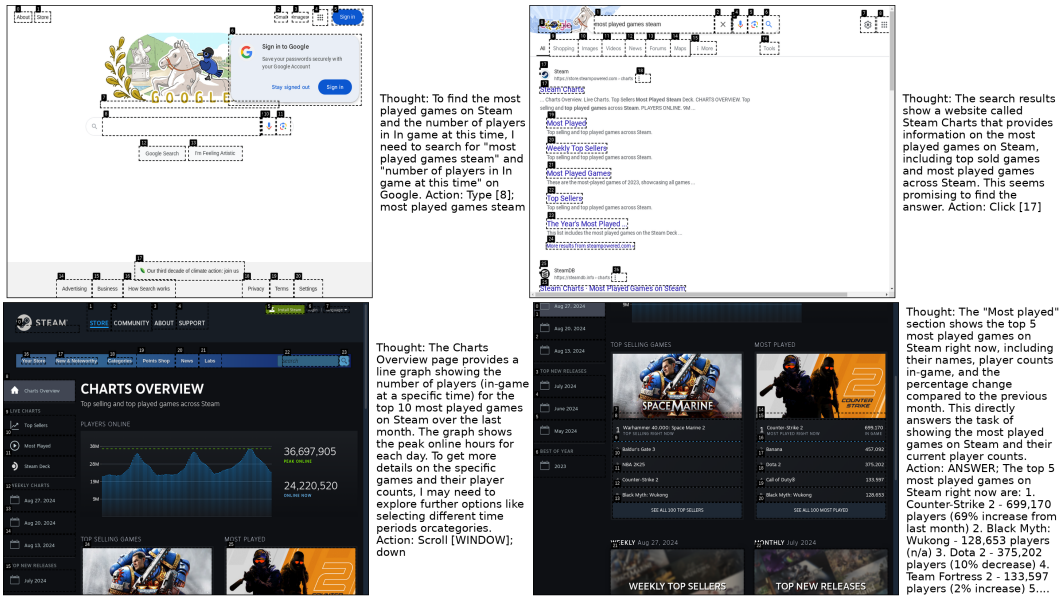


Figure 20: Extra full trajectories of successful trajectory 1 with task ‘Show most played games in Steam. And tell me the number of players in In game at this time’ executed by model LLaVa-7B PAE.

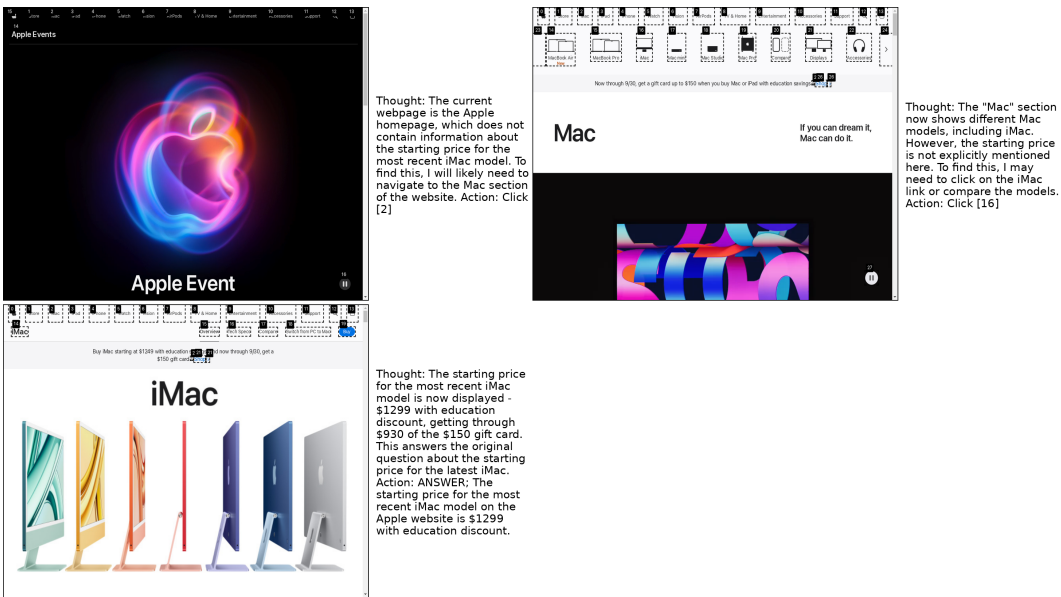


Figure 21: Extra full trajectories of successful trajectory 2 with task ‘Find out the starting price for the most recent model of the iMac on the Apple website’ executed by model LLaVa-7B PAE.



