WEBLINX: REAL-WORLD WEBSITE NAVIGATION WITH MULTI-TURN DIALOGUE

Xing Han Lù*
Mila Quebec AI Institute
McGill University
xing.han.lu@mail.mcgill.ca

Zdeněk Kasner*†
ÚFAL
Charles University
kasner@ufal.mff.cuni.cz

Siva Reddy
Mila Quebec AI Institute
McGill University
Facebook CIFAR AI Chair
siva.reddy@mila.quebec

ABSTRACT

We propose the problem of *conversational web navigation*, where a digital agent controls a web browser and follows user instructions to solve real-world tasks in a multi-turn dialogue fashion. To support this problem, we introduce WEBLINX – a large-scale benchmark of 100K interactions across 2300 expert demonstrations of conversational web navigation. Our benchmark covers a broad range of patterns on over 150 real-world websites and can be used to train and evaluate agents in diverse scenarios. Due to the magnitude of information present, Large Language Models (LLMs) cannot process entire web pages in real-time. To solve this bottleneck, we design a retrieval-inspired model that efficiently prunes HTML pages by ranking relevant elements. We use the selected elements, along with screenshots and action history, to assess a variety of models for their ability to replicate human behavior when navigating the web. Our experiments span from small text-only to proprietary multimodal LLMs. We find that smaller finetuned decoders surpass the best zero-shot LLMs (including GPT-4V), but also larger finetuned multimodal models which were explicitly pretrained on screenshots. However, all finetuned models struggle to generalize to unseen websites. Our findings highlight the need for large multimodal models that can generalize to novel settings. Our code, data and models are available for research: https://mcgill-nlp.github.io/weblinx.

1 INTRODUCTION

Proprietary conversational assistants like ChatGPT (OpenAI, 2022) are capable of more than just conversing; they can also browse websites through plugins (OpenAI, 2023d; Pinsky, 2023), allowing them to perform actions and provide more useful responses. However, this capability is limited: the plugins must be developed separately for each website and may not cover all of a website's functionality. This limitation raises an important research question: can we leverage the models behind those assistants to navigate websites directly in the user's browser, while retaining their conversational capabilities?

Motivated by this question, we define the problem of **conversational web navigation**: given the initial user instruction, an agent must complete a real-world task inside a web browser while communicating with the user via multi-turn dialogue. This problem is relevant in many real-world scenarios: helping visually impaired users efficiently navigate websites through a chat interface, enhancing smart speakers and digital assistants with voice-controlled web navigation, and improving the productivity of knowledge workers by reducing highly repetitive steps while staying in control. From a research perspective, this problem can be used to assess the ability of LLM agents to not only follow self-contained instructions, but also engage with their environment through dialogue and generalize to unforeseen situations.

To address this problem, we introduce **WEBLINX**[1] (§3), a benchmark containing 2337 demonstrations of *conversational web navigation* produced by human experts across 155 real-world websites.

*These authors contributed equally to this work.
†Work done during an internship at Mila Quebec AI Institute.
[1]**Web L**anguage **I**nterface for **N**avigation & e**X**ecuting actions

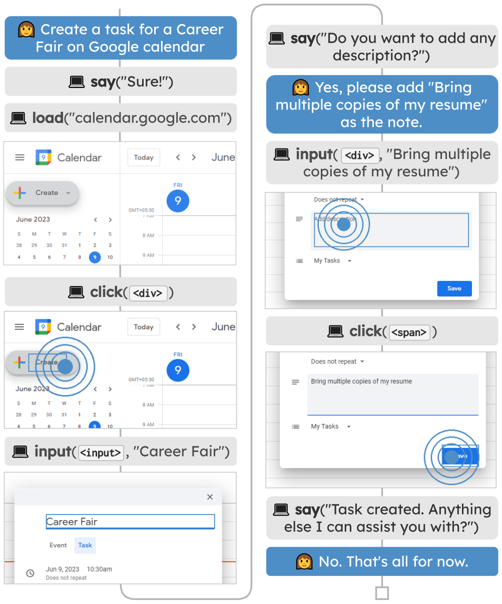


Figure 1: An example of the *conversational web navigation* task. The instructor (**blue**) communicates with the navigator (**grey**) using only natural language. The latter controls the browser, having access to screenshots and textual website representation.

Table 1: Overview of the WEBLINX core action space. We shorten `textinput` to `input`. For full set of actions, see Table 7.

Action	Description
`click(element)`	click on an element
`load(url)`	load URL of a new page
`say(text)`	navigator's utterance
`submit(element)`	submit a form
`input(elem,val)`	type text into the element

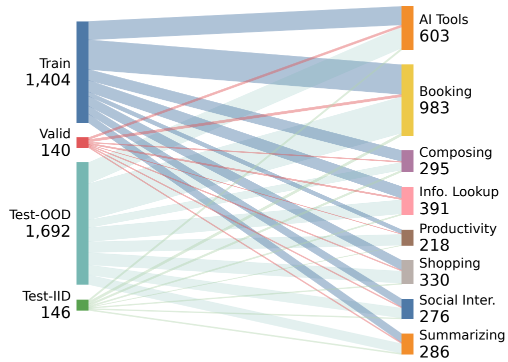


Figure 2: Distribution of demonstrations in WEBLINX across categories (Section 5.2) and splits (Table 4). Each category has many subcategories as shown in Appendix A.2.

Figure 1 shows a demonstration. Each demonstration captures the full sequence of actions performed by a human *navigator* when interacting with the user (known as *instructor*) through a conversational interface. We record over 100K occurrences of actions and utterances, where each action is associated with a Document Object Model (DOM)[2] tree, browser screenshots, and frames from demonstration-level video recordings. Table 2 highlights the unique aspects of WEBLINX. Unlike previous works focused on mobile apps or specialized applications, ours is the first large-scale benchmark that can be used to train dialogue-enabled navigation agents and evaluate their generalization capabilities to realistic scenarios, such as adapting to new websites, categories, and geographies; we also reserve a split to assess the ability of agents to interact with instructors without visual access to the browser.

A naive way to use this benchmark would be to give the full DOM tree directly to an agent and instruct it to predict the correct action. As some HTML pages contain thousands of elements, fitting them completely within the context of a LLM poses a significant challenge; even if it was possible, existing LLMs would be unable to process them in real-time. Consequently, we design a method called *Dense Markup Ranking* (§5.1), which compares each element in an HTML page with the full action history. By using a similarity-based approach to both learn and rank elements, we can leverage compact architectures used in text retrieval. This lets us find the most relevant elements and prune irrelevant ones to obtain a compact representation of the DOM. We combine it with the action history, detailed instruction and screenshot (in a multimodal context) to construct an input representation for LLMs, which can now meaningfully predict which actions to take. However, even if a predicted action is correct, it may be identified as incorrect by existing metrics, which can happen when there are minor differences in an agent's response or when an overlapping element is selected. Thus, we design a suite of evaluation metrics (§4) tailored for specific types of action (for instance, *clicking* should be evaluated differently from what the navigator *says*).

We examine 19 models based on 8 architectures (§6), including smaller image-to-text, larger text-only decoders, LLMs, and multimodal models (capable of accessing both image and text). Among them, 5

[2] Tree representation of HTML page as rendered in the browser.

Table 2: WEBLINX is the first benchmark featuring real-world websites with multi-turn dialogue. The columns indicate: use of multi-turn dialogue (*Chat*), if tasks are general or specialized (*Gener.*), a web browser is used (*Browse*), number of app/website domains (*# Dom.*), number of instances (*# Inst.*), average number of HTML elements per page (*Avg. # El.*), average number of turns per instance (*Avg. # Turns*). *AITW has 30K unique prompts with multiple demos each and the browsing data is strictly from Android devices.

Benchmark	Chat	Gener.	Browse	# Dom.	# Inst.	Avg. # El.	Avg. # Turns	Setting
MiniWob++ (Liu et al., 2018)	✗	✗	✗	100	100	28	3.6	Simplified
WebShop (Yao et al., 2022)	✗	✗	✓	1	12K	38	11.3	E-Commerce
WebArena (Zhou et al., 2023)	✗	✓	✓	6	812	-	-	Real-world
VWA (Koh et al., 2024)	✗	✓	✓	3	910	-	-	Real-world
WebVoyager (He et al., 2024)	✗	✓	✓	15	300	-	-	Real-world
Mind2Web (Deng et al., 2023)	✗	✓	✓	137	2350	1135	7.3	Real-world
AITW* (Rawles et al., 2023)	✗	✓	✓	357	30K	-	6.5	Android/Apps
RUSS (Xu et al., 2021)	✓	✗	✓	22	80	801	5.4	Help center
WorkArena (Drouin et al., 2024)	✓	✗	✓	1	23K	-	10	ServiceNow
META-GUI (Sun et al., 2022)	✓	✓	✗	11	1125	79	4.3	Mobile apps
WEBLINX (ours)	✓	✓	✓	155	2337	1775	43.0	Real-world

are in the zero-shot setting, and the remaining are finetuned using the training split of WEBLINX. We find that even the best zero-shot model, GPT-4V (OpenAI, 2023a), is surpassed by finetuned models (§6.1). Notably, a smaller model like Sheared-LLaMA (Xia et al., 2023) outperforms the much larger Fuyu (Bavishi et al., 2023), which was pretrained with browser screenshots. However, all models face challenges in generalizing to new settings, such as unseen websites from a different geographic location or when the instructor gives instructions without seeing the screen. Those findings prompted us to qualitatively look at the behavior of the models (§6.2), where we find that GPT-4V lacks situational awareness and can make obvious blunders. However, the best finetuned models still fail in simple cases, such as clicking on non-existing links or failing to change the language of a translation app. Thus, we believe that significant effort will be needed to make progress on the problem of *conversational web navigation*, as we discuss in Section 7.

