AutoGuide: Automated Generation and Selection of Context-Aware Guidelines for Large Language Model Agents

Yao Fu*[1 2] Dong-Ki Kim*[1] Jaekyeom Kim[1] Sungryull Sohn[1]
Lajanugen Logeswaran[1] Kyunghoon Bae[1] Honglak Lee[1 2]

Abstract

Recent advances in large language models (LLMs) have empowered AI agents to perform various sequential decision-making tasks. However, effectively guiding LLMs to perform well in unfamiliar domains like web navigation, where they lack sufficient knowledge, has proven to be difficult with the demonstration-based in-context learning paradigm. In this paper, we introduce a novel framework, called AUTOGUIDE, which addresses this limitation by automatically generating context-aware guidelines from offline experiences. As a result, our guidelines facilitate the provision of relevant knowledge for the agent's current decision-making process. Our evaluation demonstrates that AUTOGUIDE significantly outperforms competitive baselines in complex benchmark domains.

1. Introduction

Recent advances in large language models (LLMs) have empowered AI agents to address various sequential decision-making tasks and applications (Wang et al., 2023; Xi et al., 2023). The foundation of these successes involves the planning and reasoning capabilities of pre-trained LLMs, enabling agents to execute effective policies (Brohan et al., 2023; Wei et al., 2022). The predominant approach to leveraging these (typically closed source) models for sequential decision making tasks is to provide demonstrations in the form of in-context examples. However, direct application of this learning paradigm can be limited, especially in target domains where the LLM has insufficient prior knowledge such as in web navigation, where LLM agents generally achieve low success rates due to diverse and dynamic contents (Koh et al., 2024; Deng et al., 2023; Gur et al., 2023; Zhou et al., 2023). Providing all available experiences as demonstrations to an agent can further be unsuccessful due to context length limitations, prompt sensitivity, and difficulty with complex reasoning (Lu et al., 2022; Dong et al., 2022; Min et al., 2022; Kaddour et al., 2023).

On the other hand, LLMs excel in interpreting concise instructions provided as natural language, an ability that is also reinforced in the instruction-tuning phase of LLMs. Inpired by this, we explore data-driven strategies that leverage offline experiences to extract actionable knowledge to help guide LLM agents. As offline experiences implicitly convey valuable knowledge about desirable and undesirable policies in domains, they promise to serve as a useful resource for improving an LLM agent's decision-making in situations where the pre-trained LLM lacks understanding. Despite this potential benefit, a critical challenge lies in effectively extracting the implicit information embedded in offline data.

To address the challenge of extracting knowledge from offline data, we propose a novel framework, called AUTOGUIDE. Specifically, AUTOGUIDE automatically derives a comprehensive set of context-aware guidelines from offline experiences. Our method applies these context-conditional guidelines to enhance the performance of an LLM agent by retrieving guidelines relevant to the agent's current state and incorporating them into the prompt during testing (see Figure 1). Notably, we generate context-aware guidelines in concise natural language statements, effectively compressing knowledge in offline data. Moreover, context-aware guidelines clearly describe the contexts where they are applicable, so AUTOGUIDE enables an LLM agent to select pertinent guidelines for its current decision-making process. As a result, AUTOGUIDE achieves the highest success rates compared to competitive baselines in complex sequential decision-making benchmark environments.

2. AUTOGUIDE: Principled Method Based on Context-Aware Guidelines

This section details how AUTOGUIDE automatically constructs context-aware guidelines and applies them to guide

*Equal contribution. [1]LG AI Research [2]University of Michigan. Correspondence to: Yao Fu <violetfy@umich.edu>, Dong-Ki Kim <dkkim@lgresearch.ai>, Honglak Lee <honglak@eecs.umich.edu, honglak@lgresearch.ai>. Work was partially done while Yao Fu interned at LG AI Research.

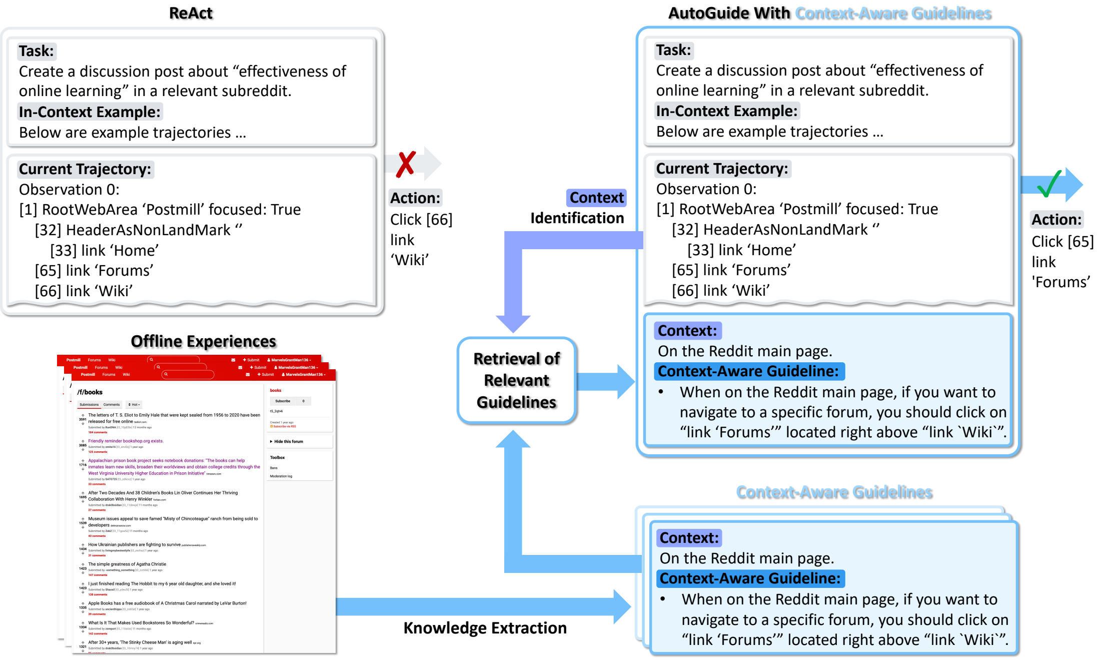

ReAct
Task:
Create a discussion post about “effectiveness of online learning” in a relevant subreddit.
In-Context Example:
Below are example trajectories ...
Current Trajectory:
Observation 0:
[1] RootWebArea ‘Postmill’ focused: True
[32] HeaderAsNonLandMark ‘’
[33] link ‘Home’
[65] link ‘Forums’
[66] link ‘Wiki’
✗
Action:
Click [66] link ‘Wiki’
AutoGuide With Context-Aware Guidelines
Task:
Create a discussion post about “effectiveness of online learning” in a relevant subreddit.
In-Context Example:
Below are example trajectories ...
Current Trajectory:
Observation 0:
[1] RootWebArea ‘Postmill’ focused: True
[32] HeaderAsNonLandMark ‘’
[33] link ‘Home’
[65] link ‘Forums’
[66] link ‘Wiki’
Context:
On the Reddit main page.
Context-Aware Guideline:
When on the Reddit main page, if you want to navigate to a specific forum, you should click on “link ‘Forums’” located right above “link `Wiki`”.
✓
Action:
Click [65] link 'Forums'
Context Identification
Retrieval of Relevant Guidelines
Offline Experiences
Knowledge Extraction
Context-Aware Guidelines
Context:
On the Reddit main page.
Context-Aware Guideline:
When on the Reddit main page, if you want to navigate to a specific forum, you should click on “link ‘Forums’” located right above “link `Wiki`”.

Figure 1. AUTOGUIDE aims to extract the implicit knowledge embedded in offline experiences and help the decision-making process of an LLM agent. Specifically, our method generates a comprehensive set of context-aware guidelines from offline data and explicitly identifies when each guideline is applicable by generating its corresponding context. Our context-aware guidelines enable providing pertinent guidelines at test time by identifying the context of the current trajectory, leading to correct decision-making compared to baselines without context-aware guidelines.

action generation.

2.1. Problem Statement

Formally, AUTOGUIDE is given offline data $\mathcal{D}_{\text{train}} = (\boldsymbol{\tau}^1, ..., \boldsymbol{\tau}^N)$ that consist of N trajectories from training tasks. Each trajectory $\boldsymbol{\tau} = (x_0, a_0, r_0, ..., r_T)$ is a sequence of observations, actions, and rewards following the partially observable Markov decision process (Sutton & Barto, 2018). The return of a trajectory is defined as the sum of rewards throughout the trajectory: $R(\tau) = \sum_{t=0}^{T} r_t$. The objective of AUTOGUIDE is to distill knowledge from offline experiences into a useful natural language format to maximize the expected return $\mathbb{E}_\tau[R(\tau)]$ during test time.

2.2. Extraction of Context-Aware Guidelines

AUTOGUIDE generates a set of context-aware guidelines by utilizing pairs of contrastive trajectories from offline data. Each context-aware guideline is expressed in concise natural language and follows a conditional structure, clearly describing the context in which the guideline is applicable. Intuitively, contrasting a pair of trajectories with different returns provides important information about when and which actions are effective or ineffective in maximizing expected returns. Building on this insight, we develop two modules for automatically extracting context-aware guidelines:

Context identification module. This module is responsible for abstracting the given partial trajectory into its CONTEXT, a concise natural language description of the agent's state. More specifically, for a timestep t and the corresponding trajectory $\boldsymbol{\tau}^i_{:t} := (x_0, a_0, ..., x_t)$, we prompt LLMs to clearly describe the agent's status:

$$\text{CONTEXT} \leftarrow \mathcal{M}_{\text{context}}(\boldsymbol{\tau}^i_{:t}), \qquad (1)$$

Our prompt templates are shown in Appendix D.1.

Guideline extraction module. This module aims to generate a desired guideline corresponding to a specific context. Let $\boldsymbol{\tau}^i_+$ and $\boldsymbol{\tau}^i_-$ represent a contrasting pair of trajectories for the same task i in offline data $\mathcal{D}_{\text{train}}$, where $R(\boldsymbol{\tau}^i_+) > R(\boldsymbol{\tau}^i_-)$. We want to contrast the pair of trajectories to find desired behaviors at an important timestep. To do this, we compare these two trajectories to find the deviation timestep t at which they begin to diverge due to different actions. Then we apply $\mathcal{M}_{\text{context}}$ to summarize the context for the shared part of the trajectory $\boldsymbol{\tau}^i_{:t}$. Eventually, we extract a useful natural language guideline by examining the paired contrastive trajectories $\boldsymbol{\tau}^i_+$ and $\boldsymbol{\tau}^i_-$ with respect to the context:

$$\text{GUIDELINE} \leftarrow \mathcal{M}_{\text{guideline}}(\boldsymbol{\tau}^i_+, \boldsymbol{\tau}^i_-, \text{CONTEXT}), \qquad (2)$$

where we refer to Appendix D.2 for our prompt template.

Construction of context-aware guidelines. We collect context-aware guidelines $\mathcal{G}$ by iterating through available

Algorithm		Offline data?	Context aware?	ALFWorld	WebShop		WebArena
				Success Rate (SR)↑	Reward↑	SR↑	SR↑
ReAct		✗	✗	54.5%	66.4	30%	8.0%
ExpeL		✓	✗	59.0%	60.9	35%	21.8%
AUTOGUIDE		✓	✓	**79.1%**	**73.4**	**46%**	**47.1%**
ReAct	+ Reflexion	✗	✗	67.2%	77.1	51%	N/A
ExpeL	+ Reflexion	✓	✗	71.6%	71.7	42%	N/A
AUTOGUIDE	+ Reflexion	✓	✓	**88.1%**	**81.4**	**57%**	N/A

Table 1. Test reward and success rate on ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022a), and WebArena (Zhou et al., 2023). The base agent model for ALFWorld and WebShop is GPT-3.5-turbo and for WebArena is GPT-4-turbo. Reflexion is done by GPT-4-turbo for at most 3 trials. In our experiments, due to token limit of GPT, we did not experiment with Reflexion on WebArena tasks.

pairs in $\mathcal{D}_{\text{train}}$ and organize the guidelines in a dictionary format, using context as the key and the corresponding guidelines as the value (see Algorithm 1). In particular, we observe that the context identification module occasionally produces contexts that describe the same situation but are expressed slightly differently. To minimize redundancy, we employ an LLM to determine if the current context corresponds to any previously identified context. If a match is found, we reuse the existing context; otherwise, we introduce a new context into $\mathcal{G}$. The specific prompt template for this context-matching procedure is in Appendix D.3.

2.3. Applying Context-Aware Guidelines at Test Time

After extracting a set of context-aware guidelines $\mathcal{G}$ from offline experiences, our method employs these guidelines to enhance the decision-making of an LLM agent during testing. At each timestep, AUTOGUIDE identifies the CONTEXT of the current test trajectory $\boldsymbol{\tau}$ up to timestep t (which represents the agent's interactions up to the current time-step) using our context identification module $\mathcal{M}_{\text{context}}$. Our guideline selection module $\mathcal{M}_{\text{select}}$ then selects relevant guidelines for CONTEXT from $\mathcal{G}$. More specifically, the module applies CONTEXT as the key to fetch a set of possible guidelines $\mathcal{G}[\text{CONTEXT}]$. If there are more than k guidelines in $\mathcal{G}[\text{CONTEXT}]$, $\mathcal{M}_{\text{select}}$ prompts an LLM to choose top-k guidelines for the specific $\boldsymbol{\tau}$:

$$\text{SELECTED} \leftarrow \mathcal{M}_{\text{select}}(\text{CONTEXT}, \boldsymbol{\tau}; \mathcal{G}, k), \tag{3}$$

where Appendix D.4 details the prompt template for this selection procedure. Subsequently, AUTOGUIDE incorporates both the context and relevant guidelines into the agent's action generation prompt. Therefore, the agent selects an action by considering the provided context and guidelines (see Figure 1 for an example). This process iterates until the end of the test trajectory (see Algorithm 2).