1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781

Thought: To find examples of modal verbs like "might", "could", and "may" expressing possibility, I need to search for those words in the grammar section on the Cambridge Dictionary website. Action: Type [8]; might could may

Thought: The search box has been populated with "might could may" which are the modal verbs I wanted to search for expressing possibility. I should now click the search button to get the results. Action: Click [17]

Thought: The search results show that "might", "could", and "may" expressing possibility have been entered correctly. However, it seems there are no results found for these words, which is quite surprising given their common usage and importance as modal verbs. Action: Try a new search with similar spelling or pronunciations, perhaps I missed some letters or the pronunciation might be helpful.

Thought: After multiple unsuccessful searches, I realize I may be missing some key information or steps necessary to find examples and explanations for modal verbs like "might", "could", and "may" on this Cambridge Dictionary website. Action: GoBack

Thought: With the search set to the Grammar section, I now search directly for "modal verbs" which should provide examples and explanations for the usage of modal verbs like "might", "could", "may". Action: Type [14]; modal verbs

Thought: After going back to the search page, I see there is a "Grammar" option to set the dictionary search to grammar terms. This along with searching directly for "modal verbs" or related terms like "might", "could", "may" may provide the examples and explanations I need. Action: Click [18]; Set dictionary search to Grammar

Thought: The search results for "modal verbs" provide a useful definition and examples for the usage of modal verbs like "might", "could", "may" to express possibility, willingness, obligation, necessity and ability. The first result seems to cover all the key points I need about the usage of modal verbs, including: \* They express the speaker's attitude towards what they are saying. \* Possibility, willingness, obligation, necessity and ability can be expressed with modal verbs. The example sentences shown illustrate these different uses of modal verbs. Action:...

Figure 22: Extra full trajectories of successful trajectory 3 with task ‘Look up the use of modal verbs in grammar section for expressing possibility (e.g., ‘might’, ‘could’, ‘may’) and find examples of their usage in sentences on the Cambridge Dictionary’ executed by model LLaVa-7B PAE.