Our contributions are summarized as follows:

- We introduce the task of **conversational web navigation** and a large-scale expert-annotated benchmark for it, named WEBLINX (§3).
- We propose a suite of action-specific metrics, which we combine to assess overall model performance (§4).
- We design a method to simplify HTML pages (§5.1), allowing us to evaluate a wide range of models (§5.2).
- We find that smaller text-only decoders outperform multimodal LLMs, but all finetuned models struggle to generalize to novel scenarios (§6).

2 RELATED WORK

2.1 WEB NAVIGATION AGENTS

Previous work predominantly focused on building web agents for a single task. A prominent work for task-driven web navigation is MiniWoB++ (Shi et al., 2017; Liu et al., 2018), a simulated web environment with an extensive list of task primitives (e.g., select value from a dropdown or date from a calendar). Its well-defined input space and the flexibility of its simulated environments lead to reinforcement learning approaches reaching human-level performance (Liu et al., 2018; Humphreys et al., 2022). However, the ability of those methods to transfer to realistic settings have been limited, even after introducing environment extensions (Gur et al., 2021) and sample-efficient methods (Kim et al., 2023). Other works also explored grounding language commands to web elements and mobile UIs (Pasupat et al., 2018; Li et al., 2020; Burns et al., 2022), or question answering (QA) by navigating Wikipedia (Nogueira & Cho, 2016).

In an effort to build more realistic environments, Yao et al. (2022) introduced WebShop, an e-commerce environment with over 12K human-written task instructions. Models trained on WebShop

Table 3: Definition of state (§3.1)

	Description
c_t	Candidate elements that can be targeted by a_t
d_t	Current DOM tree of the page
i_t	Screenshot of the navigator's browser
u_t	Instructor's utterance
v_t	Viewport size (height and width)
h_t	Interaction history

Table 4: Demonstration (demo) splits for training and evaluation.

Split	Description
TRAIN	Demos used to train models in Section 5
VALID	Demos for hyperparameters selection
TEST_{IID}	Demos to test in-domain generalization
TEST_{OOD}	*Aggregation of splits for OOD evaluation*
TEST_{WEB}	Unseen websites from the same subcategories
TEST_{CAT}	New subcategories within the same categories
TEST_{GEO}	Geographic locations not in TRAIN
TEST_{VIS}	Instructor does not see the screen

achieved strong performance, but still relied on clean HTML and simple visual representations (Furuta et al., 2023). Instead, we aim to build agents that can act on *any real-world website*, often existing in noisy and dynamic environments.

The prospect of using LLMs to act on real websites (Nakano et al., 2021) has lead to the development of LLM-based navigation services (Adept, 2023; Multi-On, 2023; HyperWrite, 2023), which has set the stage for academic counterparts. MIND2WEB (Deng et al., 2023) and WebArena (Zhou et al., 2023) are large-scale resources for building autonomous navigation agents like SeeAct (Zheng et al., 2024) and WebVoyager (He et al., 2024). On the other hand, WEBLINX is a benchmark for building agents that can interact with users in a multi-turn dialogue fashion, allowing them to be steered towards precise goals.

2.2 Website Representations

Efficiently representing real-world websites is a long-standing challenge in web understanding (Wu et al., 2023), including subtasks like web information extraction (Chang et al., 2006) and web segmentation (Kiesel et al., 2020). The approaches for simplifying or compressing the *textual* representation of the website – its HTML code or DOM tree – include rule-based algorithms (Zhou et al., 2021), accessibility-tree representations offered by browsers (Assouel et al., 2023), graph embeddings (Wang et al., 2022), and model-based approaches (Deng et al., 2022; Li et al., 2022; Aghajanyan et al., 2022). Previous works for representing the *visual* information of the webpage usually rely on feature extraction (Liu et al., 2010; Cormer et al., 2017), closely following the research on graphical UIs (Wu et al., 2021; Bunian et al., 2021). We propose a novel dense markup retriever which selects relevant DOM elements, and use these elements optionally combined high-resolution browser screenshots.

2.3 Conversational Interfaces

Using conversational interfaces to complete tasks is the basis of task-oriented dialogue (Chen et al., 2017; Zhang et al., 2020b). End-to-end solutions have shown promising results (Zhang et al., 2020a; Kann et al., 2022), but the use of LLMs remains under scrutiny (Hudeček & Dušek, 2023). For real-world services, Dialog2API (Shu et al., 2022) proposed an interface for interacting with API-based services, whereas META-GUI (Sun et al., 2022) introduced a dataset focused on automating actions in mobile apps rather than general websites. In terms of dialogue-centric web navigation, RUSS (Xu et al., 2021) is the first dataset designed to help support services through 80 demonstrations annotated with a domain-specific language. WEBLINX extends previous dialogue-centric datasets by covering a wide range of real-world tasks spanning 2337 demonstrations, with considerably longer demonstrations due to dynamic topic switching, a subject studied by Adlakha et al. (2022).

3 WebLINX

In this section, we introduce WEBLINX, a large-scale benchmark for conversational web navigation consisting of 2337 demonstrations with an average of 43 turns. It contains interactions between a human user (referred to as *instructor*) and human assistant (*navigator*) aiming to complete tasks across 155 real-world websites selected from 15 geographic areas. We classify the websites into 8 categories and 50 subcategories based on their domains.

Statistics The data statistics are summarized in Table 2 and a breakdown by category and split is illustrated by Figure 2. Additional statistics about the dataset, including the number of demonstrations in split, can be found in Appendix A.1, along with the list of categories in Appendix A.2.

Demonstration Framework The demonstrations capture real-time interactions, which are recorded by the navigator controlling the web browser. Each demonstration $\mathcal{D} = \{s_1, a_1, \ldots, s_n, a_n\}$ is a sequence of n states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$. At each turn $t \in \{1, \ldots, n\}$, the state s_t contains the representation of the website. Each action follows one of the 5 core intents described in Figure 1. The full list of intents is provided in Section A.6.

Data Collection To collect the demonstrations, we worked with a professional data labeling company,[3] who enlisted 8 expert annotators that received detailed instructions and extensive training to complete our tasks. The annotators worked in pairs: an instructor interacts with a navigator who completes the tasks in a web browser. Both use the chat interface to communicate, but only the navigator controls the browser. We designed an app, browser extension, and processing pipeline to record the demonstrations, which are subsequently validated by a different annotator under the supervision of the original navigator (details in Appendix A.5).

Evaluation Splits In addition to a TRAIN split, we create VALID and TEST_{IID} to assess in-domain generalization, and 4 out-of-domain splits for various scenarios (see Table 4).

3.1 REPRESENTING ACTIONS AND STATES FOR MODELING

At each turn t, we have access to the state s_t to predict an action a_t. The state consists of the components presented in Table 3.

Note that a state need not contain all of the above. For example, at the start of a demonstration, the instructor and navigator may need multiple rounds of dialogue to properly define the objective, in which case the initial states do not have DOM trees or screenshots. A model m predicts an action a_t for a given state s_t based on a prompt template p_m which indicates how to make use of the contents in a state.

Interaction history Since a model m has a limited input length in practice, we represent history h as the set of past five actions (denoted as a_r) and five utterances (u_r). We could not include the representation of past states such as elements or screenshots.

Parsing Action Output An action consists of an intent and argument and can be generated by an agent in a textual format. It must follow a pre-defined structure (see Figure 1) that allows it to be parsed into a structured form, which can be executed in a browser using tools like Selenium.[4] We discuss additional details in Appendix A.4.

4 EVALUATION FRAMEWORK

4.1 METRICS

A commonly used metric in prior work on web navigation is *task success rate*, which measures the proportion of demonstrations where the model reached the desired final state (Shi et al., 2017; Yao et al., 2022; Deng et al., 2023). However, this metric is inappropriate for our benchmark because the objective is not fully defined in the first turn or later turns; instead, it evolves as the conversation proceeds. We instead leverage *turn-level* automatic evaluation metrics, following established approaches in dialogue systems (Rastogi et al., 2020; Zhang et al., 2020a). The aim of the metrics is to provide a heuristic estimate of the similarity between the predicted action and the reference action.