3. Evaluation

This section demonstrates the efficacy of AUTOGUIDE by conducting experiments on a diverse suite of sequential decision-making benchmark domains. We refer to Appendix B.2 for further experiment results and Appendix C for additional experimental details.

3.1. Baselines

We compare AUTOGUIDE against the following baselines:

- **ReAct (Yao et al., 2022b):** This method integrates reasoning and acting to address sequential decision-making tasks. However, it does not leverage offline experiences.

- **ExpeL (Zhao et al., 2023)**: This method also extracts natural language knowledge from offline data. However, it always provides all guidelines to an LLM agent while not considering applicability. ExpeL has two contributions: guideline generation and in-context example selection. Because the latter is orthogonal to our analysis, we consider ExpeL with guidelines in our experiments.

- **Reflexion (Shinn et al., 2023):** This approach converts environmental feedback into text statements to assist an LLM agent (e.g., ReAct) in the next trial. We demonstrate how AUTOGUIDE can be combined with the feedback.

3.2. Main Results

Q1. *How effective is* AUTOGUIDE *compared to baselines without context-aware guidelines?*

We compare methods on ALFWorld, WebShop, and WebArena benchmarks. The performance on the test datasets is presented in Table 1. There are two notable observations:

1. **Effectiveness of context-aware guidelines.** Our approach surpasses baseline performance on both ALFWorld and WebShop, achieving the highest success rates in Table 1. These results highlight the effectiveness of employing context-aware guidelines in language-based decision-making domains. ExpeL approach also helps ReAct by extracting knowledge from offline experiences, but its impact is not as significant. Recall that for ExpeL, the guidelines are neither generated for specific contexts at training time nor selected to only provide context-aware guidelines at test time. As a result, irrelevant guidelines can be introduced to an agent, potentially causing confusion for the agent.

Algorithm	GitHub	Flights	Coursera
	SR↑	SR↑	SR↑
SoM	2/30	5/20	1/20
AUTOGUIDE	**19/30**	**9/20**	**14/20**

Table 2. Test results of AUTOGUIDE on 3 real-world web domains within multi-modal settings. The base agent runs with GPT-4V.

Algorithm	WebShop	
	Reward↑	SR↑
ReAct (1-shot)	66.4	30%
ReAct (2-shot)	66.0	35%
ReAct (4-shot)	70.2	37%
ReAct (6-shot)	71.0	38%
AUTOGUIDE	**73.4**	**46%**

Table 3. AUTOGUIDE against ReAct with varying numbers of in-context examples.

Algorithm	WebArena–Shopping
	SR↑
ReAct	10.2%
AUTOGUIDE	20.4%

Table 4. Out-of-distribution generalization of WebShop guidelines on the 98 WebArena–Shopping tasks with a product in the intent template.

2. **Scalability to complex environments.** We conduct experiments on WebArena-Reddit, which features diverse tasks on realistic and complex websites. We observe that ReAct scores low (8.0%) due to the complex observation and action spaces and long task horizon. For ExpeL, the issue of presenting all guidelines to an agent is exacerbated compared to simpler environments like ALFWorld and WebShop. WebArena's wide variety of tasks require a larger number of guidelines to cover the knowledge needed. This results in either an overload of irrelevant guidelines that could mislead the agent or a lack of crucial information when the number of guidelines is limited, as suggested in ExpeL (Zhao et al., 2023). In contrast, AUTOGUIDE achieves a more significant performance enhancement (47.1%) compared to ExpeL (21.8%) by efficiently providing pertinent guidelines and minimizing the burden on context capacity.

Q2. *How does* AUTOGUIDE *perform when combined with test-time self-feedback approaches?*

Our context-aware guidelines effectively provide *inter-task* knowledge by considering multiple training tasks. Meanwhile, self-feedback methods (e.g., Reflexion) offer *intra-task* knowledge based on test-time environmental feedback. In this question, we explore the effectiveness of integrating both inter-task and intra-task information. The results in Table 1 demonstrate that the combination of AUTOGUIDE with Reflexion achieves the highest performance on WebShop and ALFWorld, showing that our context-aware guidelines positively complement the intra-task knowledge of Reflexion. Another observation is that, while ExpeL + Reflexion outperforms ExpeL alone, this combination is not as effective as other approaches. This limitation may stem from ExpeL introducing irrelevant knowledge, potentially leading to conflicts with Reflexion's feedback and having an adverse impact.

Q3. *Can* AUTOGUIDE *generate context-aware guidelines for multi-modal inputs?*

Going beyond text-only inputs is an essential step toward building capable agents for solving real-world environments and tasks. We test AUTOGUIDE in a complex multi-modal setting with image and text observations. Specifically, we introduce a set of real-world website navigation tasks in 3 domains: GitHub, Google Flights, and Coursera. For these multi-modal tasks, we employ the Set-of-Marks (SoM) agent (Yang et al., 2023; Koh et al., 2024) as our base method. We apply AUTOGUIDE with GPT-4V to generate natural language context-aware guidelines from collected trajectories with image and text observations. Table 2 shows the effectiveness of AUTOGUIDE, demonstrating its generalization ability to complex real-world multi-modal settings.

3.3. Analyses of AUTOGUIDE

Q4. *How does* AUTOGUIDE *compare to ReAct with varying numbers of in-context examples?*

Table 3 shows that, while increasing the number of in-context examples for ReAct gradually improves performance, there is a plateau at a certain number of shots. Additionally, ReAct with more than 6 shots often exceeds the token limit of GPT-3.5-turbo. Therefore, directly inputting raw trajectories into ReAct for in-context learning is not an effective way to fully leverage offline data. In contrast, AUTOGUIDE extracts knowledge from offline data by summarizing them into concise context-aware guidelines, making them easy to integrate with prompt-based agents.

Q5. *How do* AUTOGUIDE*'s context-aware guidelines generalize to out-of-domain environments?*

We conduct an experiment to further demonstrate AUTOGUIDE's out-of-domain capability across different domains but relevant tasks. We extract context-aware guidelines from WebShop and apply them to WebArena-Shopping, which is a distinct domain with variations in observation/action spaces, task intentions, and episodic horizons. For this domain adaptation case, we additionally incorporate a grounding module to align the context-aware guidelines from WebShop to WebArena's observations based on GPT-4-Turbo. As shown in Table 4, the transferred guidelines bring a notable improvement compared to the ReAct baseline.

4. Conclusion

We present AUTOGUIDE, an effective framework for exploiting important domain knowledge from offline experiences for improving decision-making with pre-trained LLMs. We proposed to generate context-aware guidelines that can be incorporated into prompts for LLM agents. Empirically, AUTOGUIDE outperforms strong baselines by a large margin and achieves outstanding performance in decision-making benchmarks.

References

Google Flights. https://www.google.com/travel/flights. Accessed: 2024-05-21.

Branavan, S., Silver, D., and Barzilay, R. Learning to win by reading manuals in a monte-carlo framework. *Journal of Artificial Intelligence Research*, 43:661–704, 2012.

Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pp. 287–318. PMLR, 2023.

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. *arXiv preprint arXiv:2306.06070*, 2023.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., and Neubig, G. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.

Gur, I., Nachum, O., Miao, Y., Safdari, M., Huang, A., Chowdhery, A., Narang, S., Fiedel, N., and Faust, A. Understanding HTML with large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2803–2821, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.185. URL https://aclanthology.org/2023.findings-emnlp.185.

Hanjie, A. W., Zhong, V. Y., and Narasimhan, K. Grounding language to entities and dynamics for generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 4051–4062. PMLR, 2021.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pp. 9118–9147. PMLR, 2022.

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., and McHardy, R. Challenges and applications of large language models, 2023.

Kim, G., Baldi, P., and McAleer, S. Language models can solve computer tasks. *arXiv preprint arXiv:2303.17491*, 2023.

Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M. C., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., and Fried, D. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

Logeswaran, L., Fu, Y., Lee, M., and Lee, H. Few-shot subgoal planning with language models. In *NAACL: HLT*, 2022.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *ACL*, 2022.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*, 2022.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2024.

Parisi, A., Zhao, Y., and Fiedel, N. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*, 2022.

Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K. R., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *ICLR*, 2021.

Sun, H., Zhuang, Y., Kong, L., Dai, B., and Zhang, C. Adaplanner: Adaptive planning from feedback with language models. *arXiv preprint arXiv:2305.16653*, 2023.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

Yang, J., Zhang, H., Li, F., Zou, X., Li, C., and Gao, J. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Zeng, A., Liu, M., Lu, R., Wang, B., Liu, X., Dong, Y., and Tang, J. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*, 2023.

Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y.-J., and Huang, G. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*, 2023.

Zheng, B., Gou, B., Kil, J., Sun, H., and Su, Y. Gpt-4v(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

Zhong, V., Rocktäschel, T., and Grefenstette, E. Rtfm: Generalising to new environment dynamics via reading. In *ICLR*, pp. 1–17. ICLR, 2020.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Algorithm 1 Extracting context-aware guidelines from offline data

```
Input: Offline data $\mathcal{D}_{\text{train}}$, context identification module $\mathcal{M}_{\text{context}}$, guideline extraction module $\mathcal{M}_{\text{guideline}}$
Initialize context-aware guideline dictionary $\mathcal{G}$
for Each pair $\boldsymbol{\tau}^i_+, \boldsymbol{\tau}^i_- \in \mathcal{D}_{\text{train}}$ do
    # Identify the context from a trajectory
    Find the deviating timestep $t$ from $\boldsymbol{\tau}^i_+$ and $\boldsymbol{\tau}^i_-$
    $\textsc{context} \leftarrow \mathcal{M}_{\text{context}}(\boldsymbol{\tau}^i_{:t})$
    # Check if the current context matches any
        existing contexts
    if $\textsc{context} \notin \mathcal{G}$ then
        $\mathcal{G}[\textsc{context}] = \{\}$
    end if
    # Generate the context-aware guideline
    $\textsc{guideline} \leftarrow \mathcal{M}_{\text{guideline}}(\boldsymbol{\tau}^i_+, \boldsymbol{\tau}^i_-, \textsc{context})$
    $\mathcal{G}[\textsc{context}] \leftarrow \mathcal{G}[\textsc{context}] \cup \{\textsc{guideline}\}$
end for
Return Context-aware guideline dictionary $\mathcal{G}$
```

Algorithm 2 Applying context-aware guidelines at test time

```
Input: Context-aware guideline dictionary $\mathcal{G}$, context identification module $\mathcal{M}_{\text{context}}$, guideline selection module $\mathcal{M}_{\text{select}}$, LLM agent policy $\pi$
Initialize test trajectory $\boldsymbol{\tau} = \{x_0\}$
for Each timestep $t$ do
    # Identify the current context from a trajectory
    $\textsc{context} \leftarrow \mathcal{M}_{\text{context}}(\boldsymbol{\tau})$
    # If the current context matches any existing
        ones, perform top-$k$ guideline selection
    if $\textsc{context} \in \mathcal{G}$ then
        $\textsc{guidelines} \leftarrow \mathcal{M}_{\text{select}}(\textsc{context}, \boldsymbol{\tau}; \mathcal{G}, k)$
    else
        $\textsc{guidelines} \leftarrow \varnothing$
    end if
    # Action selection based on guidelines
    $a_t \sim \pi(\boldsymbol{\tau}, \textsc{context}, \textsc{guidelines})$
    Execute action $a_t$ and observe $x_{t+1}$
    Update trajectory $\boldsymbol{\tau} \leftarrow \boldsymbol{\tau} \cup \{\textsc{context}, a_t, x_{t+1}\}$
end for
```

A. Algorithms

We refer to Algorithm 1 and Algorithm 2 for more details of AutoGuide.

B. Additional Evaluation Details

B.1. Sequential Decision-Making Benchmark Domains

We consider the following interactive sequential decision-making benchmarks to study various aspects of AutoGuide (see Figure 2):

- **ALFWorld (Shridhar et al., 2021):** In this embodied benchmark, an LLM agent interacts with an environment to carry out household tasks, such as placing a pan on the dining table. Observations and actions are expressed in natural language statements, and the agent must navigate through the space and manipulate objects to successfully complete the tasks.

- **WebShop (Yao et al., 2022a):** This interactive web environment simulates the task of online shopping on an e-commerce website. The agent's goal is to understand a text instruction and buy a product that meets specified criteria. This involves querying the website's search engine, understanding the descriptions and details of each item, and selecting necessary options.

- **WebArena (Zhou et al., 2023):** This web-based benchmark introduces realistic environments by replicating the functionality and data found in popular web domains (e.g., Gitlab, Reddit, Wikipedia). Compared to WebShop, WebArena presents more challenges and difficulties for an LLM agent due to its large observation and action space, along with tasks that involve longer planning horizons. We focus oe the Reddit domain for the main WebArena experiments.

- **Real-world multi-modal websites:** Finally, we consider evaluating AutoGuide on a variety of real-world website tasks. These span from a collaborative software development platform (e.g., GitHub) to a flight search engine (e.g., Google Flights) and an online education platform (e.g., Coursera). Please refer to Appendix C.4 for example tasks. In particular, in comparison to WebShop and WebArena, we design our tasks to be multi-modal such that the agent must consider both visual (e.g., images) and textual information (e.g., HTML) to complete these tasks.