1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835

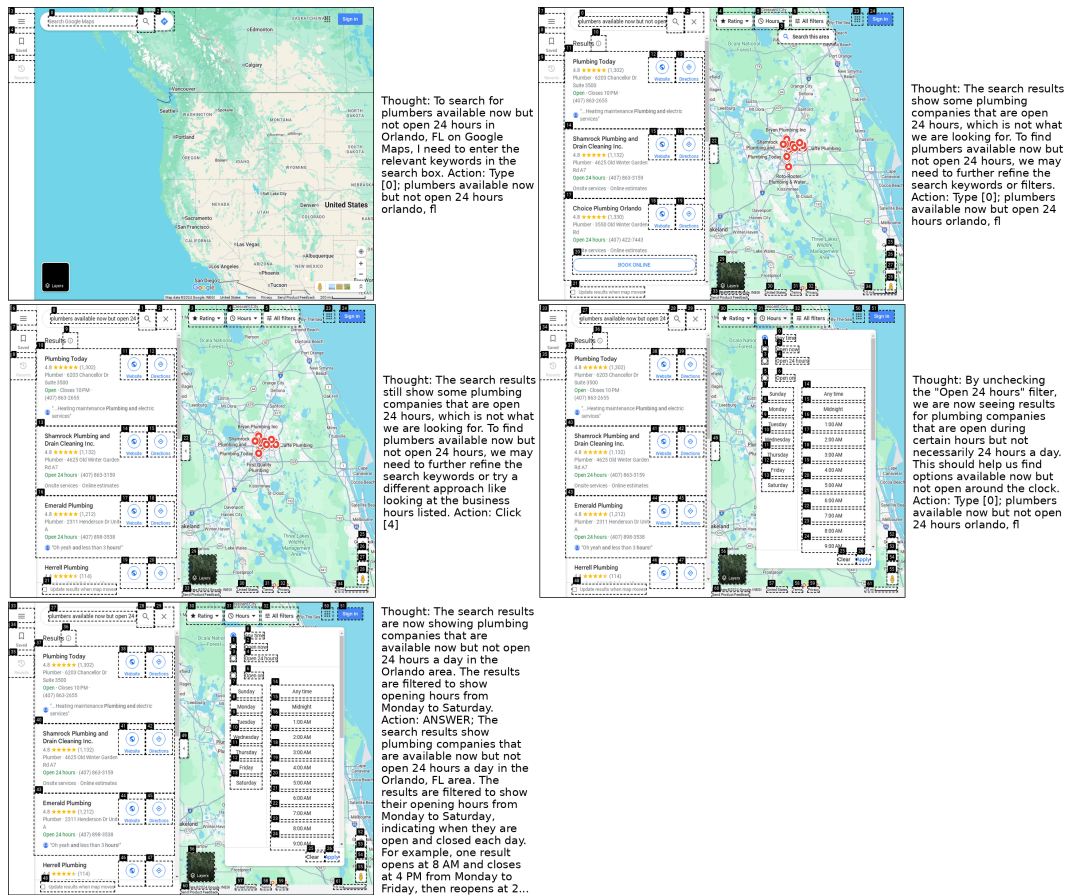
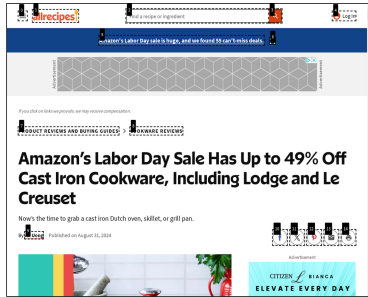
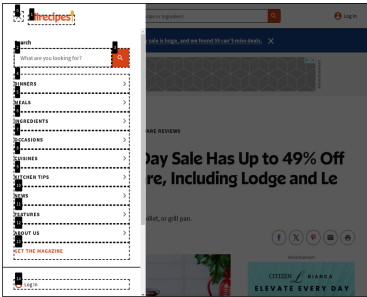


Figure 23: Extra full trajectories of successful trajectory 4 with task ‘Search for plumbers available now but not open 24 hours in Orlando, FL’ executed by model LLaVa-7B PAE.

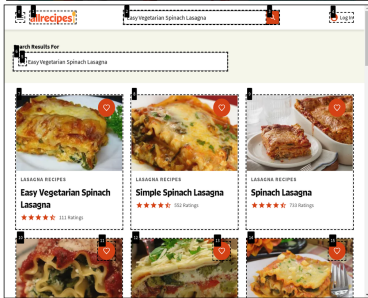
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889



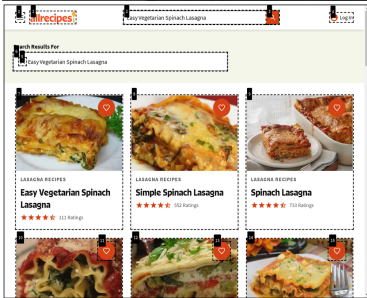
Thought: The current webpage is discussing Amazon's Labor Day sale deals on cookware, including a set that can be used for casting iron, skillets, grills, and pampers. This is not directly relevant to finding the Easy Vegetarian Spinach Lasagna recipe or getting the latest review. To proceed with the task, I should return to the main Allrecipes website. Action: Click [0]