Intent Match (IM) Given prediction a' and reference a, the intent match is $\text{IM}(a', a) = 1$ if the intents are equal, otherwise $\text{IM}(a', a) = 0$. This tells us if a model can correctly identify which action to perform, but does not indicate if the model can predict the correct arguments.

[3]EsyCommerce: esycommerce.com
[4]https://www.selenium.dev/

Element Similarity using IoU For actions with elements as arguments (`click`, `textinput`, `submit`), we compute the **intersection over union** (**IoU**; Jaccard 1912):

$$\mathrm{IM}(a', a) \times (\mathcal{B}_{\text{reference}} \cap \mathcal{B}_{\text{predicted}}) \div (\mathcal{B}_{\text{reference}} \cup \mathcal{B}_{\text{predicted}})$$

Where $\mathcal{B}$ is the bounding box area; to compute it, we use `(x,y)` coordinates of the reference and predicted bounding boxes. This formulation (1) favors elements with high visual overlap, (2) penalizes predicting elements much smaller or larger than reference elements even if one is completely contained by the other, and (3) assigns 0 if the elements do not overlap.

Text Similarity using F1 To measure lexical similarity of text arguments in `say` and `textinput`, we calculate **chrF** (Popovic, 2015), an F1-score for character n-gram matches (we use the default setting of $n = 6$). Similar to IoU, we scale by the IM, resulting in $\mathrm{IM}(a', a) \times \textsc{chrF}(a', a)$. In the case of `load` intent, URLs follow a structure that can be consistently segmented, which leads us to apply the F1-score on segments instead of n-grams; we call this measure **URLF**. We use **F1** to refer to either chrF and URLF, depending on whether an action contains a text or URL argument.

4.2 Turn-level score and overall score

To allow better comparisons between models, we divide the intents into groups: The **element group (EG)** contains `click`, `textinput`, and `submit`, and is evaluated with IoU. The **text group (TG)** encompasses `load`, `say`, and `textinput`, and is evaluated with F1.

We assign a turn level score based on the following: If the turn involves an action in EG, the score is the same as IoU: 0 when the intent is incorrect or the element doesn't overlap, and 1 when intent is correct and the element perfectly overlaps, and it is somewhere in between for the rest. For TG actions `load` and `say`, the score mirrors F1: 0 when either intent is incorrect or there is no text overlap, and 1 when intent is correct and the text matches exactly, and it is somewhere in between for the rest. For `textinput`, the turn score is IoU $\times$ F1 since it contains both text and element arguments. We finally compute the **overall score** using the **micro-average** of turn-level scores.

5 Methods

In this section, we describe a method for selecting candidate elements (§5.1) and how to use them in textual input. We use these methods to build models that can accurately predict actions (§5.2). We report results in Section 6 and provide implementation details in Appendix B.

5.1 Dense Markup Ranking for Candidate Selection and Input Representation

To choose a set of suitable candidates for the model input (§3.1), we need a candidate selection stage that filters the full set of elements in the DOM tree. Deng et al. (2023) proposed to pair each DOM element with the task query and input them into a DeBERTa model (He et al., 2021), which is finetuned using a cross-encoder loss (Reimers & Gurevych, 2019). We found this method takes on average 916ms to select candidates for a given turn.[5] When factoring in network latency and LLM inference, this would result in poor processing time. It is thus crucial that we use efficient ranking method to build agents that can operate in real time and learn from interactions with users.

To solve this, we propose **Dense Markup Ranking (DMR)**, which is 5 times faster than the previous approach, at the cost of slightly lower recall. The method consists of: (1) a simplified element representation to reduce computational overhead; (2) a dual encoder-based approach (Reimers & Gurevych, 2019; Karpukhin et al., 2020); (3) similarity-based learning between the text representation of s_t and $a_{1:t-1}$ and corresponding HTML elements. Using this method, we finetune a variant of *MiniLM* (Wang et al., 2020). We formulate the cosine-based learning objective, examine the inference speed improvements, and evaluate alternatives in Appendix B.4.

Even after our candidate selection, the input sequence length to a model can exceed its limit, so we truncate the sequence. To reduce information loss from traditional truncation (e.g., for large DOM elements and long history), we design a strategy that leverages the hierarchical nature of the input to determine which subsection should be truncated. We introduce several improvements to the representation used in prior works by including the full HTML attributes, viewport size, XML Path, and the bounding boxes of candidate elements (implementation details in Appendices B.1 and B.2).

[5] Calculated on the training set, see Appendix B.4.1.

5.2 Modeling Actions

Upon selecting the most promising candidates for a given state s_t, we can combine them with the remaining information in s_t to construct a representation that can be used to predict action strings, which can be parsed and executed (§3.1). To understand which factors matter for predicting actions, we examine 19 zero-shot and finetuned models (using the TRAIN split) with different input modalities: image-only, text-only, and both. We provide implementation details in Appendix B.6 and hyperparameters in Appendix B.7.

Model Categories We categorize action models by the input modality, since the output is always in a structured format (§3.1). We define the following types: (1) **text-only**, which receives instructions, pruned DOM tree, candidate element description and history; (2) **image-to-text**, which receives the screenshot, instructions and past actions directly embedded in the image; (3) multimodal, which receives the screenshot, instructions, pruned DOM tree, candidate description and history directly as text. Additional discussions are found in Appendix B.3.

Text-only models **MindAct** (Deng et al., 2023) is a Flan-T5 (Chung et al., 2022a) model that has been finetuned on Mind2Web. We further fine-tune it on WEBLINX using its original configuration. To quantify the improvements brought by DMR-based representation (§5.1), we directly finetune **Flan-T5** to control for size and architecture with respect to MindAct. We also finetune **LLaMA-2** (Touvron et al., 2023a;b)[6] and a distilled version, **Sheared-LLaMA** (S-LLaMA; Xia et al. 2023).

Proprietary text-only LLMs We report results for GPT-3.5 Turbo (Brown et al., 2020; Peng et al., 2023), in both zero-shot (**3.5T**) and finetuned (**3.5F**) settings. We also include zero-shot results for **GPT-4T** (OpenAI, 2023b).

Image-to-text modeling We explore **Pix2Act** (Shaw et al., 2023) an encoder-decoder (Vaswani et al., 2017) purely finetuned on pixels. It uses a Pix2Struct backbone (Lee et al., 2023), which is pretrained on screenshots using a Vision Transformer encoder (Dosovitskiy et al., 2021) and a text decoder. We follow the behavior cloning approach used by Pix2Act by finetuning the same backbone.

Multimodal models We finetune **Fuyu-8B** (Bavishi et al., 2023), a base model pretrained on browser screenshots by modeling images and text using a unified architecture. We also report zero-shot results for the variant of GPT-4 with vision capabilities (**GPT-4V**; OpenAI 2023a).

Table 5: Aggregated results across major models, sorted by parameter count. Results on TEST_{OOD} except the last column on TEST_{IID}. 👁 indicates models with access to screenshots.

Models	Size	Intent IM	Element IoU	Text F1	**Overall Score** $\textbf{TEST}_{\textbf{OOD}}$	$\textbf{TEST}_{\textbf{IID}}$
Zero-shot						
Llama-2	13B	43.5	4.9	1.4	5.2	5.6
GPT-3.5T	–	42.7	9.0	3.5	8.8	10.3
GPT-4T	–	41.8	11.2	6.9	11.0	12.2
GPT-4V👁	–	42.3	11.4	6.4	10.9	12.9
Finetuned						
Pix2Act👁	1.3B	82.1	9.3	26.6	18.4	23.9
S-LLaMA	2.7B	**84.7**	25.3	**29.2**	27.6	**37.4**
MindAct	3B	80.1	17.7	23.4	21.9	25.7
Flan-T5	3B	81.6	22.1	26.4	25.2	31.1
Fuyu👁	8B	80.9	17.8	24.5	22.2	30.9
Llama-2	13B	83.0	**25.7**	28.7	**27.8**	37.0
GPT-3.5F	–	78.5	21.1	23.8	23.3	30.8

6 Experimental Results

In this section, we report the results of our experiments (§5) for groups defined in Section 4.2. We aggregate the results for 11 models in Table 5. In Section 6.2, we qualitatively assess two major models: GPT-4V and LLaMA-2-13B. See Appendix C for supplementary results and Appendix D for the detailed overview (including the remaining 8 variants).

6.1 Overview of Results

Impact of representation for text-only models In Table 5, we observe that MindAct trails behind Flan-T5 finetuned using DMR-based input representation (§5.1), when comparing the 3B-parameter variants. Although MindAct was finetuned for a related task, it was never exposed to multi-turn dialogue. However, Flan-T5 was never trained on any navigation actions. Thus, DMR-based representation plays an important role in achieving a better performance for the same

[6]We use the variants finetuned on chat.

architecture and model size. Moreover, both LLaMa-based models outperform Flan-T5 and MindAct despite Sheared-LLaMa being smaller than Flan-T5. This could be due to the high quality training of LLaMa models on a large number of instruction-following tasks compared to Flan-T5. However, it is intriguing that Sheared-LLaMa performs equally well compared to LLaMA-2 13B.