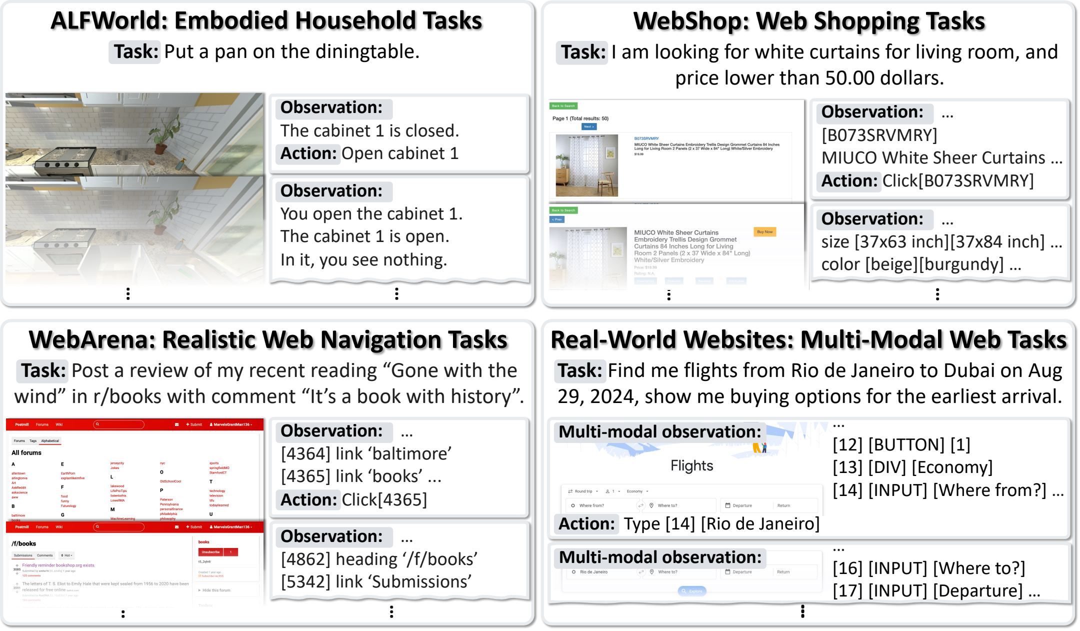
B.2. Additional Results

Q. *Qualitative Comparison between* AutoGuide *and baselines.*

To further examine the action selection differences among ReAct, ExpeL, and our method, we present their trajectories in Figure 3. We observed that ReAct makes common mistakes such as trying to take soapbar that is not visible, or taking a soapbottle instead of soapbar due to their similar names. Both ExpeL and AutoGuide improve on this by extracting guidelines from similar mistakes in the offline experience. However, ExpeL often erroneously applies incorrect guidelines due to the availability of all guidelines at each timestep. In Figure 3, ExpeL mistakenly attends to the second guideline *"ensure to specify the item's number and location ..."*, leading to wrong reasoning and action. AutoGuide presents relevant guidelines at necessary moments, enabling accurate task completion by avoiding the mistakes seen in ExpeL and ReAct.

Q. *How does altering the number of top-k guidelines impact the performance of* AutoGuide*?*

We conducted an ablation study on WebShop using various values of k in Table 5. We find that employing context-



ALFWorld: Embodied Household Tasks

Task: Put a pan on the diningtable.

Observation: The cabinet 1 is closed.
Action: Open cabinet 1

Observation: You open the cabinet 1. The cabinet 1 is open. In it, you see nothing.

⋮

WebShop: Web Shopping Tasks

Task: I am looking for white curtains for living room, and price lower than 50.00 dollars.

Observation: … [B073SRVMRY] MIUCO White Sheer Curtains …
Action: Click[B073SRVMRY]

Observation: … size [37x63 inch][37x84 inch] … color [beige][burgundy] …

⋮

WebArena: Realistic Web Navigation Tasks

Task: Post a review of my recent reading “Gone with the wind” in r/books with comment “It’s a book with history”.

Observation: … [4364] link ‘baltimore’ [4365] link ‘books’ …
Action: Click[4365]

Observation: … [4862] heading ‘/f/books’ [5342] link ‘Submissions’

⋮

Real-World Websites: Multi-Modal Web Tasks

Task: Find me flights from Rio de Janeiro to Dubai on Aug 29, 2024, show me buying options for the earliest arrival.

Multi-modal observation: … [12] [BUTTON] [1] [13] [DIV] [Economy] [14] [INPUT] [Where from?] …
Action: Type [14] [Rio de Janeiro]

Multi-modal observation: … [16] [INPUT] [Where to?] [17] [INPUT] [Departure] …

⋮

Figure 2. Sequential decision-making benchmark domains considered in our work: ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022a), WebArena (Zhou et al., 2023), and multi-modal real-world websites. Graphic credit: (Shridhar et al., 2020; Yao et al., 2022a; Zhou et al., 2023; gtr).

aware guidelines consistently outperforms the no-guideline baseline ($k=0$; ReAct). The $k = 3$ yields the best performance. The largest k value of 5 can lead an LLM agent to overthink, potentially resulting in a slight decrease in performance. Conversely, a smaller k, like $k=1$, may cause LLM to overlook additional helpful guidelines, leading to slightly worse performance.

Top-k	WebShop
	SR↑
$k=0$	30%
$k=1$	42%
$k=2$	46%
$k=3$	47%
$k=5$	43%

Table 5. Ablation study of AUTOGUIDE with various top-k values.

Q. *How does each component of* AUTOGUIDE *contribute to the final results?*

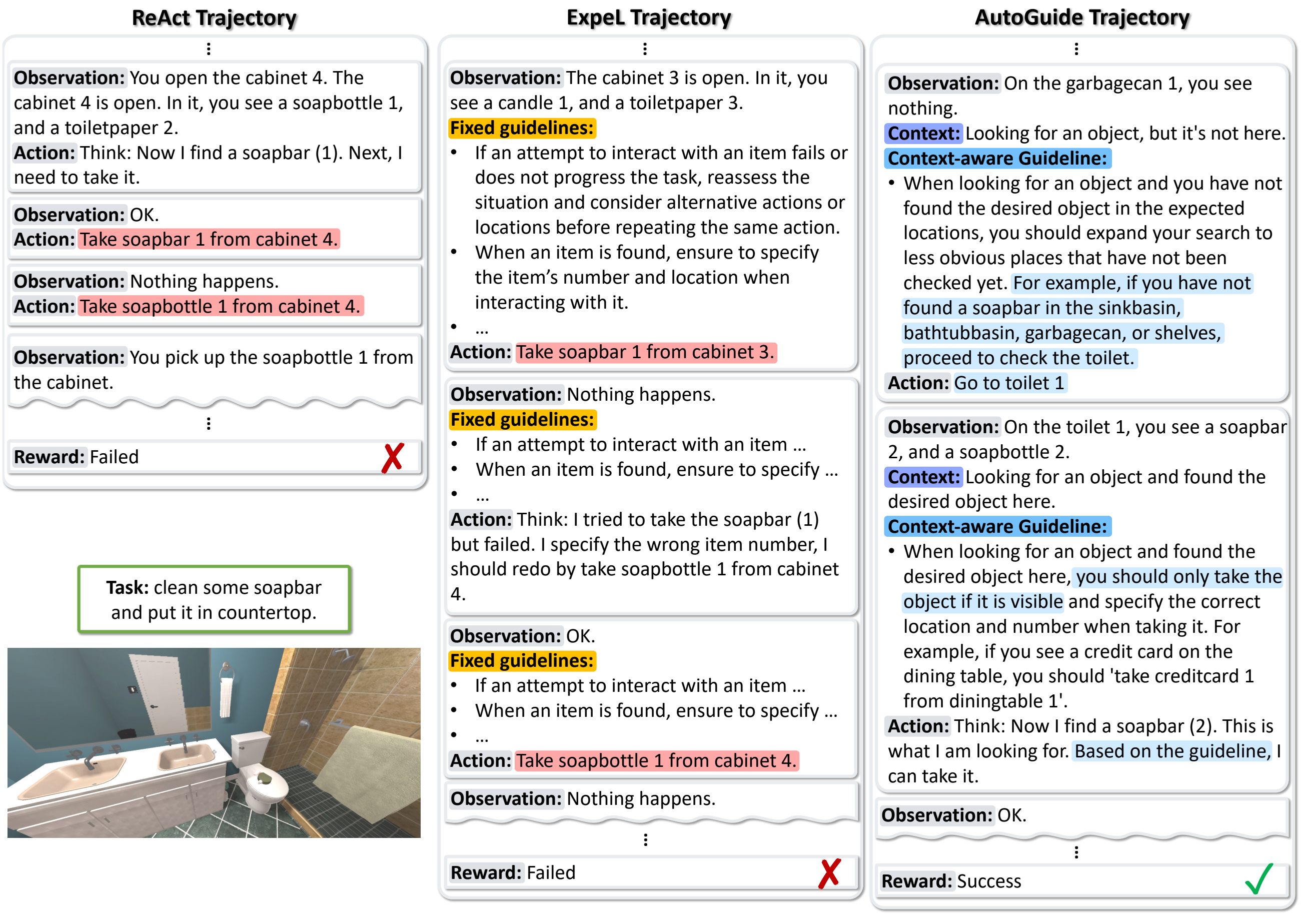
We evaluate the impact of different components within AUTOGUIDE on its performance in WebShop, as detailed in Table 6. We examine two variants: ReAct+CI and ReAct+GES. The ReAct+CI, which incorporates contexts into observations without guidelines, shows improvement over ReAct. This suggests that contexts enhance decision-making by verifying the current state before action selection. ReAct+GES, which generates guidelines from trajectories without contexts and employs GPT-3.5-turbo for guideline selection, also enhances performance but is less effective than the full AUTOGUIDE. This indicates that choosing relevant guidelines based on the trajectory alone is more challenging than using contexts. Therefore, integrating both context summaries and guidelines is crucial for maximizing the benefits of AUTOGUIDE.

Algorithm	CI	GES	WebShop SR↑
ReAct	✗	✗	30%
ReAct + CI	✓	✗	36%
ReAct + GES	✗	✓	37%
AUTOGUIDE	✓	✓	**46%**

Table 6. Ablation study of AUTOGUIDE, analyzing each module’s contribution in WebShop. CI denotes our context identification module, and GES denotes the guideline extraction and selection modules.

C. Implementation Details

We collect offline experiences either by running ReAct and Reflexion, or incorporating human demonstrations. We use ReAct with GPT-3.5-turbo as our base LLM agent for WebShop and ALFWorld, GPT-4-turbo for WebArena, and GPT-4V for real-world website domains. For each benchmark, we apply the same GPT model for action generation, context



ReAct Trajectory

⋮

Observation: You open the cabinet 4. The cabinet 4 is open. In it, you see a soapbottle 1, and a toiletpaper 2.
Action: Think: Now I find a soapbar (1). Next, I need to take it.

Observation: OK.
Action: Take soapbar 1 from cabinet 4.

Observation: Nothing happens.
Action: Take soapbottle 1 from cabinet 4.

Observation: You pick up the soapbottle 1 from the cabinet.

⋮

Reward: Failed ✗

Task: clean some soapbar and put it in countertop.

ExpeL Trajectory

⋮

Observation: The cabinet 3 is open. In it, you see a candle 1, and a toiletpaper 3.
Fixed guidelines:
- If an attempt to interact with an item fails or does not progress the task, reassess the situation and consider alternative actions or locations before repeating the same action.
- When an item is found, ensure to specify the item's number and location when interacting with it.
- …

Action: Take soapbar 1 from cabinet 3.

Observation: Nothing happens.
Fixed guidelines:
- If an attempt to interact with an item …
- When an item is found, ensure to specify …
- …

Action: Think: I tried to take the soapbar (1) but failed. I specify the wrong item number, I should redo by take soapbottle 1 from cabinet 4.

Observation: OK.
Fixed guidelines:
- If an attempt to interact with an item …
- When an item is found, ensure to specify …
- …

Action: Take soapbottle 1 from cabinet 4.

Observation: Nothing happens.

⋮

Reward: Failed ✗

AutoGuide Trajectory

⋮

Observation: On the garbagecan 1, you see nothing.
Context: Looking for an object, but it's not here.
Context-aware Guideline:
- When looking for an object and you have not found the desired object in the expected locations, you should expand your search to less obvious places that have not been checked yet. For example, if you have not found a soapbar in the sinkbasin, bathtubbasin, garbagecan, or shelves, proceed to check the toilet.

Action: Go to toilet 1

Observation: On the toilet 1, you see a soapbar 2, and a soapbottle 2.
Context: Looking for an object and found the desired object here.
Context-aware Guideline:
- When looking for an object and found the desired object here, you should only take the object if it is visible and specify the correct location and number when taking it. For example, if you see a credit card on the dining table, you should 'take creditcard 1 from diningtable 1'.

Action: Think: Now I find a soapbar (2). This is what I am looking for. Based on the guideline, I can take it.

Observation: OK.

⋮

Reward: Success ✓

Figure 3. Trajectories of ReAct, ExpeL, and AutoGuide from the same test task. ReAct (Left) chose the wrong item, consequently failing the task in the end. ExpeL (middle) was confused by guidelines that were irrelevant to current context, leading to incorrect reasoning and actions. AutoGuide (right) selects relevant guidelines to the agent's context, enabling the agent to accomplish the task.

identification, and guideline selection. We extract context-aware guidelines from offline data with GPT-4-turbo (OpenAI et al., 2024) or GPT-4V and evaluate their effectiveness by applying them to the test set with non-overlapping tasks. Here are more details for each benchmark:

C.1. ALFWorld (Shridhar et al., 2021)

C.1.1. Environment details

Each task in ALFWorld starts with a description of the specific environment and the goal to achieve. At each timestep, an agent can choose one of the following actions to interact with the objects (e.g., apple, pan), and receptacles (e.g., microwave, cabinet) in the environment:

- go to [recep]
- take [object] from [recep]
- put [object] in/on [recep]
- open/close/use [recep]
- clean/heat/cool [object] with [recep]

Alternatively, following the setting in ReAct, the agent can generate think actions for planning and reflection, which help with decision-making but do not change the environment itself. After one action is performed, the environment returns an observation that describes view changes, whether the action is successfully performed or not, or only a *OK* for a think action.

Following ReAct, we concatenate a list of (observation, action) pairs to show the entire trajectory up to the current timestep for LLM agents to generate the next action. We experiment on 134 unseen test tasks with 6 categories of `pick_and_place`, `pick_clean_then_place`, `pick_heat_then_place`, `pick_cool_then_place`, `look_at_obj`, and `pick_two_obj`. For each task, the agent is allowed to take a maximum of 50 actions.