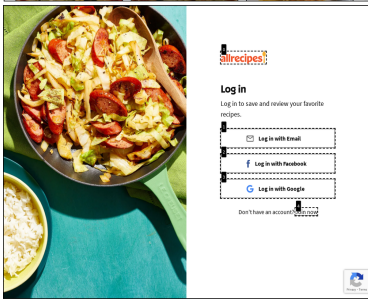
Thought: I am back on the Allrecipes homepage. To find the Easy Vegetarian Spinach Lasagna recipe and get the latest review, I should use the search functionality on this site. Action: Type [2] Easy Vegetarian Spinach Lasagna



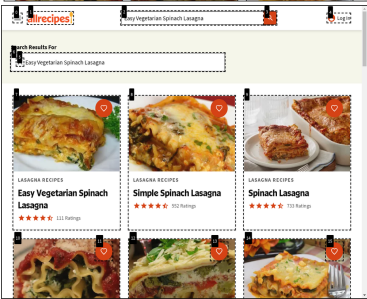
Thought: The search results for "Easy Vegetarian Spinach Lasagna" are shown. The top result seems to be the recipe I'm looking for based on the title and description. To get the latest review, I should click on this recipe link. Action: Click [606]



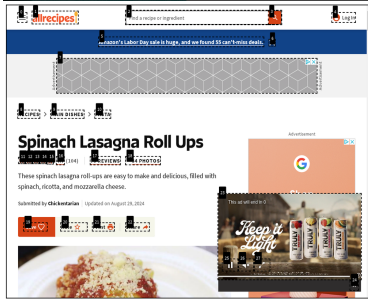
I apologize, let me take another look at the search results. Thought: Upon closer inspection, it seems the top search result is not actually a recipe from Allrecipes, but rather a search snippet from Google. To get to the full Allrecipes recipe page, I should click on the "Save Recipe" button next to that search result. Action: Click [11]



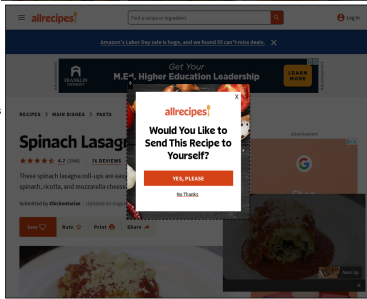
Thought: It appears I have navigated back to the login/registration page on Allrecipes, which is not necessary to find the vegetarian lasagna recipe. I should return to the search results page to continue looking for the full recipe. Action: GoBack



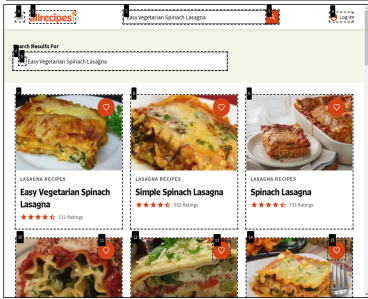
Thought: I'm back on the search results page for "easy vegetarian spinach lasagna" on Allrecipes. The top few results still seem to be just collections of lasagna recipes, not the full recipe details I'm looking for. I should try clicking on one of the lower numbered result links to hopefully find the complete Easy Vegetarian Spinach Lasagna recipe. Action: Click [10]



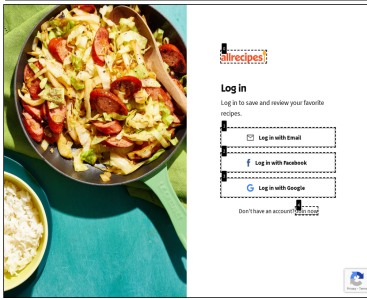
Thought: I have navigated to the full recipe details page for "Bach Lasagna Roll Ups" on Allrecipes. This seems to be a different recipe from the one I originally searched for - the Easy Vegetarian Spinach Lasagna. To find that specific recipe, I will need to go back to the search results and try clicking on a different lower numbered result link. Action: GoBack



Thought: The Allrecipes.com pop-up window is blocking the search results page I was previously on. To dismiss this and access the search results, I will need to go back from this pop-up page. Action: GoBack