Image-to-text vs. multimodal models We further highlight the difference between smaller image-to-text and larger multimodal models by comparing Pix2Act (1.3B parameters) and Fuyu-8B. Overall, Fuyu outperforms Pix2Act, which could be due its ability to receive text as input and greater parameter count. However, it trails behind Pix2Act for intent matching and text prediction.

Comparing multimodal with chat-based models We observe that Fuyu-8B is outperformed by chat-based text-only LLaMA models. This shows that multimodal models finetuned on screenshots are still behind chat-based models optimized for instruction-based finetuning.

Comparison with proprietary models In the zero-shot setting, where models solely rely on the instructions, we observe that proprietary models (GPT-3.5T and GPT-4T) outperform the open-sourced LLaMA-2. However, when finetuned, GPT-3.5F is outperformed by Sheared-LLaMA and LLaMA-2, but the cause is unclear as most hyperparameters are inaccessible for commercial training. Finally, GPT-4V and GPT-4T achieve similar performance, suggesting that existing multimodal models might not be able to effectively use screenshots for predicting actions.

Generalization capabilities When comparing TEST_{OOD} with TEST_{IID} results, we observe a major difference across all finetuned models. This highlights a weakness of finetuned models: although they perform well on familiar websites, they will struggle to generalize to unseen websites. For example, we observe in Table 6 that LLaMa-13B achieves poor results on TEST_{CAT}, indicating that unseen subcategories are more challenging than new websites from the same categories. For instance, if the model learns how to book seats at a restaurant, it can adapt to a different restaurant but will struggle to book a medical appointment.

Table 6: Results on out-of-domain splits for finetuned LLaMA-2-13B. TEST_{CAT} shows the highest difficulty among splits.

Splits	IM	IoU	F1	Overall
TEST_{WEB}	82.7	24.2	28.7	27.0
TEST_{CAT}	81.0	20.7	26.1	24.3
TEST_{GEO}	78.6	22.0	27.7	25.9
TEST_{VIS}	85.3	26.1	23.9	25.0

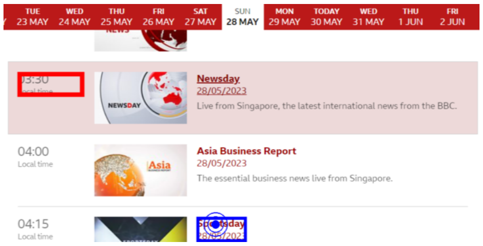
6.2 Qualitative Assessment

To better understand the performance gap separating the strongest zero-shot and finetuned models, we qualitatively examine two models, GPT-4V and LLaMA-2-13B, which respectively represent the two paradigms. Although the gap can be partially attributed to incorrectly predicted intents (see Appendix D), models can still make poor predictions even when the intent is predicted correctly. We focus on this scenario by assessing actions from 3 intents: `click`, `textinput` and `say`; for each, we show two examples in Figure 3. Extended assessments can be found in Appendix C.5.

Assessing `click` In scenarios where models select objects through clicks, we find that GPT-4V chose an incorrect tab (C1), was unaware it has already started a sub-task (C2), and chose a less optimal option (§C.5). Although those scenarios are correctly addressed by the finetuned LLaMA-2, it can still fail by clicking on irrelevant elements (even when GPT-4V selects the correct one).

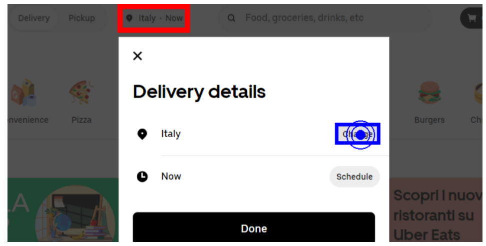
Assessing `textinput` When looking at examples where models are selecting and typing text inside inputs, we observe that GPT-4V tried to write the name of a email recipient instead of the subject title (T1), the username inside a password field (T2), typed a passage already in the target textbox, and skip the title when drafting a post. Although LLaMA succeeded in the first two cases, it may attempt to `click` instead of `textinput` and also omit the title.

Assessing `say` For `say` actions, GPT-4V used a different writing style (S1), whereas LLaMA-2 learned the writing style of the annotators. Additionally, GPT-4V provided unhelpful responses by sharing irrelevant links (S2) and refused to assist the instructor even when it is possible. Even though LLaMA-2 is finetuned, it missed certain follow-up questions (such as asking "Who should receive this?" when asked to write an email).



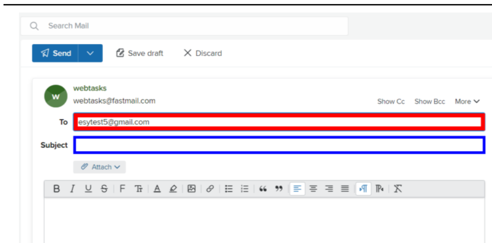
C1: Instructor wants Navigator to open a specific tab on a News site, i.e., "Sportsday on 28 May 2023 at 4.15 AM".

GPT-4V (R) clicks on an incorrect (3:30AM) tab, even though the instructor requested a different time (4:15AM).
LLaMA-WL (B) clicks on the correct 4:15AM tab.



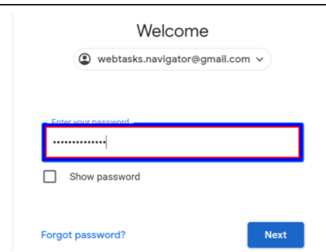
C2: Instructor requests the location on a food delivery site to be set to Las Vegas. The Delivery details window is already open.

GPT-4V (R) attempts to exit the Delivery details page and reopen it, which could potentially lead to a loop.
LLaMA-WL (B) correctly clicks on the *Change* button.



T1: Compose a "Invitation to Collaboration" email.

Reference (B): Invitation to Collaboration
GPT-4V (R): Leon Tales
LLaMA-WL (B): Invitation to Collaboration



T2: Open Google translate and sign in using the following credentials: [email] [password]
Reference (B): *[password]*
GPT-4V (R): *[email]*
LLaMA-WL (B): *[password]*

***S1**: Instructor says "Like & Bookmark this listing."*

Reference: Alright
LLaMA-WL: Alright
GPT-4V: Understood. If you need further assistance, just let me know.

***S2**: Instructor says " Please share the link."*

Reference: Alright.
LLaMA-WL: Okay.
GPT-4V: Here is the link to the discussion: [*incorrect link*]

Figure 3: Comparison of GPT-4V and LLaMA-2-13B (finetuned). Incorrect predictions are in red (R), reference are in blue (B). We show scenarios for `click` (C1,C2), `textinput` (T1,T2) and `say` (S1, S2).

7 Discussion and Conclusion

Through our experiments (Section 5), we find that larger multimodal models can surpass smaller image-only models when finetuned, but they are still behind finetuned text-only models. We also find that employing an DMR-based representation leads to better performance (§6.1). When evaluated on out-of-domain splits, the performance of text-only decoders are very close to smaller variant; nonetheless, zero-shot models are consistently surpassed by their finetuned counterparts. We confirm, through qualitative assessments (§6.2), that even the best zero-shot models can make simple and unjustified errors. Our findings highlight the need to build models that can better generalize to unseen scenarios if we want to build agents that will work in the real world.

In conclusion, we introduced WebLINX, a large-scale expert-built benchmark covering a wide range of demonstrations for conversational web navigation on real-world websites. The framework we built around the benchmark includes the task definition, data representation, and evaluation metrics. We also introduced a dense markup ranker (DMR) to effectively summarize webpages. We evaluated finetuned and zero-shot models with various modalities, and found that chat-based decoder models finetuned on WebLINX achieve the best results, but still struggle to generalize to out-of-domain splits. We believe that multi-turn dialogue can enhance flexibility and sterability of agents for web navigation, leading to their wider adoption.

Acknowledgments

XHL acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference no. 579403]. ZK is supported by the European Research Council (Grant agreement No. 101039303, NG-NLG) and Charles University project SVV 260 698. SR is supported by a Facebook CIFAR AI Chair and NSERC Discovery Grant program. The project is supported partially by the Google-Mila grant. We thank Esycommerce for providing their data annotation services and actively working with us in order to reach a consistent and high quality data collection process. We thank Benno Krojer, Chris Pal, Dilek Hakkani-Tür, Gokhan Tur, Ismail Haritaoglu, Nicolas Chapados, Ondřej Dušek, Peter Shaw, Sai Rajeswar, Vaibhav Adlakha, the UI Assist team at ServiceNow Research, and the McGill NLP group members for helpful discussions.

References

Adept. Adept ACT-1 – ”A machine learning model that can interact with everything on your computer.”. https://www.adept.ai/blog/act-1, 2023. Accessed: 2023/08/31.