C.1.2. Baseline and Models

For ALFWorld tasks, we follow the same setting as ReAct by providing 2 in-context examples for each of the 6 task categories. The original results of ReAct in their paper are produced based on Text-Davinci-002. However, this GPT version is no longer available, so we apply gpt-3.5-turbo-instruct instead to generate actions. For ExpeL, we directly take the guidelines from their appendix and append them to the ReAct agent at test time.

C.1.3. Implementation details ofAutoGuide

We run the first 100 training tasks of ALFWorld to collect (τ_+, τ_-) pairs with ReAct+Reflexion and extract context-dependant guidelines on the collected data. For context identification, we provide 2-shot demonstrations for each of the 6 task categories. The corresponding prompt templates can be found in appendix D. All parameter details are shown in Table 7.

Parameter name	Value
Allowed Episode Length	50
n-shots	2
Agent Model	gpt-3.5-turbo-instruct
Context Identification Model	gpt-3.5-turbo-instruct
Guideline Selection Model	gpt-3.5-turbo-instruct
Guideline Extraction Model	gpt-4-1106-preview
Reflexion Model	gpt-4-1106-preview
top-k guideline selection	2

Table 7. Experiment hyperparameters on ALFWorld. The maximum allowed episode length and n-shots follow the same setup in ReAct.

C.2. WebShop (Yao et al., 2022a)

C.2.1. Environment details

WebShop provides an e-commerce environment, where the objective is to find and buy the product that matches the task-specific Instruction. The agent can select one of the following actions to perform:

- search[query]
- click[button]

Following ReAct, the agent can generate think actions to do planning or reflection. After buying a product, the environment returns a reward showing how well the bought product matches the target one in type, price, buying options, and attributes. The reward is calculated by:

$$r = r_{type} \cdot \frac{|U_{att} \cap Y_{att}| + |U_{opt} \cap Y_{opt}| + \mathbb{1}[y_{price} \leq u_{price}]}{|U_{att}| + |U_{opt}| + 1}$$

where y is the bought product and u is the target product. Same as ALFWorld, for WebShop, the agent takes $(\mathrm{obs}_t, \mathrm{act}_t)$ pairs for every previous timestep t as input to generate the next action.

C.2.2. Baseline and Models

Following ReAct, experiments are done in a one-shot setting. We apply gpt-3.5-turbo-0613 to generate actions, but when the token number exceeds the token limit (for example, for the n-shot ReAct experiments in Table 1), we use the 16k version of gpt-3.5-turbo-0613 instead. For ExpeL, we could not find how many training tasks the framework used for training. Therefore, we directly apply the guidelines from the appendix of their paper at test time. We only consider ExpeL with guidelines, not ExpeL with in-context example selection in our experiments for a fair comparison. The in-context example selection method is orthogonal to our work and can be easily combined with our method. For Reflexion, as shown in their paper, their 2-shot Reflexion prompt does not work well on WebShop. Therefore, we re-write the prompt and apply gpt-4-1106-preview to generate episode-level reflections for all Reflexion experiments. Following Reflexion and ExpeL, the evaluation is done on 100 test tasks. The maximum number of allowed actions for each task is 15. At the same time, each search action shows the top 3 products for the search query. Please refer to Table 8 for more details about the experiments.

C.2.3. Implementation details of AutoGuide

We randomly sample 50 training tasks from the training set of WebShop, on which we run ReAct+Reflexion to collect pairs and generate guidelines. The context identification prompt is one-shot, which is shown in Appendix D. At test time, we ask gpt-3.5-turbo-0613 to select each state's most relevant top 2 guidelines.

Parameter name	Value
Allowed Episode Length	15
# of Search Results	3
n-shots	1
Agent Model	gpt-3.5-turbo-0613
Context Identification Model	gpt-3.5-turbo-0613
Guideline Selection Model	gpt-3.5-turbo-0613
Guideline Extraction Model	gpt-4-1106-preview
Reflexion Model	gpt-4-1106-preview
top-k guideline selection	2

Table 8. Experiment hyperparameters on WebShop. The maximum allowed episode length, the number of search results per page, and n-shots follow the same setup in ReAct.

C.3. WebArena (Zhou et al., 2023)

C.3.1. Environment details

WebArena provides web-based benchmark environments that closely follow the data and functionality of real-world websites. Unlike other benchmarks like WebShop that provide clean text of website information as observation, WebArena's webpage content is represented as an accessibility tree, which is a subset of the DOM tree with useful elements of a webpage. For our expeirments, we focus on WebArena Reddit, which simulates the real Reddit websites with users, forums, and posts with abundant text information.

For each task in WebArena, the agent is expected to achieve a task-specific intent. At each timestep, WebArena provides a list of opened tabs, the accessibility tree of the focused webpage, and the URL of the current page as observation. For WebArena, each observation is long. Therefore, following the baseline in WebArena, at each timestep, we only provide the observation of the current timestep to the agent. We additionally provide up to 5 past actions for the agent to understand what it did in the past. The allowed actions in WebArena include the following:

- goto [url]
- click [element_id]
- type [element_id] [text] [1 for enter or 0 for not enter]
- press [key_combination]
- scroll [up or down]
- go_back

The maximum allowed number of actions for a single task is 20. Note that WebArena does not provide training tasks, but the work provides 19 demonstrations for Reddit, each of a different category. Therefore, we set these 19 tasks as the training tasks and then test on the rest 87 tasks.

C.3.2. Baseline and Models

We directly run the two-shot ReAct-style baseline in the official codebase of WebArena using gpt4-preview-1106. For ExpeL, the original paper does not include experiments on WebArena, therefore we try our best to implement our own version and run on the same training tasks as our method.

C.3.3. Implementation details of AutoGuide

We directly run ReAct on the tasks with $\boldsymbol{\tau}_+$ to collect $\boldsymbol{\tau}_-$ actions and generate guidelines correspondingly. We provide a 5-shot prompt for the context identification module, which is shown in Figure 6. At test time, the top 2 guidelines at each timestep are selected to guide action generation.

Parameter name	Value
Allowed Episode Length	20
n-shots	2
Agent Model	gpt-4-1106-preview
Context Identification Model	gpt-4-1106-preview
Guideline Selection Model	gpt-4-1106-preview
Guideline Extraction Model	gpt-4-1106-preview
top-k guideline selection	2

Table 9. Experiment hyperparameters on WebArena. The number of shots follows the same setup in ReAct.

C.4. Real Websites

C.4.1. Environment details

We design a set of real-world website navigation tasks from 3 domains: Software Development (GitHub), Travel (Google Flights), and Education (Coursera), which have 30, 20, and 20 test tasks accordingly. Here are some example tasks:

- GitHub
  - Navigate to the repository for the Python library Seaborn with the most stars and show me all the open issues labeled with bug.
  - Go to the GitHub org for Spotify and open the pinned project with the most stars for me.
- Google Flights
  - Find me the one-way flight from Hong Kong to Paris departing on Oct 15th 2024 with the least emmisions.
  - Show me the booking options of the one-way flight departing from Auckland on September 11, 2024, and arriving in Rome with the earliest departure time on that day.
- Coursera
  - Show me a Cybersecurity course that can finish within 1 month and show me all the reviews for the selected course.
  - Find me a Coursera guided project that covers Unity and show me its main page.

We follow the action space design of Visual WebArena (Koh et al., 2024), which has the following action types available:

- goto [url]
- click [element_id]
- hover [element_id]
- type [element_id] [text] [1 for enter or 0 for not enter]

- press [key_combination]
- scroll [up or down]
- tab_focus [tab_index]
- close_tab
- go_back
- go_forward

As the real websites are constantly and dynamically changing, we evaluate the completed task with human experts.

C.4.2. Baseline and Models

We directly run the two-shot SoM algorithm in the official codebase of Visual WebArena. The only modification we made from the original codebase is the bounding box detection algorithm, in which we further filter invisible bounding boxes from the list and add the list elements as interactable elements to consider.

C.4.3. Implementation details of AutoGuide

We provide a total of 6 training tasks (3 for GitHub, 2 for Google Travel, and 1 for Coursera), on which we collect human demonstration as τ_+'s and run SoM to collect τ_-'s. From the pairs with multi-modal observations we generate text-based guidelines to guide action selection. Both context identification and guideline extraction are done by gpt-4-vision-preview, and we provide a 3-shot prompt for context identification. All parameter details are shown in Table 10.

Parameter name	Value
Allowed Episode Length	15
n-shots	2
Agent Model	gpt-4-vision-preview
Context Identification Model	gpt-4-vision-preview
Guideline Selection Model	gpt-4-vision-preview
Guideline Extraction Model	gpt-4-vision-preview
top-k guideline selection	2

Table 10. Experiment hyperparameters on multi-modal real-world website tasks.

D. Prompt Templates

D.1. Context Identification

We present our prompt templates for the context identification module $\mathcal{M}_{\text{context}}$ (Equation (1)) for WebShop, ALFWorld, and WebArena in Figures 4 to 6, respectively. In ALFWorld, there exist six categories of tasks, and we use context identification prompting with 2-shot examples for each task, following the practice by (Yao et al., 2022b). Figure 5 shows one example for the `pick_and_place` tasks.

D.2. Guideline Extraction

Figures 7 to 9 detail our prompt templates $\mathcal{M}_{\text{guideline}}$ for extracting guidelines (Equation (2)) in the WebShop, ALFWorld, and WebArena domains.

D.3. Context Matching

Figure 10 shows our prompt template for matching the generated context with one of the existing contexts if there is any similar context, for the construction of the set of context-aware guidelines (Section 2.2) and the retrieval of relevant guidelines during testing (Section 2.3) in all the three domains: WebShop, ALFWorld, and WebArena.

D.4. Guideline Selection

For selecting only k most relevant guidelines in case there are more corresponding context-aware guidelines during testing (Equation (3)), we use the prompt with Figure 11 for WebShop and ALFWorld and Figure 12 for WebArena.

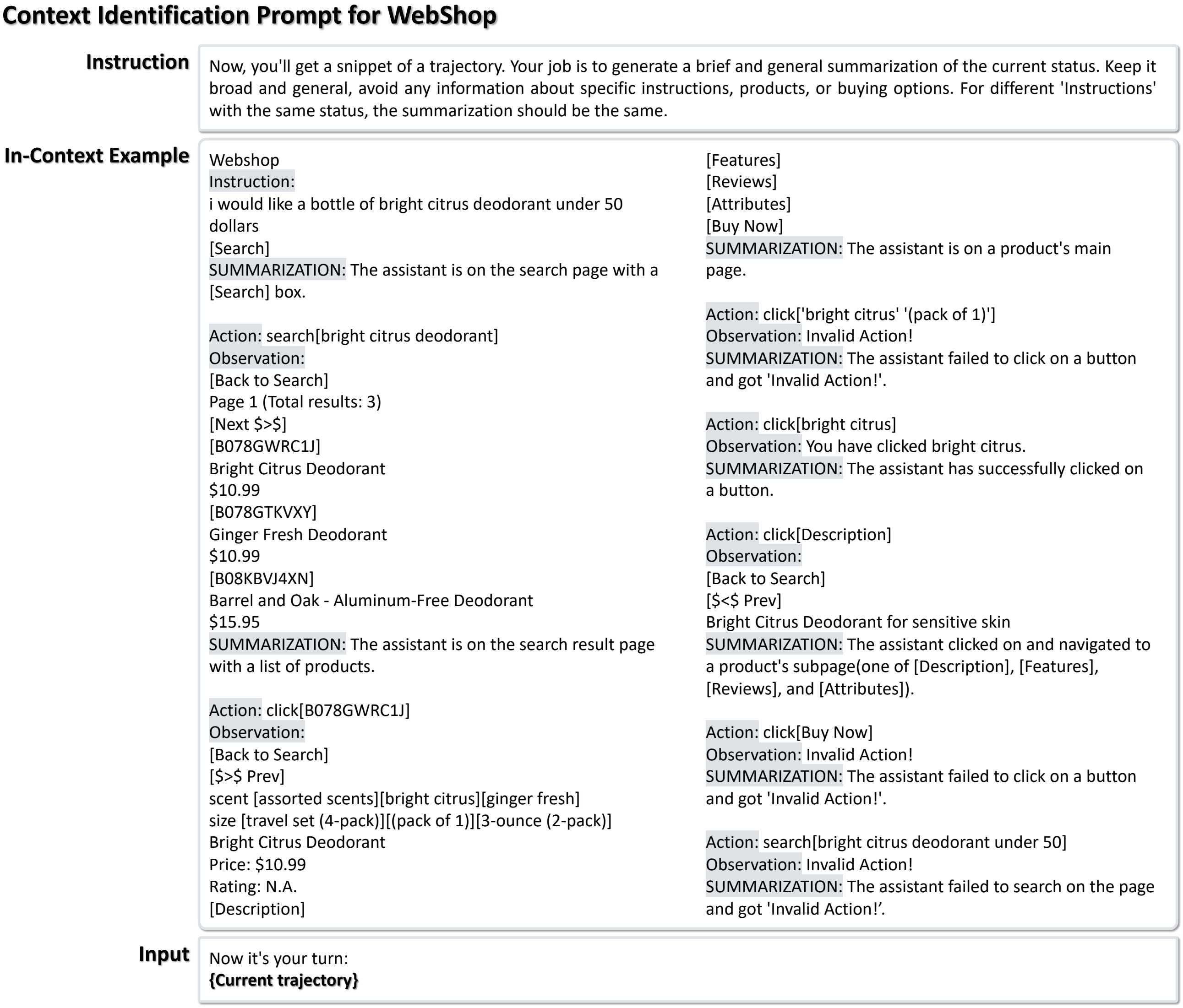
E. Example Context-Aware Guidelines

In Figures 13 and 14, we show a list of possible contexts and context-aware guidelines on WebArena and real-world websites, respectively.