Thought: The search results now show two different vegetarian lasagna recipes from Allrecipes - the Simple Spinach Lasagna and the Vegetarian Spinach Lasagna. I have the full recipe details for the Simple Spinach version from a previous attempt. To find the latest review for either recipe, I will need to click through to the individual recipe page. Action: Click [11]



Thought: I have navigated back to the Simple Spinach Lasagna recipe page on Allrecipes. The page is displaying the full recipe details again, including the ability to log in or join Allrecipes via email, Facebook, or Google. However, I don't actually need to create an account to find and review this specific recipe. Action: GoBack

Figure 24: Extra full trajectories of fail trajectory 1, with error type **Low-level Operational error**, executed by model LLaVa-7B SFT. The task is 'Find the Easy Vegetarian Spinach Lasagna recipe on Allrecipes and tell me what the latest review says'.

1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943

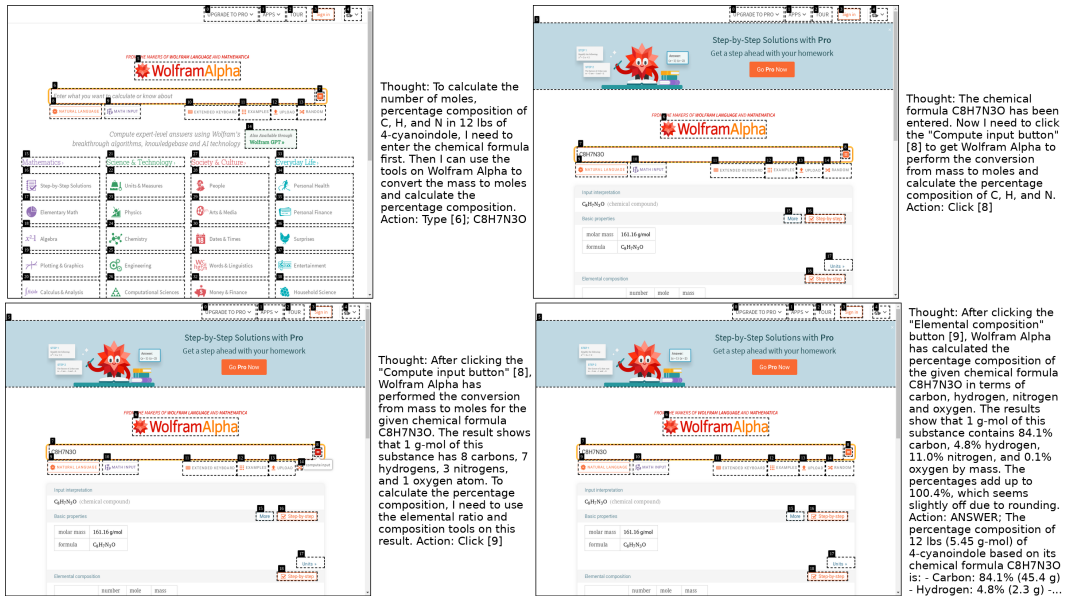


Figure 25: Extra full trajectories of fail trajectory 2, with error type **Planning or Reasoning error**, executed by model LLaVa-7B PAE. The task is 'Give 12 lbs of 4-cyanoindole, converted to molar and indicate the percentage of C, H, N'.



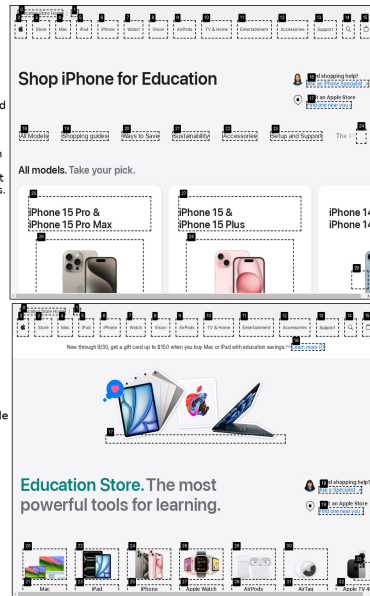
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997



Thought: The task is to find the trade-in value for an iPhone 13 Pro Max on the Apple website. Based on the screenshot, there is an iPhone section that likely contains information about trade-in values for iPhones. Action: Click [24]