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. TopiOCQA: Open-domain Conversational Question Answering with Topic Switching. *Trans. Assoc. Comput. Linguistics*, 10:468–483, 2022. URL https://doi.org/10.1162/tacl_a_00471.

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. HTLM: Hyper-text Pre-training and Prompting of Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. URL https://openreview.net/forum?id=P-pPW1nxf1r.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a Visual Language Model for Few-shot Learning. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html.

Rim Assouel, Tom Marty, Massimo Caccia, Issam Laradji, Alexandre Drouin, Sai Rajeswar, Hector Palacios, Quentin Cappart, David Vazquez, Nicolas Chapados, Maxime Gasse, and Alexandre Lacoste. The unsolved challenges of LLMs in open-ended web tasks: A case study. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL https://openreview.net/forum?id=jt3il4fC5B.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Fuyu-8B: A Multimodal Architecture for AI Agents, October 2023. URL https://www.adept.ai/blog/fuyu-8b/.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-shot Learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Sara Bunian, Kai Li, Chaima Jemmali, Casper Harteveld, Yun Fu, and Magy Seif El-Nasr. VINS: Visual Search for Mobile User Interface Design. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pp. 423:1–423:14, 2021. URL https://doi.org/10.1145/3411764.3445762.

Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A. Plummer. A Dataset for Interactive Vision-language Navigation with Unknown Command Feasibility. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part VIII*, volume 13668 of *Lecture Notes in Computer Science*, pp. 312–328, 2022. URL https://doi.org/10.1007/978-3-031-20074-8_18.

John M. Carroll and Mary Beth Rosson. Usability Engineering. In Heikki Topi and Allen Tucker (eds.), *Computing Handbook, Third Edition: Information Systems and Information Technology*, pp. 32: 1–22. CRC Press, 2014.

Chia-Hui Chang, Mohammed Kayed, Moheb R. Girgis, and Khaled F. Shaalan. A Survey of Web Information Extraction Systems. *IEEE Trans. Knowl. Data Eng.*, 18(10):1411–1428, 2006. URL https://doi.org/10.1109/TKDE.2006.152.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *SIGKDD Explor.*, 19(2):25–35, 2017. URL https://doi.org/10.1145/3166054.3166058.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling Instruction-finetuned Language Models. *CoRR*, abs/2210.11416, 2022a. URL https://doi.org/10.48550/arXiv.2210.11416.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling Instruction-finetuned Language Models. *CoRR*, abs/2210.11416, 2022b. URL https://doi.org/10.48550/arXiv.2210.11416.

Michael Cormer, Richard Mann, Karyn Moffatt, and Robin Cohen. Towards an Improved Vision-based Web Page Segmentation Algorithm. In *2017 14th Conference on Computer and Robot Vision (CRV)*, pp. 345–352, 2017.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. *CoRR*, abs/2307.08691, 2023. URL https://doi.org/10.48550/arXiv.2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and Memory-efficient Exact Attention with IO-awareness. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew W. Senior, Paul A. Tucker, Ke Yang, and Andrew Y. Ng. Large Scale Distributed Deep Networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing*

Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States, pp. 1232–1240, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html.

Xiang Deng, Prashant Shiralkar, Colin Lockard, Binxuan Huang, and Huan Sun. DOM-LM: Learning Generalizable Representations for HTML Documents. *CoRR*, abs/2201.10608, 2022. URL https://arxiv.org/abs/2201.10608.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a Generalist Agent for the Web. *CoRR*, abs/2306.06070, 2023. URL https://doi.org/10.48550/arXiv.2306.06070.

Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit Inference Scaling Laws. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7750–7774, 2023. URL https://proceedings.mlr.press/v202/dettmers23a.html.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. Workarena: How capable are web agents at solving common knowledge work tasks?, 2024.

Hiroki Furuta, Ofir Nachum, Kuang-Huei Lee, Yutaka Matsuo, Shixiang Shane Gu, and Izzeddin Gur. Multimodal Web Navigation with Instruction-finetuned Foundation Models. *CoRR*, abs/2305.11854, 2023. URL https://doi.org/10.48550/arXiv.2305.11854.

Google. The bfloat16 numerical format — Cloud TPU, December 2023. URL https://cloud.google.com/tpu/docs/bfloat16.

Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *Int. J. Comput. Vis.*, 127(4):398–414, 2019. URL https://doi.org/10.1007/s11263-018-1116-0.

Izzeddin Gur, Natasha Jaques, Yingjie Miao, Jongwook Choi, Manoj Tiwari, Honglak Lee, and Aleksandra Faust. Environment Generation for Zero-shot Compositional Reinforcement Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 4157–4169, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/218344619d8fb95d504ccfa11804073f-Abstract.html.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an End-to-end Web Agent with Large Multimodal Models, 2024.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: decoding-enhanced Bert with Disentangled Attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Vojtěch Hudeček and Ondřej Dušek. Are LLMs All You Need for Task-oriented Dialogue? *CoRR*, abs/2304.06556, 2023. URL https://doi.org/10.48550/arXiv.2304.06556.

Peter C. Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Adam Santoro, and Timothy P. Lillicrap. A data-driven approach for learning to control computers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9466–9482, 2022. URL https://proceedings.mlr.press/v162/humphreys22a.html.

HyperWrite. HyperWrite AI Personal Assistant – ”The first publicly available AI agent that can operate a browser like a human.”. https://www.hyperwriteai.com/personal-assistant, 2023. Accessed: 2023/08/31.

Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *CoRR*, abs/2310.06825, 2023. URL https://doi.org/10.48550/arXiv.2310.06825.

Katharina Kann, Abteen Ebrahimi, Joewie J. Koh, Shiran Dudy, and Alessandro Roncone. Open-domain Dialogue Generation: What We Can Do, Cannot Do, And Should Do Next. In *Proceedings of the 4th Workshop on NLP for Conversational AI, ConvAI at ACL 2022, Dublin, Ireland, May 27, 2022*, pp. 148–165, 2022. URL https://doi.org/10.18653/v1/2022.nlp4convai-1.13.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-domain Question Answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 6769–6781, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A Diagram is Worth a Dozen Images. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pp. 235–251, 2016. URL https://doi.org/10.1007/978-3-319-46493-0_15.

Johannes Kiesel, Florian Kneist, Lars Meyer, Kristof Komlossy, Benno Stein, and Martin Potthast. Web Page Segmentation Revisited: Evaluation Framework and Dataset. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pp. 3047–3054, 2020. URL https://doi.org/10.1145/3340531.3412782.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. Language Models can Solve Computer Tasks. *CoRR*, abs/2303.17491, 2023. URL https://doi.org/10.48550/arXiv.2303.17491.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In Jason Flinn, Margo I. Seltzer, Peter Druschel, Antoine Kaufmann, and Jonathan Mace (eds.), *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pp. 611–626, 2023. URL https://doi.org/10.1145/3600006.3613165.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. OBELICS: An Open Web-scale Filtered Dataset of Interleaved Image-text Documents, 2023.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18893–18912, 2023. URL https://proceedings.mlr.press/v202/lee23g.html.

Junlong Li, Yiheng Xu, Lei Cui, and Furu Wei. MarkupLM: Pre-training of Text and Markup Language for Visually Rich Document Understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 6078–6087, 2022. URL https://doi.org/10.18653/v1/2022.acl-long.420.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. Mapping Natural Language Instructions to Mobile UI Action Sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 8198–8210, 2020. URL https://doi.org/10.18653/v1/2020.acl-main.729.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards General Text Embeddings with Multi-stage Contrastive Learning. *CoRR*, abs/2308.03281, 2023a. URL https://doi.org/10.48550/arXiv.2308.03281.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards General Text Embeddings with Multi-stage Contrastive Learning. *CoRR*, abs/2308.03281, 2023b. URL https://doi.org/10.48550/arXiv.2308.03281.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement Learning on Web Interfaces using Workflow-guided Exploration. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=ryTp3f-0-.

Wei Liu, Xiaofeng Meng, and Weiyi Meng. ViDE: A Vision-based Approach for Deep Web Data Extraction. *IEEE Trans. Knowl. Data Eng.*, 22(3):447–460, 2010. URL https://doi.org/10.1109/TKDE.2009.109.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 3195–3204, 2019. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Marino_OK-VQA_A_Visual_Question_Answering_Benchmark_Requiring_External_Knowledge_CVPR_2019_paper.html.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: Massive Text Embedding Benchmark. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pp. 2006–2029, 2023. URL https://doi.org/10.18653/v1/2023.eacl-main.148.

Multi-On. Multi-on – "The world's first Personal AI Agent & Life Copilot". https://multion.ai, 2023. Accessed: 2023/08/31.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021. URL https://arxiv.org/abs/2112.09332.

Rodrigo Frassetto Nogueira and Kyunghyun Cho. End-to-end Goal-driven Web Navigation. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1903–1911, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/1579779b98ce9edb98dd85606f2c119d-Abstract.html.

OpenAI. Introducing Triton: Open-source GPU programming for neural networks, July 2021. URL https://openai.com/research/triton.