F. Related Work

LLM-based agents. Language models have recently been shown to possess strong priors for sequential decision-making tasks, which has given rise to LLM-powered agents (Wang et al., 2023; Xi et al., 2023; Zheng et al., 2024; Zeng et al., 2023). Agents need to possess various skills to be effective in practice including planning (Brohan et al., 2023; Huang et al., 2022; Logeswaran et al., 2022), reasoning (Wei et al., 2022; Gao et al., 2023), tool manipulation (Qin et al., 2023; Patil et al., 2023; Parisi et al., 2022; Schick et al., 2023), code generation (Sun et al., 2023; Logeswaran et al., 2022), among others. In this work, we focus on building effective agents for web (Yao et al., 2022b; Zhou et al., 2023) and embodied (Shridhar et al., 2021) environments.

Self-reflection from past experiences. An important capability for agents to succeed is the ability to learn from past experiences and update their behavior based on feedback. Self-feedback (Madaan et al., 2023; Kim et al., 2023; Shinn et al., 2023) has emerged as an effective technique where a model inspects its own incorrect predictions, reflects on it to identify what went wrong and attempts to improve its prediction. While self-feedback provides intra-task (i.e.,



Context Identification Prompt for WebShop

Instruction

Now, you'll get a snippet of a trajectory. Your job is to generate a brief and general summarization of the current status. Keep it broad and general, avoid any information about specific instructions, products, or buying options. For different 'Instructions' with the same status, the summarization should be the same.

In-Context Example

```
Webshop
Instruction:
i would like a bottle of bright citrus deodorant under 50 dollars
[Search]
SUMMARIZATION: The assistant is on the search page with a [Search] box.

Action: search[bright citrus deodorant]
Observation:
[Back to Search]
Page 1 (Total results: 3)
[Next $>$]
[B078GWRC1J]
Bright Citrus Deodorant
$10.99
[B078GTKVXY]
Ginger Fresh Deodorant
$10.99
[B08KBVJ4XN]
Barrel and Oak - Aluminum-Free Deodorant
$15.95
SUMMARIZATION: The assistant is on the search result page with a list of products.

Action: click[B078GWRC1J]
Observation:
[Back to Search]
[$>$ Prev]
scent [assorted scents][bright citrus][ginger fresh]
size [travel set (4-pack)][(pack of 1)][3-ounce (2-pack)]
Bright Citrus Deodorant
Price: $10.99
Rating: N.A.
[Description]
[Features]
[Reviews]
[Attributes]
[Buy Now]
SUMMARIZATION: The assistant is on a product's main page.

Action: click['bright citrus' '(pack of 1)']
Observation: Invalid Action!
SUMMARIZATION: The assistant failed to click on a button and got 'Invalid Action!'.

Action: click[bright citrus]
Observation: You have clicked bright citrus.
SUMMARIZATION: The assistant has successfully clicked on a button.

Action: click[Description]
Observation:
[Back to Search]
[$<$ Prev]
Bright Citrus Deodorant for sensitive skin
SUMMARIZATION: The assistant clicked on and navigated to a product's subpage(one of [Description], [Features], [Reviews], and [Attributes]).

Action: click[Buy Now]
Observation: Invalid Action!
SUMMARIZATION: The assistant failed to click on a button and got 'Invalid Action!'.

Action: search[bright citrus deodorant under 50]
Observation: Invalid Action!
SUMMARIZATION: The assistant failed to search on the page and got 'Invalid Action!’.
```

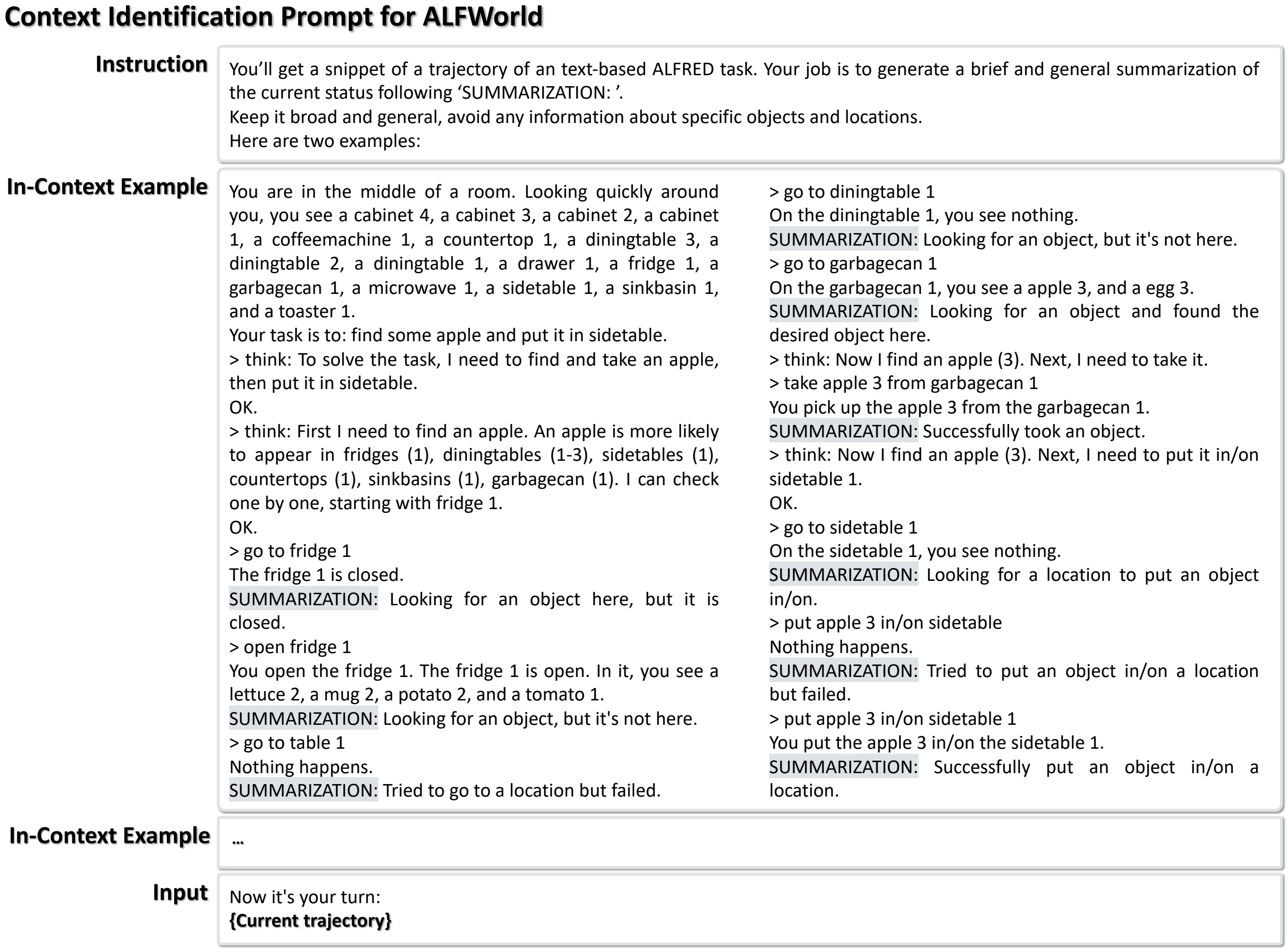
Input

Now it's your turn:
{Current trajectory}

Figure 4. Our prompt template for context identification (Equation (1)) in the WebShop domain.

per-episode) knowledge within a task based on immediate feedback, our approach offers an orthogonal and complementary aspect of inter-task knowledge (over multiple tasks) by considering multiple train tasks in offline data. AutoGuide enhances learning efficiency and credit assignment by utilizing detailed feedback from multiple tasks. However, these self-feedback approaches are complementary to AutoGuide and can be used in conjunction with our approach, as shown in our experiments (see Section 3.2).

Leveraging natural language guidance. Natural Language can be a rich source of information for agents to learn to act efficiently. Prior work has explored the notion of learning from human-written text manuals, which describe details about the environment (Branavan et al., 2012; Hanjie et al., 2021; Zhong et al., 2020). Recent work has explored automatically generating such guidance in the form of chain-of-thought reasoning (Wei et al., 2022; Yao et al., 2022b), which emulates a thought process or rationale for agent's predictions. In contrast to approaches which generate such guidance dynamically on the fly by imitating example guidance demonstrations provided by a human, our approach carefully compares trajectories in offline data to generate appropriate guidance and uses these guidelines for predicting better actions. ExpeL (Zhao et al., 2023) proposed a related approach to derive guidelines. In contrast to ExpeL, where all guidelines are provided to an agent as a prompt, our guideline selection process is contextual, where guidelines relevant to the agent's current state are retrieved and used for prediction. We show that this substantially improves over ExpeL's non-contextual guideline-based approach.



Context Identification Prompt for ALFWorld

Instruction

You'll get a snippet of a trajectory of an text-based ALFRED task. Your job is to generate a brief and general summarization of the current status following 'SUMMARIZATION: '.
Keep it broad and general, avoid any information about specific objects and locations.
Here are two examples:

In-Context Example

You are in the middle of a room. Looking quickly around you, you see a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 1, a diningtable 3, a diningtable 2, a diningtable 1, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sidetable 1, a sinkbasin 1, and a toaster 1.
Your task is to: find some apple and put it in sidetable.
> think: To solve the task, I need to find and take an apple, then put it in sidetable.
OK.
> think: First I need to find an apple. An apple is more likely to appear in fridges (1), diningtables (1-3), sidetables (1), countertops (1), sinkbasins (1), garbagecan (1). I can check one by one, starting with fridge 1.
OK.
> go to fridge 1
The fridge 1 is closed.
SUMMARIZATION: Looking for an object here, but it is closed.
> open fridge 1
You open the fridge 1. The fridge 1 is open. In it, you see a lettuce 2, a mug 2, a potato 2, and a tomato 1.
SUMMARIZATION: Looking for an object, but it's not here.
> go to table 1
Nothing happens.
SUMMARIZATION: Tried to go to a location but failed.
> go to diningtable 1
On the diningtable 1, you see nothing.
SUMMARIZATION: Looking for an object, but it's not here.
> go to garbagecan 1
On the garbagecan 1, you see a apple 3, and a egg 3.
SUMMARIZATION: Looking for an object and found the desired object here.
> think: Now I find an apple (3). Next, I need to take it.
> take apple 3 from garbagecan 1
You pick up the apple 3 from the garbagecan 1.
SUMMARIZATION: Successfully took an object.
> think: Now I find an apple (3). Next, I need to put it in/on sidetable 1.
OK.
> go to sidetable 1
On the sidetable 1, you see nothing.
SUMMARIZATION: Looking for a location to put an object in/on.
> put apple 3 in/on sidetable
Nothing happens.
SUMMARIZATION: Tried to put an object in/on a location but failed.
> put apple 3 in/on sidetable 1
You put the apple 3 in/on the sidetable 1.
SUMMARIZATION: Successfully put an object in/on a location.

In-Context Example

…

Input

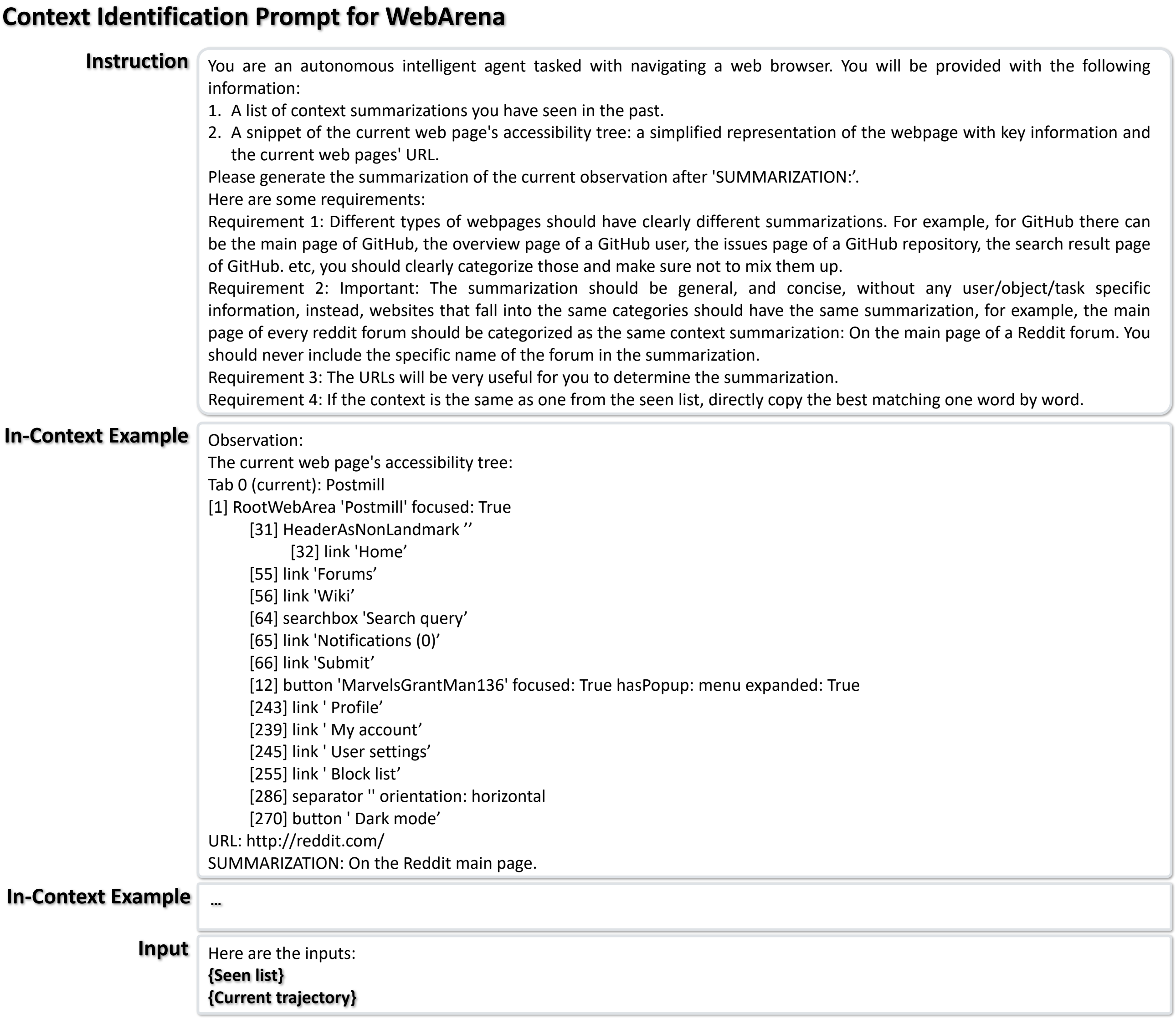
Now it's your turn:
{Current trajectory}

Figure 5. Our prompt template for context identification (Equation (1)) in the ALFWorld domain.