Thought: The current screenshot shows the iPhone section on the Apple Store website. There is an iPhone 13 Pro Max model listed under the product filters. Clicking on that model should allow me to check the trade-in value details for it. Action: Click [26161]

Thought: The page is now showing the detailed product information for the iPhone 13 Pro Max model. I can see the pricing, options to get a gift card with education pricing, and trade-in values are listed under the "Limited-time offer" section. Action: ANSWER: The trade-in values listed for the iPhone 13 Pro Max in good condition on this Apple website are: - Trade in an eligible iPhone X or later for \$450 off the \$1,099 price. - Or trade in any iPhone model to get \$375 off the \$1,099 price. With education savings, you can get this top iPhone 13 Pro Max for just \$1,049, or \$26.12 per month for 24...



Thought: The previous iPhone section has been filtered to only show iPad models. To find the trade-in value for an iPhone 13 Pro Max, I need to navigate back to the main iPhone section. Action: GoBack

You're correct, my previous action was incorrect. Let me re-evaluate the situation. Thought: After filtering for the iPhone 13 Pro Max model, I should now see the specific product details on this page, including any trade-in offer information. I don't need to click on any additional links or elements. Action: Scroll [WINDOW]; down

Figure 26: Extra full trajectories of fail trajectory 3, with error type **Visual Hallucination**, executed by model LLaVa-7B SFT. The task is 'Find out the trade-in value for an iPhone 13 Pro Max in good condition on the Apple website'.

1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051



Figure 27: Extra full trajectories of fail trajectory 4, with error type **Timeouts**, executed by model Claude 3.5 Sonnet. The task is 'Go to the Plus section of Cambridge Dictionary, find Image quizzes and do an easy quiz about Animals and tell me your final score'.

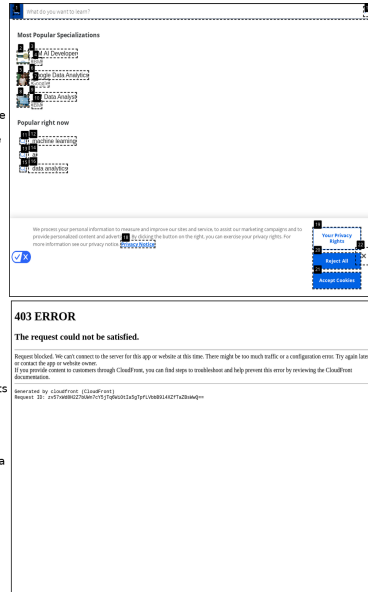
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105



Thought: The current page is the homepage of Coursera. To find a course on introduction to psychology, I need to use the search functionality. Action: Click [2]

Thought: The search results show some relevant courses on psychology, including one titled "Introduction to Human Behavioral Genetics" from the University of Minnesota which seems promising to find the requested information. Action: Click [17]

You're correct, my previous action was incorrect. Let me re-evaluate the information provided. Thought: The "CloudFront" text is likely a part of the website that cannot be interacted with directly. To find the instructor's name and approval time, I should look for more relevant details on this page. Action: ANSWER: The instructor's name for the "Introduction to Human Behavioral Genetics" course is not provided, along with the approval time. The course details indicate it requires 3 minutes of effort to complete the troubleshooting steps before reviewing it.



Thought: The search bar is now visible to search for courses related to introduction to psychology. Action: Type [1]; introduction to psychology

Thought: The course page provides details about the "Introduction to Human Behavioral Genetics" course, including that it is from the University of Minnesota and requires 3 minutes of effort to complete the troubleshooting steps before reviewing it. However, it does not directly provide the instructor's name or approval time. Action: Click [11]

Figure 28: Extra full trajectories of fail trajectory 5, with error type **Technical issues**, executed by model LLaVa-7B PAE. The task is 'Identify a course on Coursera that provides an introduction to Psychology, list the instructor's name, the institution offering it, and how many hours it will approximately take to complete'.