OpenAI. Introducing ChatGPT, November 2022. URL https://openai.com/blog/chatgpt.

OpenAI. GPT-4V(ision) System Card. *Technical Report*, 2023a. URL https://cdn.openai.com/papers/GPTV_System_Card.pdf.

OpenAI. GPT-4 Technical Report. *CoRR*, abs/2303.08774, 2023b. URL https://doi.org/10.48550/arXiv.2303.08774.

OpenAI. New models and developer products announced at DevDay, November 2023c. URL https://openai.com/blog/new-models-and-developer-products-announced-at-devday.

OpenAI. ChatGPT Plugins. https://openai.com/blog/chatgpt-plugins, 2023d. Accessed: 2023/09/03.

Panupong Pasupat, Tian-Shun Jiang, Evan Zheran Liu, Kelvin Guu, and Percy Liang. Mapping natural language commands to web elements. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 4970–4976, 2018. URL https://aclanthology.org/D18-1540/.

Andrew Peng, Michael Wu, Logan Kilpatrick, and Steven Heidel. GPT-3.5 Turbo fine-tuning and API updates, August 2023. URL https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates.

Yury Pinsky. Bard can now connect to your google apps and services, Sep 2023. URL https://blog.google/products/bard/google-bard-new-features-update-sept-2023/.

Maja Popovic. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT at EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pp. 392–395, 2015. URL https://doi.org/10.18653/v1/w15-3049.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards Scalable Multi-domain Conversational Agents: The Schema-guided Dialogue Dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8689–8696, 2020. URL https://doi.org/10.1609/aaai.v34i05.6394.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy P. Lillicrap. Android in the Wild: A Large-scale Dataset for Android Device Control. *CoRR*, abs/2307.10088, 2023. URL https://doi.org/10.48550/arXiv.2307.10088.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. URL https://aclanthology.org/D19-1410.

Peter Shaw, Mandar Joshi, James Cohan, Jonathan Berant, Panupong Pasupat, Hexiang Hu, Urvashi Khandelwal, Kenton Lee, and Kristina Toutanova. From Pixels to UI Actions: Learning to Follow Instructions via Graphical User Interfaces. *CoRR*, abs/2306.00245, 2023. URL https://doi.org/10.48550/arXiv.2306.00245.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of Bits: An Open-domain Platform for Web-based Agents. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3135–3144, 2017. URL http://proceedings.mlr.press/v70/shi17a.html.

Raphael Shu, Elman Mansimov, Tamer Alkhouli, Nikolaos Pappas, Salvatore Romeo, Arshit Gupta, Saab Mansour, Yi Zhang, and Dan Roth. Dialog2API: Task-oriented Dialogue with API Description and Example Programs. *CoRR*, abs/2212.09946, 2022. URL https://doi.org/10.48550/arXiv.2212.09946.

Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. META-GUI: Towards Multi-modal Conversational Agents on Mobile GUI. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 6699–6712, 2022. URL https://doi.org/10.18653/v1/2022.emnlp-main.449.

Together. Redpajama: an open dataset for training large language models, october 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971, 2023a. URL https://doi.org/10.48550/arXiv.2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-tuned Chat Models. *CoRR*, abs/2307.09288, 2023b. URL https://doi.org/10.48550/arXiv.2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. WebFormer: The Web-page Transformer for Structure Information Extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 3124–3133, 2022. URL https://doi.org/10.1145/3485447.3512032.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep Self-attention Distillation for Task-agnostic Compression of Pre-trained Transformers. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language Models are Zero-shot Learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

Jason Wu, Xiaoyi Zhang, Jeffrey Nichols, and Jeffrey P. Bigham. Screen Parsing: Towards Reverse Engineering of UI Models from Screenshots. In *UIST '21: The 34th Annual ACM Symposium on User Interface Software and Technology, Virtual Event, USA, October 10-14, 2021*, pp. 470–483, 2021. URL https://doi.org/10.1145/3472749.3474763.

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P. Bigham. WebUI: A Dataset for Enhancing Visual UI Understanding with Web Semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023, Hamburg, Germany, April 23-28, 2023*, pp. 286:1–286:14, 2023. URL https://doi.org/10.1145/3544548.3581158.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning. *CoRR*, abs/2310.06694, 2023. URL https://doi.org/10.48550/arXiv.2310.06694.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. C-Pack: Packaged Resources To Advance General Chinese Embedding. *CoRR*, abs/2309.07597, 2023a. URL https://doi.org/10.48550/arXiv.2309.07597.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. C-Pack: Packaged Resources To Advance General Chinese Embedding. *CoRR*, abs/2309.07597, 2023b. URL https://doi.org/10.48550/arXiv.2309.07597.

Nancy Xu, Sam Masling, Michael Du, Giovanni Campagna, Larry Heck, James A. Landay, and Monica Lam. Grounding Open-domain Instructions to Automate Web Support Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 1022–1032, 2021. URL https://doi.org/10.18653/v1/2021.naacl-main.80.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-world Web Interaction with Grounded Language Agents. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2207.01206.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. DIALOGPT : Large-scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pp. 270–278, 2020a. URL https://doi.org/10.18653/v1/2020.acl-demos.30.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10): 2011–2027, 2020b.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. *Proc. VLDB Endow.*, 16(12):3848–3860, 2023. URL https://www.vldb.org/pvldb/vol16/p3848-huang.pdf.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a Generalist Web Agent, if Grounded. *CoRR*, abs/2401.01614, 2024. URL https://doi.org/10.48550/arXiv.2401.01614.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A Realistic Web Environment for Building Autonomous Agents. *CoRR*, abs/2307.13854, 2023. URL https://doi.org/10.48550/arXiv.2307.13854.

Yichao Zhou, Ying Sheng, Nguyen Vo, Nick Edmonds, and Sandeep Tata. Simplified DOM Trees for Transferable Attribute Extraction from the Web. *CoRR*, abs/2101.02415, 2021. URL https://arxiv.org/abs/2101.02415.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing Vision-language Understanding with Advanced Large Language Models. *CoRR*, abs/2304.10592, 2023. URL https://doi.org/10.48550/arXiv.2304.10592.

CONTENTS

APPENDIX

IMPACT STATEMENT

Web navigation agents have the potential to become a powerful technology with large societal impacts. Therefore, multiple aspects need to be taken into consideration when conducting further research in this area:

Automating vs. Elevating Users A major risk of automating web navigation is the automation of work traditionally performed by knowledge workers; deploying highly capable models could lead to job losses. However, one major difference between autonomous navigation and our framework is that we require the inclusion of a human instructor to provide the real-time instructions needed to complete the task. Thus, conversational web navigation's ultimate purpose is not to automate the human completely, but automate difficult, repetitive, and error-prone steps so that the user can focus on reliably solving high-level problems.

Malicious Usage and Mitigation As web navigation models become increasingly sophisticated, there are risks that they will be used for malicious purposes at scale. These models can automate harmful activities, e.g., for creating spam messages and impersonating individuals for fraudulent purposes. While these activities can already be partially automated using open-source tools,[7] web navigation agents could make automation easier and more robust. However, malicious actors can build such models in private using existing commercial services, independent of on-going research on agents. On the other hand, by making our models and data accessible to researchers, our work can be used to research ways to mitigate the risk of malicious usage; for instance, by incorporating our models as part of red teaming procedures. The resulting research can be used to build systems that are robust against malicious agents.

Unintended Actions Navigation agents can also cause harm if they misinterpret instructions and perform unintended actions; for instance, booking the wrong flight could result in significant financial loss. For this reason, we assert that conversational web navigation models should be used under human supervision (where multi-turn dialogue cannot be disabled), and that it should only be deployed after exhaustive testing with proper safeguards. Our models should not be deployed and should only be used for research.

Data Collection We have collected our benchmark using expert annotators, who were properly trained, familiarized with the task and the purpose of the project, and paid fair wage relative to their country of employment. The websites in our dataset are publicly accessible and safe. Any account appearing in the dataset was specifically created for the data collection; there are no references to their identity to preserve their privacy.

A DATASET DETAILS

A.1 SUPPLEMENTARY STATISTICS

In Section 3, we introduce WEBLINX. In this section, we provide supplementary statistics for readers wishing to gain a deeper understanding of the dataset.

In Table 8, we report demo and turn statistics by intent. We observe that `say`, `click` and `load` are heavily represented across demos. However, the latter happens less often than other intents. This is because the user loads new links only when they move to a new website, and many tasks can be accomplished within the same page (such as booking a flight). Therefore, there is no need to load new pages as frequently as other intents. Additionally, `hover` is less represented due to the removal of unnecessary hovering, which can be accidentally recorded when moving the cursor across non-target elements with callbacks.

In Table 9, we present the number of demos for each split and mean number of turns. Although most demos are in the range of 40-50 turns, the number of demos in the TEST_{VIS} split is substantially lower. This can be attributed to the lack of follow-up based on what is happening on the screen.