G. Limitation and Broader Impacts

Limitation. The performance of AutoGuide depends on the diversity of offline experiences. As such, one important direction for improvement is to automatically collect diverse offline experiences through continual learning, where we iteratively generate guidelines and use them to gather more trajectories with high rewards. Another avenue is the need for quantifying the quality of generated contexts and guidelines. Currently, apart from applying context-aware guidelines to ReAct and measuring test time performance, there lacks a standardized method for quantifying the quality of generated contexts and selected guidelines. Introducing a quantifiable metric to approximate the quality could pave the way for new optimization approaches such as reinforcement learning.

Broader impact. This paper introduces research aimed at enhancing the decision-making capabilities of LLMs. In terms of societal impact, while we develop a generic LLM-based autonomous agent, having biased offline datasets may lead to making decisions with suboptimal outcomes. Additionally, autonomous agents may be misused for malicious applications. To mitigate these risks, potential solutions would include diversifying datasets, implementing ethical oversight, ensuring transparency and accountability, engaging with stakeholders for a broader perspective, and incorporating security measures to prevent misuse. We believe that the research community, including ourselves, should responsibly advance LLM-based agent research, prioritizing societal well-being and ethical considerations.



Context Identification Prompt for WebArena

Instruction

You are an autonomous intelligent agent tasked with navigating a web browser. You will be provided with the following information:

1. A list of context summarizations you have seen in the past.
2. A snippet of the current web page's accessibility tree: a simplified representation of the webpage with key information and the current web pages' URL.

Please generate the summarization of the current observation after 'SUMMARIZATION:’.
Here are some requirements:
Requirement 1: Different types of webpages should have clearly different summarizations. For example, for GitHub there can be the main page of GitHub, the overview page of a GitHub user, the issues page of a GitHub repository, the search result page of GitHub. etc, you should clearly categorize those and make sure not to mix them up.
Requirement 2: Important: The summarization should be general, and concise, without any user/object/task specific information, instead, websites that fall into the same categories should have the same summarization, for example, the main page of every reddit forum should be categorized as the same context summarization: On the main page of a Reddit forum. You should never include the specific name of the forum in the summarization.
Requirement 3: The URLs will be very useful for you to determine the summarization.
Requirement 4: If the context is the same as one from the seen list, directly copy the best matching one word by word.

In-Context Example

```
Observation:
The current web page's accessibility tree:
Tab 0 (current): Postmill
[1] RootWebArea 'Postmill' focused: True
        [31] HeaderAsNonLandmark ’’
                [32] link 'Home’
        [55] link 'Forums’
        [56] link 'Wiki’
        [64] searchbox 'Search query’
        [65] link 'Notifications (0)’
        [66] link 'Submit’
        [12] button 'MarvelsGrantMan136' focused: True hasPopup: menu expanded: True
        [243] link ' Profile’
        [239] link ' My account’
        [245] link ' User settings’
        [255] link ' Block list’
        [286] separator '' orientation: horizontal
        [270] button ' Dark mode’
URL: http://reddit.com/
SUMMARIZATION: On the Reddit main page.
```

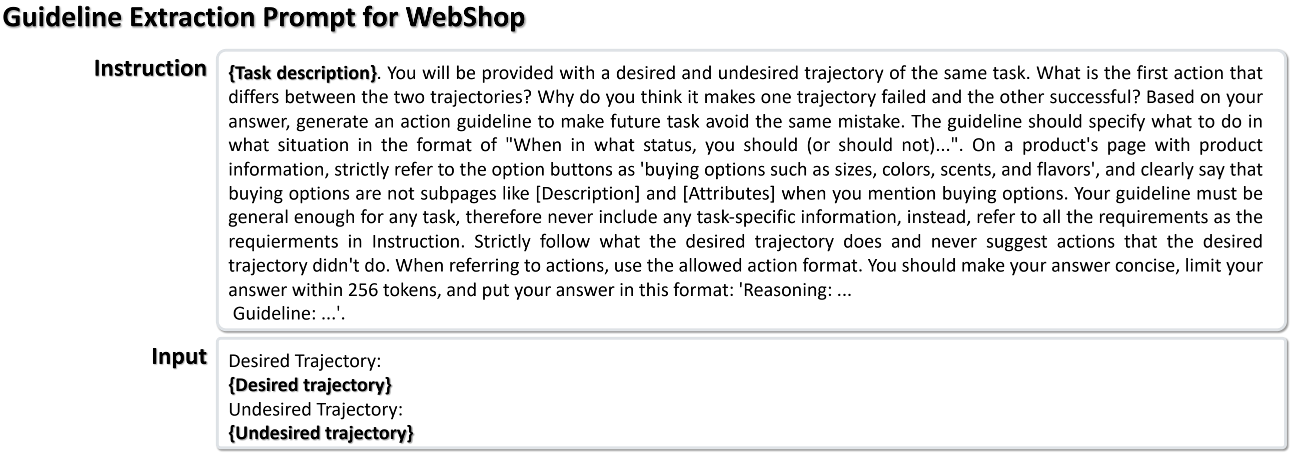
In-Context Example

…

Input

Here are the inputs:
{Seen list}
{Current trajectory}

Figure 6. Our prompt template for context identification (Equation (1)) in the WebArena domain.



Guideline Extraction Prompt for WebShop

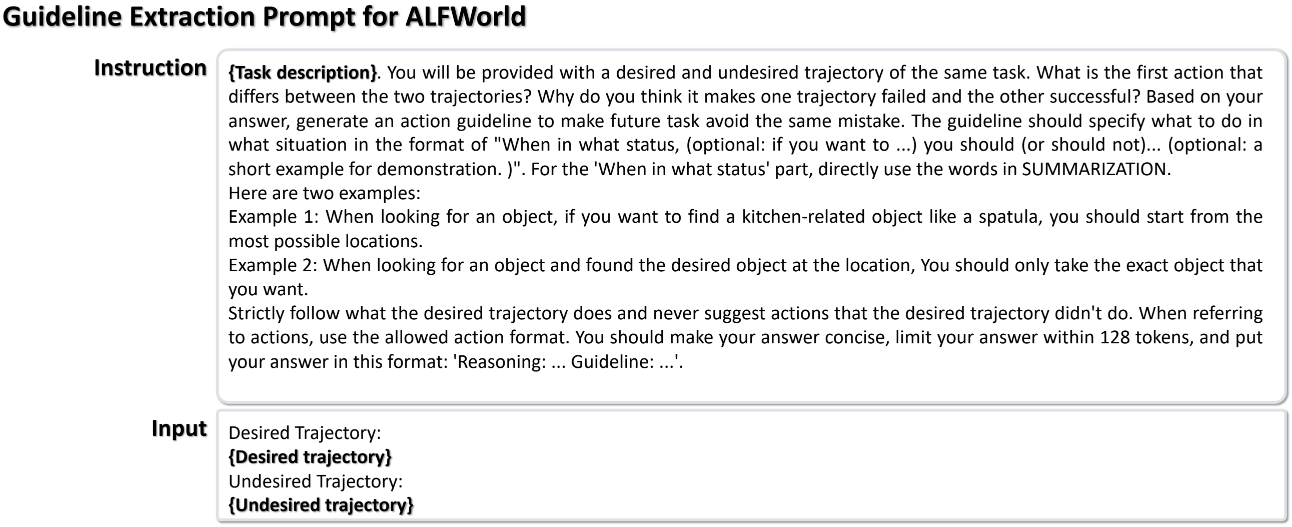
Instruction

{Task description}. You will be provided with a desired and undesired trajectory of the same task. What is the first action that differs between the two trajectories? Why do you think it makes one trajectory failed and the other successful? Based on your answer, generate an action guideline to make future task avoid the same mistake. The guideline should specify what to do in what situation in the format of "When in what status, you should (or should not)...". On a product's page with product information, strictly refer to the option buttons as 'buying options such as sizes, colors, scents, and flavors', and clearly say that buying options are not subpages like [Description] and [Attributes] when you mention buying options. Your guideline must be general enough for any task, therefore never include any task-specific information, instead, refer to all the requirements as the requierments in Instruction. Strictly follow what the desired trajectory does and never suggest actions that the desired trajectory didn't do. When referring to actions, use the allowed action format. You should make your answer concise, limit your answer within 256 tokens, and put your answer in this format: 'Reasoning: ...
Guideline: ...'.

Input

Desired Trajectory:
{Desired trajectory}
Undesired Trajectory:
{Undesired trajectory}

Figure 7. Our prompt template for guideline extraction (Equation (2)) in the WebShop domain.



Guideline Extraction Prompt for ALFWorld

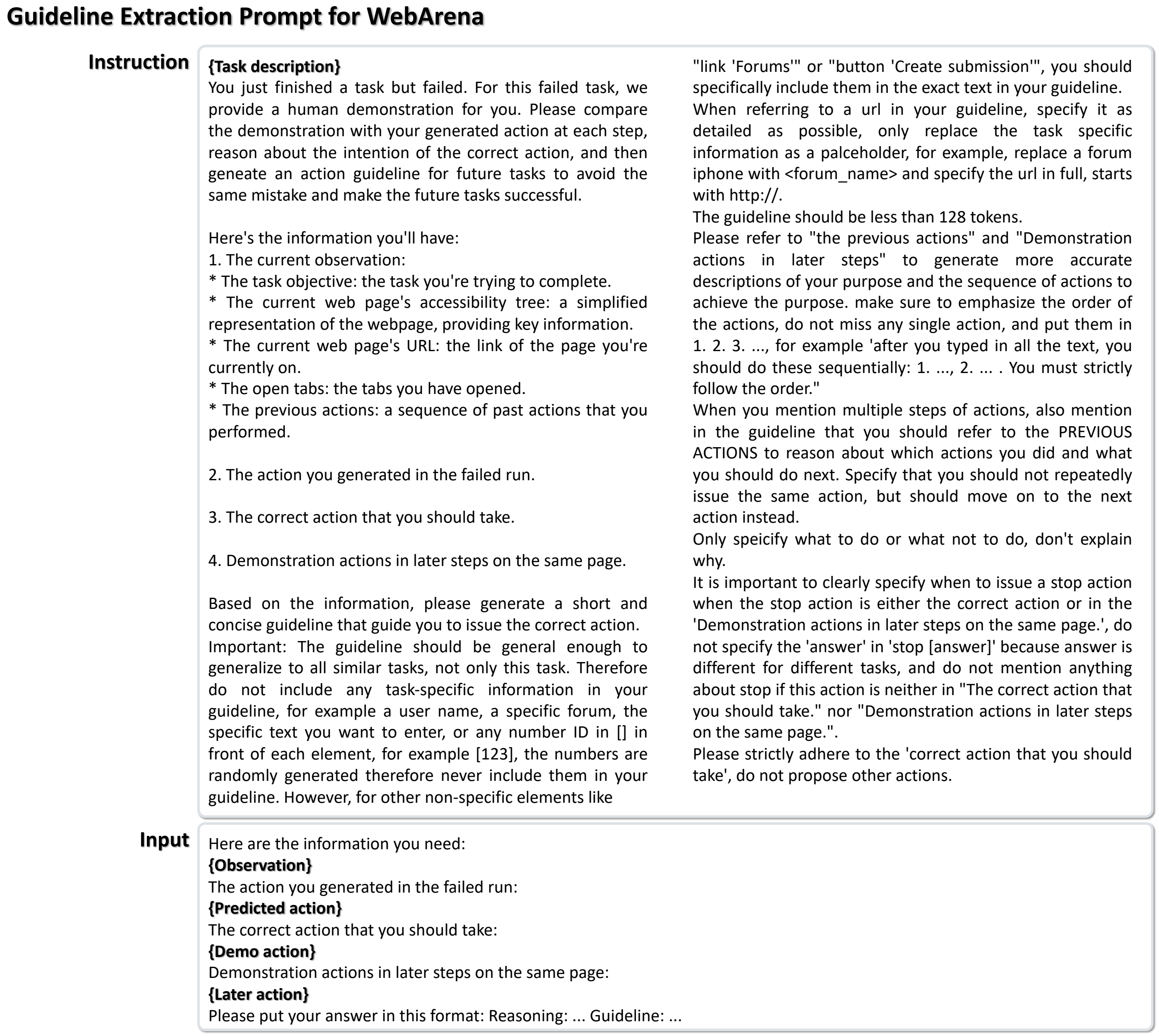
Instruction

{Task description}. You will be provided with a desired and undesired trajectory of the same task. What is the first action that differs between the two trajectories? Why do you think it makes one trajectory failed and the other successful? Based on your answer, generate an action guideline to make future task avoid the same mistake. The guideline should specify what to do in what situation in the format of "When in what status, (optional: if you want to ...) you should (or should not)... (optional: a short example for demonstration.)". For the 'When in what status' part, directly use the words in SUMMARIZATION.
Here are two examples:
Example 1: When looking for an object, if you want to find a kitchen-related object like a spatula, you should start from the most possible locations.
Example 2: When looking for an object and found the desired object at the location, You should only take the exact object that you want.
Strictly follow what the desired trajectory does and never suggest actions that the desired trajectory didn't do. When referring to actions, use the allowed action format. You should make your answer concise, limit your answer within 128 tokens, and put your answer in this format: 'Reasoning: ... Guideline: ...'.

Input

Desired Trajectory:
{Desired trajectory}
Undesired Trajectory:
{Undesired trajectory}

Figure 8. Our prompt template for guideline extraction (Equation (2)) in the ALFWorld domain.



Guideline Extraction Prompt for WebArena

Instruction

{Task description}
You just finished a task but failed. For this failed task, we provide a human demonstration for you. Please compare the demonstration with your generated action at each step, reason about the intention of the correct action, and then geneate an action guideline for future tasks to avoid the same mistake and make the future tasks successful.

Here's the information you'll have:
1. The current observation:
* The task objective: the task you're trying to complete.
* The current web page's accessibility tree: a simplified representation of the webpage, providing key information.
* The current web page's URL: the link of the page you're currently on.
* The open tabs: the tabs you have opened.
* The previous actions: a sequence of past actions that you performed.

2. The action you generated in the failed run.

3. The correct action that you should take.

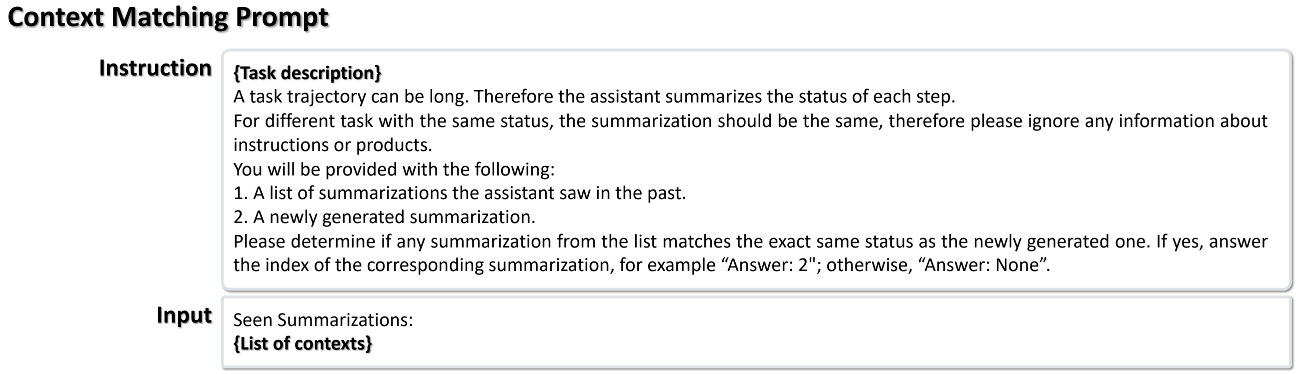
4. Demonstration actions in later steps on the same page.

Based on the information, please generate a short and concise guideline that guide you to issue the correct action.
Important: The guideline should be general enough to generalize to all similar tasks, not only this task. Therefore do not include any task-specific information in your guideline, for example a user name, a specific forum, the specific text you want to enter, or any number ID in [] in front of each element, for example [123], the numbers are randomly generated therefore never include them in your guideline. However, for other non-specific elements like "link 'Forums'" or "button 'Create submission'", you should specifically include them in the exact text in your guideline.
When referring to a url in your guideline, specify it as detailed as possible, only replace the task specific information as a palceholder, for example, replace a forum iphone with <forum_name> and specify the url in full, starts with http://.
The guideline should be less than 128 tokens.
Please refer to "the previous actions" and "Demonstration actions in later steps" to generate more accurate descriptions of your purpose and the sequence of actions to achieve the purpose. make sure to emphasize the order of the actions, do not miss any single action, and put them in 1. 2. 3. ..., for example 'after you typed in all the text, you should do these sequentially: 1. ..., 2. ... . You must strictly follow the order."
When you mention multiple steps of actions, also mention in the guideline that you should refer to the PREVIOUS ACTIONS to reason about which actions you did and what you should do next. Specify that you should not repeatedly issue the same action, but should move on to the next action instead.
Only speicify what to do or what not to do, don't explain why.
It is important to clearly specify when to issue a stop action when the stop action is either the correct action or in the 'Demonstration actions in later steps on the same page.', do not specify the 'answer' in 'stop [answer]' because answer is different for different tasks, and do not mention anything about stop if this action is neither in "The correct action that you should take." nor "Demonstration actions in later steps on the same page.".
Please strictly adhere to the 'correct action that you should take', do not propose other actions.

Input

Here are the information you need:
{Observation}
The action you generated in the failed run:
{Predicted action}
The correct action that you should take:
{Demo action}
Demonstration actions in later steps on the same page:
{Later action}
Please put your answer in this format: Reasoning: ... Guideline: ...

Figure 9. Our prompt template for guideline extraction (Equation (2)) in the WebArena domain.



Context Matching Prompt

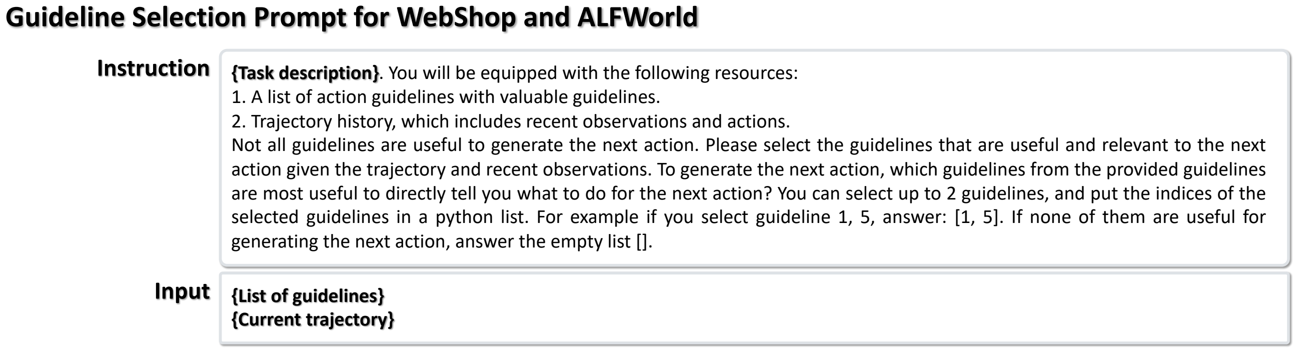
Instruction

{Task description}
A task trajectory can be long. Therefore the assistant summarizes the status of each step.
For different task with the same status, the summarization should be the same, therefore please ignore any information about instructions or products.
You will be provided with the following:
1. A list of summarizations the assistant saw in the past.
2. A newly generated summarization.
Please determine if any summarization from the list matches the exact same status as the newly generated one. If yes, answer the index of the corresponding summarization, for example “Answer: 2"; otherwise, “Answer: None”.

Input

Seen Summarizations:
{List of contexts}

Figure 10. Our prompt template for context matching (Sections 2.2 and 2.3) in all the WebShop, ALFWorld, and WebArena domains.



Guideline Selection Prompt for WebShop and ALFWorld

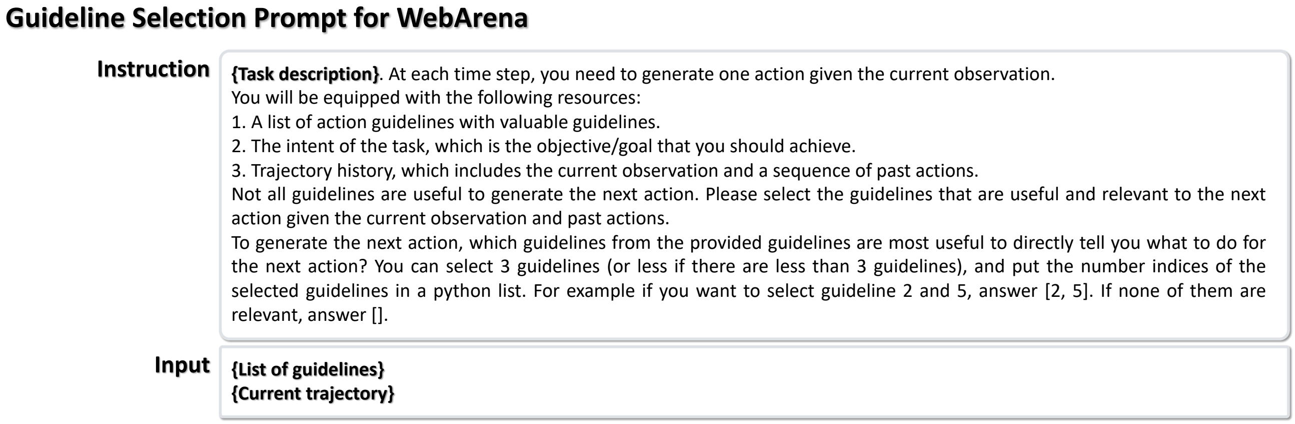
Instruction

{Task description}. You will be equipped with the following resources:
1. A list of action guidelines with valuable guidelines.
2. Trajectory history, which includes recent observations and actions.
Not all guidelines are useful to generate the next action. Please select the guidelines that are useful and relevant to the next action given the trajectory and recent observations. To generate the next action, which guidelines from the provided guidelines are most useful to directly tell you what to do for the next action? You can select up to 2 guidelines, and put the indices of the selected guidelines in a python list. For example if you select guideline 1, 5, answer: [1, 5]. If none of them are useful for generating the next action, answer the empty list [].

Input

{List of guidelines}
{Current trajectory}

Figure 11. Our prompt template for selecting the most relevant context-aware guidelines during the test time (Equation (3) from Section 2.3) in the WebShop and ALFWorld domains.



Guideline Selection Prompt for WebArena

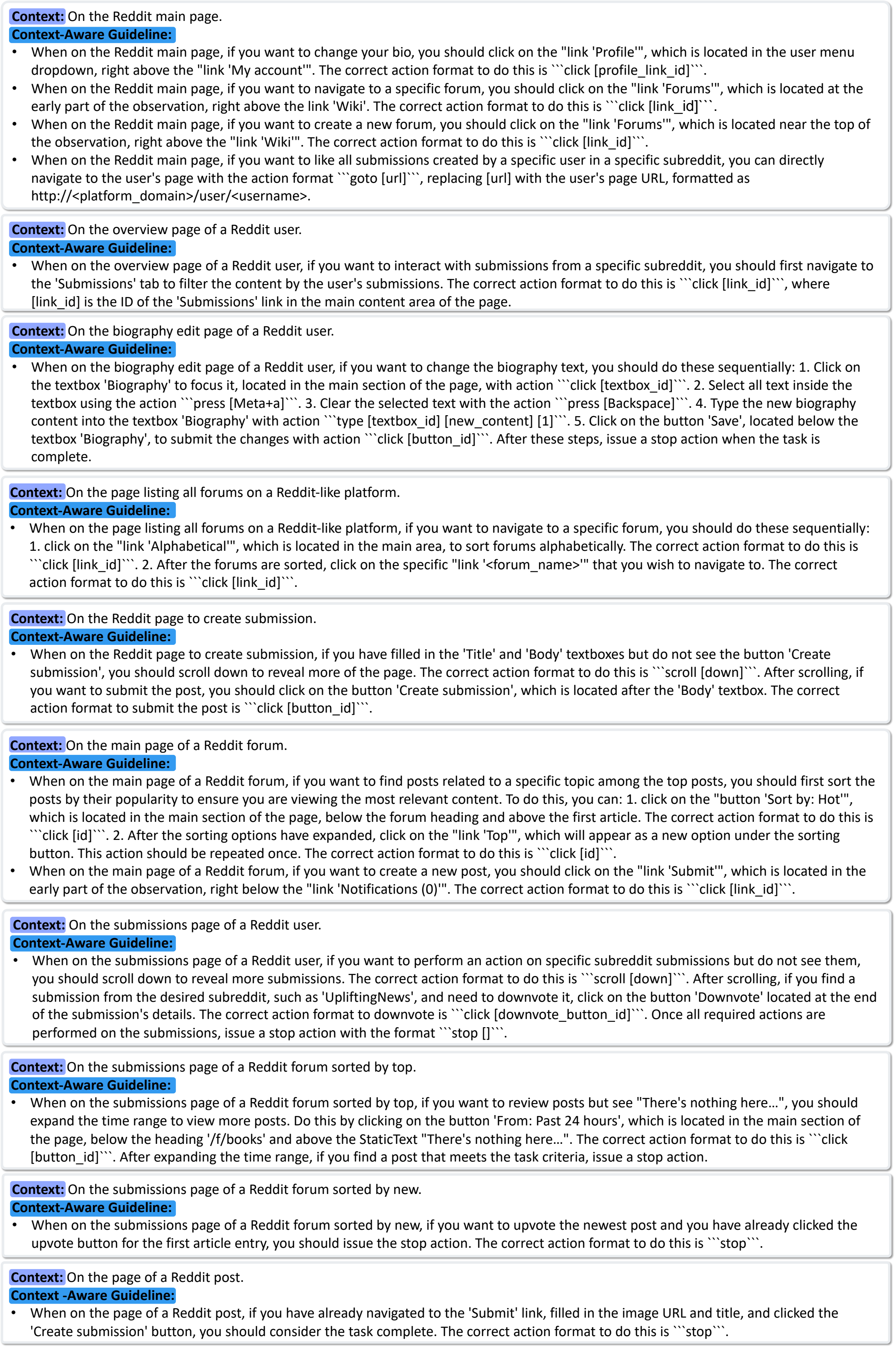
Instruction

{Task description}. At each time step, you need to generate one action given the current observation.
You will be equipped with the following resources:
1. A list of action guidelines with valuable guidelines.
2. The intent of the task, which is the objective/goal that you should achieve.
3. Trajectory history, which includes the current observation and a sequence of past actions.
Not all guidelines are useful to generate the next action. Please select the guidelines that are useful and relevant to the next action given the current observation and past actions.
To generate the next action, which guidelines from the provided guidelines are most useful to directly tell you what to do for the next action? You can select 3 guidelines (or less if there are less than 3 guidelines), and put the number indices of the selected guidelines in a python list. For example if you want to select guideline 2 and 5, answer [2, 5]. If none of them are relevant, answer [].