[7] For example, Selenium: https://www.selenium.dev/

For example, an instructor with vision can request the navigator to apply some specific filters (e.g., by saying ”Please apply the filter for Japan Airlines under the Airlines filter option”), whereas an instructor without vision would not have this request unless they are using a screen-reader.

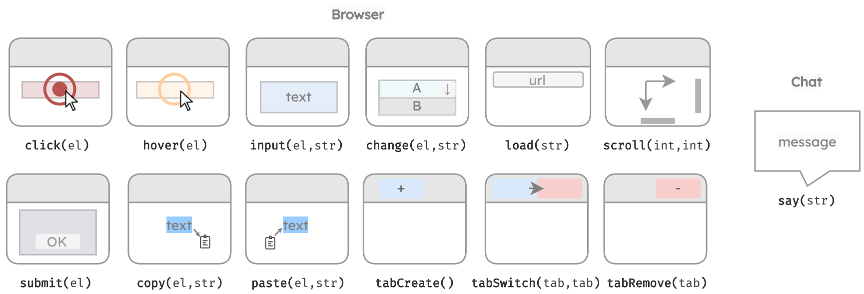


Figure 4: Overview of the actions in our benchmark, including 10 browser actions and 1 chat action. An argument of an action can be a string (`str`), an integer (`int`), an element (`el`), or a browser tab id (`tab`). The intents are described in Table 7.

Table 7: Complete list of WEBLINX action space.

Action	**Description**
`say(speaker, text)`	talking to instructor or navigator
`click(element)`	click on an element
`click(x,y)`	click on the coordinates mapping to an element
`hover(element)`	hover over an element
`hover(x,y)`	hover over the coordinates mapping to an element
`textinput(element, value)`	type text into the element
`change(element, text)`	change the value of the element to another option
`load(url)`	load the URL of a new webpage
`submit(element)`	submit the form
`scroll(x,y)`	scroll to the coordinates
`copy(element, text)`	copy the text from the element
`paste(element, text)`	paste the text into the element
`tabCreate()`	create a new tab
`tabRemove(tabId)`	remove the tab
`tabSwitch(tabIdFrom, tabIdTo)`	switch between tabs

In Table 10, we highlight the usage frequency of AI tools, which are listed in Table 13. For certain tasks, such as summarizing news articles, it is much more convenient to use AI tools. Since we focus on actions executed, models can learn general actions when dealing with AI tools, even when the tools themselves changes.

A.2 CATEGORIES AND SUBCATEGORIES

In Section 3, discuss the use of categories to classify demonstrations. We have in total 8 categories, each with their own subcategories, which add up to a total of 50 (§12); we assign one category and subcategory to Each of the 155 URL sub-domain associated with a demo turn (§13). Since a demo may leverage multiple websites (e.g. composing and information lookup), a demo will have one or more subcategory. We give the full list of categories, subcategories, and the number of demonstrations associated with each in Table 12.

In Table 11, we show the breakdown of subcategories for the TEST$_{\text{CAT}}$ split (designed to test generalization to new subcategories). We note that the subcategories were automatically chosen to be the ones with the fewer occurrences across demos, allowing to have a reasonable split size.

Table 8: Turn-level stats by intent.

Intent	# Demos	μ turns	σ turns	Total
say	2337	16.82	5.62	39305
click	2333	14.52	10.16	33865
load	2324	1.59	1.07	3702
copy	1587	4.08	3.05	6477
textInput	1465	3.28	3.06	4799
paste	1130	1.89	1.95	2141
scroll	1046	3.82	3.00	3999
tabswitch	800	3.28	3.65	2621
tabcreate	712	1.71	1.12	1220
submit	645	1.40	1.11	904
hover	361	1.55	1.11	560
tabremove	309	1.94	1.17	599
change	165	1.95	1.34	322

Table 9: Turn-level stats by split. Active turns are used for either finetuning or evaluation. Total includes turns used in history.

Split	# Demos	μ turns	σ turns	Active	Total
TRAIN	969	44.93	17.37	24418	43538
VALID	100	40.76	14.51	1717	4076
TEST_{IID}	100	43.18	16.08	1846	4318
TEST_{CAT}	223	45.30	25.43	4979	10102
TEST_{WEB}	211	40.47	18.17	4184	8540
TEST_{VIS}	444	36.05	20.09	7725	16006
TEST_{GEO}	290	48.05	18.66	6141	13934

A.3 INPUT PROCESSING DETAILS

In Section 3.1, we introduce the components of a state s_t. More formally, we define the input of a model m to be $\mathcal{P}_m(s_t, a_{1:t-1})$, consisting of a processing function $\mathcal{P}_m$ that receives s_t and $a_{1:t-1}$ and returns a representation that can serve as an input to a model. We provide details of our method below.

Adapting $\mathcal{P}$ per model For each model m, we tailor the function $\mathcal{P}_m$ to accommodate for differences in methodology. For image-to-text models, we sequentially render v_t, u_r, a_r as header text of the screenshot i_t (viewport v_t is included so models can locate bounding boxes of c_t). For text-only models, we provide d_t, v_t, u_r, c_t, a_r, which are formatted with prompt p_m. In multimodal settings, we include i_t in addition to the formatted prompt. Templates and samples can be found in Appendices B.5 and B.8.

Candidate selection Following Deng et al. (2023), we employ a separate candidate selection stage in order to reduce the number of the input elements to interact with. In the candidate selection stage, a ranking model selects a subset of k relevant elements from the DOM tree, which is then presented to the model in a multi-choice setup; in Section 5.1, we describe a novel approach towards candidate selection designed for real-time use cases. When the candidate is selected, c_t is returned to be used in $\mathcal{P}$. Each candidate contains a tag, XPath, bounding box, attributes and children tags, which are delimited with square brackets (e.g., `[[tag]]...[[xpath]]...`). Examples of candidates used inside prompts can be found in Appendix B.8.

Restricting history for input To accommodate the maximum input length a model can receive, we can restrict $a_{1:t-1}$ and $u_{1:t-1}$ to select a subset window of w. For actions, we select the last w instances by either the instructor or navigator. For instructor utterances, we only select the first and last $w - 1$ instances, allowing us to keep track of the initial request while focusing on the latest updates to the instruction. For simplicity, we denote the restricted set of actions as a_r and utterances

Table 10: Turn-level stats by use of AI tools (e.g., ChatGPT)

Uses AI	# Demos	μ turns	σ turns	Total
✗	2057	42.50	19.5	87414
✓	280	46.79	16.9	13100

Table 11: List of subcategories based on splits.

TEST$_{\text{CAT}}$	Spreadsheet, Handmade, Reviews, Computer Vision, Chatbot, Transport, Presentation, Furniture, Professional Network, Books, Tasks, Automatic Translation, Question Answering, Encyclopedia, Recipe, Geography
Others	Stay, Stays, Transport, Scientific Articles, Online Shopping, Tasks, Blog, Discussion Platform, Recipe, Spreadsheet, Email, Research Directory, Music Sharing, Chatbot, Presentation, Grocery, Delivery, Image Sharing, Automatic Translation, Video Sharing, Encyclopedia, News Articles, Forum, Entertainment, Magazine, Medical, Furniture, Educational, Kanban, Social Network, Image Generation, Question Answering, Media, Note taking, Agency, Government, Social Event, Cooking, Instant Messaging, Finance, Books, Clothing, Restaurant, Calendar, Writing Assistant
Difference	Handmade, Reviews, Computer Vision, Professional Network, Geography

as u_r. Similar to Deng et al. (2023), we choose $w = 5$, allowing the model to attend recent actions without going over context limits.

A.4 Output Processing Details

Although the model is finetuned to generate a string in the format described in Table 3, the raw output is not consistently suitable for direct execution, and may contain unnecessary artifacts. We process the output by using Regex pattern matching to find the first suitable `intent` call, then parse the α into key/value pairs, which can be compared with the ground truth actions.

Mapping coordinates to elements Vision models without access to candidate elements will instead be instructed and finetuned to choose an element by specifying its (x, y) coordinates. If there are overlapping elements at a coordinate, we choose the element with the smallest area at the given (x, y) coordinates (which should be the target of the interaction due to the properties of the CSS box model). Technically, the click targets the element with the highest z-index (the depth axis in HTML), but since we do not have access to CSS properties of the object, we rely on the default render order.

Segmenting URLs for `load` actions We use `urllib`[8] to first segment the URL into a network location (`netloc`) and the remaining hierarchical path (`path`). To normalize the `netloc`, we remove the leading `www` from it. Since a `path` is separated by a forward slash (`/`), we use this character to separate each segment in the path. The final result is a list of tokens, each representing a part of the initial URL.

A.5 Data Collection Details

In Table 3, we provide an overview of the data collection process to build the dataset component of WEBLINX. The overview of the process is outlined in Figure 5. In this section, we dive into the technical and supplementary details of the process.