Input

{List of guidelines}
{Current trajectory}

Figure 12. Our prompt template for selecting the most relevant context-aware guidelines during the test time (Equation (3)) in the WebArena domain.



Context: On the Reddit main page.

Context-Aware Guideline:

- When on the Reddit main page, if you want to change your bio, you should click on the "link 'Profile'", which is located in the user menu dropdown, right above the "link 'My account'". The correct action format to do this is ```click [profile_link_id]```.
- When on the Reddit main page, if you want to navigate to a specific forum, you should click on the "link 'Forums'", which is located at the early part of the observation, right above the link 'Wiki'. The correct action format to do this is ```click [link_id]```.
- When on the Reddit main page, if you want to create a new forum, you should click on the "link 'Forums'", which is located near the top of the observation, right above the "link 'Wiki'". The correct action format to do this is ```click [link_id]```.
- When on the Reddit main page, if you want to like all submissions created by a specific user in a specific subreddit, you can directly navigate to the user's page with the action format ```goto [url]```, replacing [url] with the user's page URL, formatted as http://<platform_domain>/user/<username>.

Context: On the overview page of a Reddit user.

Context-Aware Guideline:

- When on the overview page of a Reddit user, if you want to interact with submissions from a specific subreddit, you should first navigate to the 'Submissions' tab to filter the content by the user's submissions. The correct action format to do this is ```click [link_id]```, where [link_id] is the ID of the 'Submissions' link in the main content area of the page.

Context: On the biography edit page of a Reddit user.

Context-Aware Guideline:

- When on the biography edit page of a Reddit user, if you want to change the biography text, you should do these sequentially: 1. Click on the textbox 'Biography' to focus it, located in the main section of the page, with action ```click [textbox_id]```. 2. Select all text inside the textbox using the action ```press [Meta+a]```. 3. Clear the selected text with the action ```press [Backspace]```. 4. Type the new biography content into the textbox 'Biography' with action ```type [textbox_id] [new_content] [1]```. 5. Click on the button 'Save', located below the textbox 'Biography', to submit the changes with action ```click [button_id]```. After these steps, issue a stop action when the task is complete.

Context: On the page listing all forums on a Reddit-like platform.

Context-Aware Guideline:

- When on the page listing all forums on a Reddit-like platform, if you want to navigate to a specific forum, you should do these sequentially: 1. click on the "link 'Alphabetical'", which is located in the main area, to sort forums alphabetically. The correct action format to do this is ```click [link_id]```. 2. After the forums are sorted, click on the specific "link '<forum_name>'" that you wish to navigate to. The correct action format to do this is ```click [link_id]```.

Context: On the Reddit page to create submission.

Context-Aware Guideline:

- When on the Reddit page to create submission, if you have filled in the 'Title' and 'Body' textboxes but do not see the button 'Create submission', you should scroll down to reveal more of the page. The correct action format to do this is ```scroll [down]```. After scrolling, if you want to submit the post, you should click on the button 'Create submission', which is located after the 'Body' textbox. The correct action format to submit the post is ```click [button_id]```.

Context: On the main page of a Reddit forum.

Context-Aware Guideline:

- When on the main page of a Reddit forum, if you want to find posts related to a specific topic among the top posts, you should first sort the posts by their popularity to ensure you are viewing the most relevant content. To do this, you can: 1. click on the "button 'Sort by: Hot'", which is located in the main section of the page, below the forum heading and above the first article. The correct action format to do this is ```click [id]```. 2. After the sorting options have expanded, click on the "link 'Top'", which will appear as a new option under the sorting button. This action should be repeated once. The correct action format to do this is ```click [id]```.
- When on the main page of a Reddit forum, if you want to create a new post, you should click on the "link 'Submit'", which is located in the early part of the observation, right below the "link 'Notifications (0)'". The correct action format to do this is ```click [link_id]```.

Context: On the submissions page of a Reddit user.

Context-Aware Guideline:

- When on the submissions page of a Reddit user, if you want to perform an action on specific subreddit submissions but do not see them, you should scroll down to reveal more submissions. The correct action format to do this is ```scroll [down]```. After scrolling, if you find a submission from the desired subreddit, such as 'UpliftingNews', and need to downvote it, click on the button 'Downvote' located at the end of the submission's details. The correct action format to downvote is ```click [downvote_button_id]```. Once all required actions are performed on the submissions, issue a stop action with the format ```stop []```.

Context: On the submissions page of a Reddit forum sorted by top.

Context-Aware Guideline:

- When on the submissions page of a Reddit forum sorted by top, if you want to review posts but see "There's nothing here…", you should expand the time range to view more posts. Do this by clicking on the button 'From: Past 24 hours', which is located in the main section of the page, below the heading '/f/books' and above the StaticText "There's nothing here…". The correct action format to do this is ```click [button_id]```. After expanding the time range, if you find a post that meets the task criteria, issue a stop action.

Context: On the submissions page of a Reddit forum sorted by new.

Context-Aware Guideline:

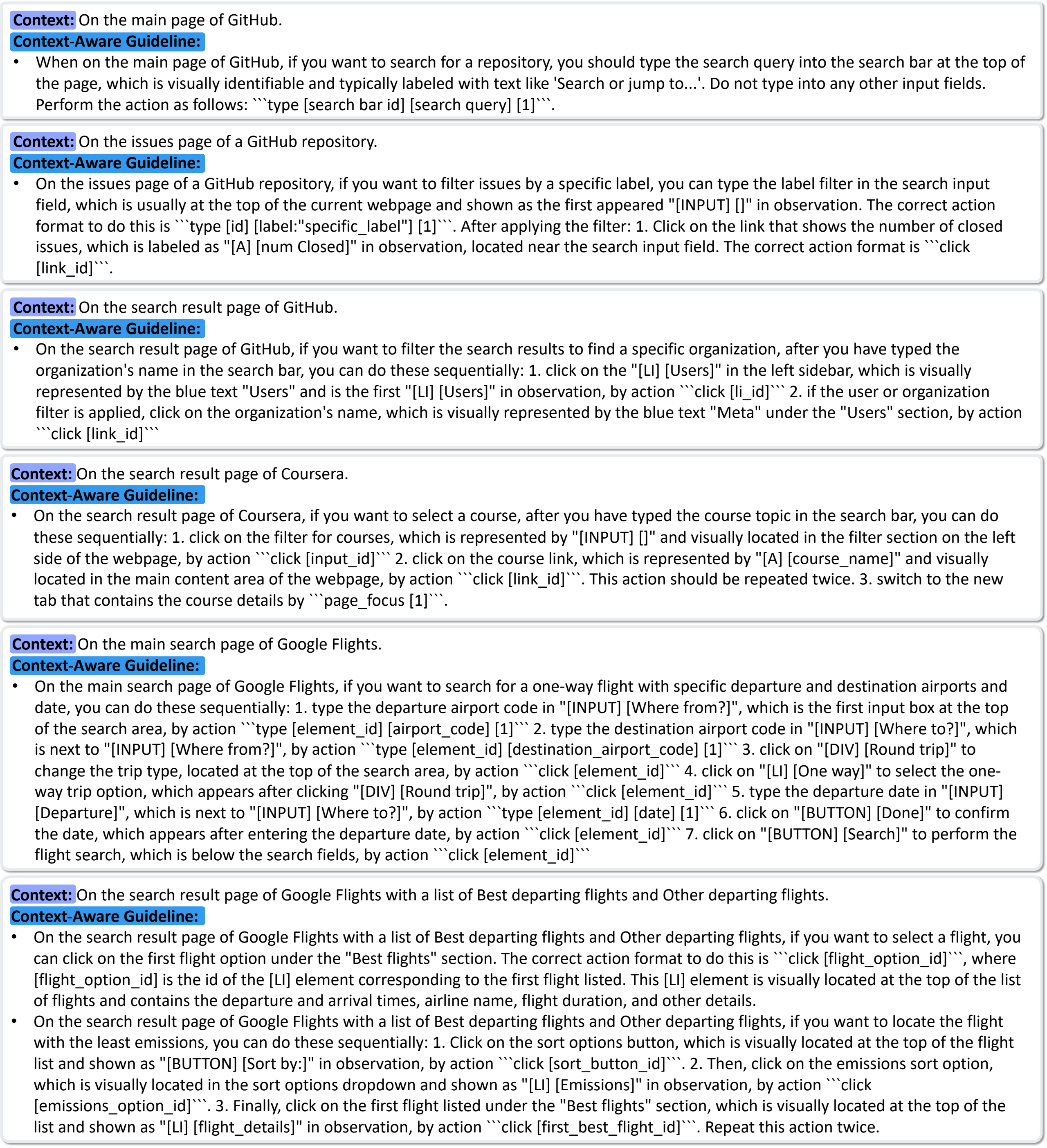
- When on the submissions page of a Reddit forum sorted by new, if you want to upvote the newest post and you have already clicked the upvote button for the first article entry, you should issue the stop action. The correct action format to do this is ```stop```.

Context: On the page of a Reddit post.

Context -Aware Guideline:

- When on the page of a Reddit post, if you have already navigated to the 'Submit' link, filled in the image URL and title, and clicked the 'Create submission' button, you should consider the task complete. The correct action format to do this is ```stop```.

Figure 13. Example contexts and corresponding guidelines for WebArena.



Context: On the main page of GitHub.

Context-Aware Guideline:

- When on the main page of GitHub, if you want to search for a repository, you should type the search query into the search bar at the top of the page, which is visually identifiable and typically labeled with text like 'Search or jump to...'. Do not type into any other input fields. Perform the action as follows: ```type [search bar id] [search query] [1]```.

Context: On the issues page of a GitHub repository.

Context-Aware Guideline:

- On the issues page of a GitHub repository, if you want to filter issues by a specific label, you can type the label filter in the search input field, which is usually at the top of the current webpage and shown as the first appeared "[INPUT] []" in observation. The correct action format to do this is ```type [id] [label:"specific_label"] [1]```. After applying the filter: 1. Click on the link that shows the number of closed issues, which is labeled as "[A] [num Closed]" in observation, located near the search input field. The correct action format is ```click [link_id]```.

Context: On the search result page of GitHub.

Context-Aware Guideline:

- On the search result page of GitHub, if you want to filter the search results to find a specific organization, after you have typed the organization's name in the search bar, you can do these sequentially: 1. click on the "[LI] [Users]" in the left sidebar, which is visually represented by the blue text "Users" and is the first "[LI] [Users]" in observation, by action ```click [li_id]``` 2. if the user or organization filter is applied, click on the organization's name, which is visually represented by the blue text "Meta" under the "Users" section, by action ```click [link_id]```

Context: On the search result page of Coursera.

Context-Aware Guideline:

- On the search result page of Coursera, if you want to select a course, after you have typed the course topic in the search bar, you can do these sequentially: 1. click on the filter for courses, which is represented by "[INPUT] []" and visually located in the filter section on the left side of the webpage, by action ```click [input_id]``` 2. click on the course link, which is represented by "[A] [course_name]" and visually located in the main content area of the webpage, by action ```click [link_id]```. This action should be repeated twice. 3. switch to the new tab that contains the course details by ```page_focus [1]```.

Context: On the main search page of Google Flights.

Context-Aware Guideline:

- On the main search page of Google Flights, if you want to search for a one-way flight with specific departure and destination airports and date, you can do these sequentially: 1. type the departure airport code in "[INPUT] [Where from?]", which is the first input box at the top of the search area, by action ```type [element_id] [airport_code] [1]``` 2. type the destination airport code in "[INPUT] [Where to?]", which is next to "[INPUT] [Where from?]", by action ```type [element_id] [destination_airport_code] [1]``` 3. click on "[DIV] [Round trip]" to change the trip type, located at the top of the search area, by action ```click [element_id]``` 4. click on "[LI] [One way]" to select the one-way trip option, which appears after clicking "[DIV] [Round trip]", by action ```click [element_id]``` 5. type the departure date in "[INPUT] [Departure]", which is next to "[INPUT] [Where to?]", by action ```type [element_id] [date] [1]``` 6. click on "[BUTTON] [Done]" to confirm the date, which appears after entering the departure date, by action ```click [element_id]``` 7. click on "[BUTTON] [Search]" to perform the flight search, which is below the search fields, by action ```click [element_id]```

Context: On the search result page of Google Flights with a list of Best departing flights and Other departing flights.

Context-Aware Guideline:

- On the search result page of Google Flights with a list of Best departing flights and Other departing flights, if you want to select a flight, you can click on the first flight option under the "Best flights" section. The correct action format to do this is ```click [flight_option_id]```, where [flight_option_id] is the id of the [LI] element corresponding to the first flight listed. This [LI] element is visually located at the top of the list of flights and contains the departure and arrival times, airline name, flight duration, and other details.
- On the search result page of Google Flights with a list of Best departing flights and Other departing flights, if you want to locate the flight with the least emissions, you can do these sequentially: 1. Click on the sort options button, which is visually located at the top of the flight list and shown as "[BUTTON] [Sort by:]" in observation, by action ```click [sort_button_id]```. 2. Then, click on the emissions sort option, which is visually located in the sort options dropdown and shown as "[LI] [Emissions]" in observation, by action ```click [emissions_option_id]```. 3. Finally, click on the first flight listed under the "Best flights" section, which is visually located at the top of the list and shown as "[LI] [flight_details]" in observation, by action ```click [first_best_flight_id]```. Repeat this action twice.

Figure 14. Example contexts and corresponding guidelines for real-world websites.