Website Selection We assembled the list of recommended websites to be used as starting points, but the annotators were allowed to visit any websites they deemed appropriate for the task (full list available in Section A.7). The annotators were given the time to become acquainted with the specific websites before recording the demonstrations. We encouraged the annotators to record both shorter, single-task demonstrations, and more complex demonstrations consisting of multiple sub-tasks. The demonstration ends once the instructor notifies the navigator that they wish to terminate the demonstration.

[8]https://docs.python.org/3/library/urllib.parse.html

Table 12: Number of demos each subcategory appears in for each split. Note that a demo might have multiple subcategories when using more than one website (for example, Information Lookup and Composing). In the last column, we also include the number of URLs associated with each subcategory; they correspond to the websites in Table 13.

Category	Subcategory	Total	Train	Valid	ID	Vis	Geo	Cat	Web	# URLs
AI Tools	Auto. Translation	53	0	0	0	10	0	43	0	4
	Chatbot	408	178	19	21	82	42	31	35	3
	Computer Vision	13	0	0	0	0	0	13	0	1
	Image Generation	59	33	7	3	5	0	0	11	4
	Writing Assistant	70	44	3	2	11	0	0	10	5
Booking	Medical	34	0	0	0	9	25	0	0	3
	Restaurant	77	28	6	5	14	24	0	0	6
	Social Event	14	0	0	0	0	14	0	0	3
	Stay	64	44	0	0	5	15	0	0	7
	Stays	37	24	0	0	11	0	0	2	3
	Transport	757	314	27	31	252	36	61	36	8
Composing	Blog	62	34	2	3	15	0	0	8	4
	Email	135	86	10	17	16	0	0	6	6
	Note taking	47	31	0	5	11	0	0	0	4
	Recipe	20	0	0	0	3	0	17	0	1
	Tasks	31	0	0	0	10	0	21	0	2
Information Lookup	Agency	46	29	2	3	0	0	0	12	3
	Educational	56	28	3	2	8	0	0	15	2
	Encyclopedia	97	56	8	7	11	0	1	14	4
	Entertainment	36	13	0	0	10	0	0	13	2
	Forum	37	12	4	1	9	0	0	11	2
	Geography	13	0	0	0	0	0	13	0	1
	Government	36	0	0	0	9	27	0	0	2
	Media	60	23	2	3	10	0	0	22	2
	Research Directory	10	0	0	0	10	0	0	0	2
Productivity	Calendar	50	17	3	2	11	3	0	14	2
	Finance	59	21	0	0	10	28	0	0	4
	Kanban	50	20	2	3	16	0	0	9	3
	Presentation	32	0	0	0	6	0	26	0	1
	Spreadsheet	27	0	0	0	10	0	17	0	2
Shopping	Clothing	93	18	6	4	8	57	0	0	6
	Delivery	91	67	4	6	14	0	0	0	7
	Furniture	6	0	0	0	5	0	1	0	1
	Grocery	38	0	0	0	8	30	0	0	2
	Handmade	15	0	0	0	0	0	15	0	1
	Online Shopping	87	51	3	2	31	0	0	0	7
Social Interaction	Discussion Platform	32	18	4	1	9	0	0	0	3
	Image Sharing	60	30	6	9	0	0	0	15	4
	Instant Messaging	32	11	0	0	11	0	0	10	2
	Music Sharing	36	14	0	0	9	0	0	13	2
	Professional Network	14	0	0	0	0	0	14	0	1
	Question Answering	20	0	0	0	5	0	15	0	1
	Social Network	62	28	4	2	13	14	0	1	4
	Video Sharing	20	10	0	0	1	0	0	9	1
Summarizing	Books	25	0	0	0	10	0	15	0	2
	Cooking	40	13	0	0	11	16	0	0	2
	Magazine	49	24	0	1	11	13	0	0	4
	News Articles	124	75	11	11	15	12	0	0	5
	Reviews	13	0	0	0	0	0	13	0	1
	Scientific Articles	35	10	4	2	10	0	0	9	2

Recording Demonstrations To capture the states and actions during the demonstration, we implemented a custom Chrome browser extension. For each action in the browser, the extension captured the screenshot of the page, the DOM tree of the page, and bounding boxes of the elements in the

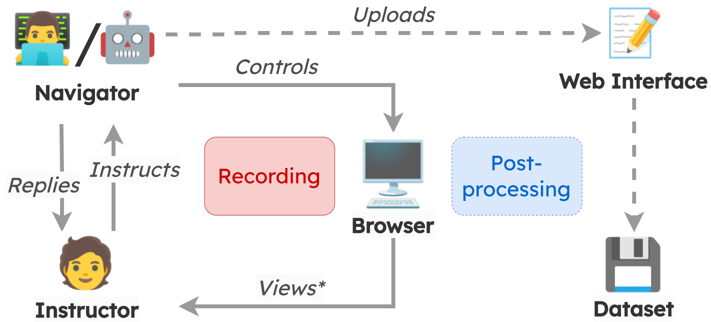


Figure 5: The data collection process. We record interactions between an instructor and a human navigator, including chat and browser actions. *Instructor can see the screen except in TEST_{VIS} split.

viewport. The user actions were captured using web event handlers[9], and Chrome `tabCapture` API[10] was used to save the state of the page for each action in the background. For screen recording, screen sharing, and chat interface, the annotators used Zoom[11], a free video meeting software. We combined the chat with the browser states and actions in the postprocessing stage. Finally, the annotators validate demonstrations to ensure there are no unnecessary or incorrectly ordered actions, and that there are no typographic errors.

Curating Demonstrations The annotators uploaded the recorded demonstration into our custom web interface to perfom basic quality checks. Using the review mode, the annotators then removed unnecessary actions (such as hovering over elements not necessary for completing the task), corrected the order of actions (which was occasionally incorrect due to asynchronous processing), and fixed typographical errors. We also improved the alignment between screenshots and actions by re-aligning the screenshots based on their similarity to the respective video frames.[12]

Annotator Pay We paid US$7.5 per hour for the demonstration recording and US$5 per hour for overhead (preparation, upload, and quality review), leading to an average US$2.58 per demonstration. The rate is substantially higher than the minimum wage in the region where the data is collected, but also includes other overhead fees.

A.6 Actions and Intents

The action a_t has a structure `intent`$(\alpha_1, \ldots, \alpha_m)$, where our core intents are: `click`, `load` (new page via URL), `say` (navigator's utterance), `submit` (e.g., a form), `textinput` (e.g., typing text in the search bar); we show examples of these actions in Figures 1 and 3. The set of arguments α will be different from each action. Commonly used arguments are the unique ID of an element in d_t and the *text* argument for `say` or `textinput`. To complement the intents described in Table 3, we show a diagram of possible arguments for each intent is provided in Figure 4, with the full list shown in Table 7.

Evaluating intents Among the 13 recorded intent types, we focus on evaluating 5 types: `click`, `load`, `say`, `submit`, `textinput`. We also use `change` and `scroll` as prediction targets during finetuning as they are necessary to complete a demonstration. However, we do not evaluate them as `change` does not appear in every split (see Table 8) and `scroll` cannot be reliably evaluated. The other intents (`copy`, `paste`, `tabswitch`, `tabcreate`, `hover`, `tabremove`) are included in the history and the associated states are available alongside active intents; `copy`, `paste`, and `hover` do not affect the state of the website, whereas the tab actions are not mandatory to navigate a website, as `load` is sufficient to go to any website.

[9] developer.mozilla.org/en-US/docs/Web/Events

[10] developer.chrome.com/docs/extensions/reference/tabCapture

[11] zoom.us

[12] The re-alignment was necessary since the Chrome API allows to capture only 1 screenshot per 500 ms which sometimes caused delays in screenshot capture.

A.7 WEBSITES OVERVIEW

Table 13 shows all entrypoints (website where a demo starts). We choose popular and also lesser known sites to achieve categorical and geographic diversity. The websites are either specifically chosen by the authors or the annotators, who collaboratively ensured they are appropriate for our tasks – consequently, we do not include unsafe websites. In the case of social interactions, we choose websites with terms of use prohibiting offensive content. For instance, https://facebook.com states that "We remove content that could contribute to a risk of harm to the physical security of persons. Content that threatens people has the potential to intimidate, exclude or silence others and isn't allowed on Facebook."[13].

Table 13: Website overview

Name	Category	Subcategory	Geography	URL
Airbnb	Booking	Stays	International	https://www.airbnb.com
Airtable	Productivity	Spreadsheet	International	https://airtable.com
Aldi (Australia)	Shopping	Grocery	Australia	https://www.aldi.com.au/en/
Aliexpress	Shopping	Online Shopping	International	https://www.aliexpress.com/
AllenAI's CV Explore	AI Tools	Computer Vision	USA	https://vision-explorer.allenai.org/
Amazon	Shopping	Online Shopping	International	https://www.amazon.com
Asana	Productivity	Kanban	International	https://asana.com/
ASOS	Shopping	Clothing	International	https://www.asos.com/men/
BBC News	Summarizing	News Articles	International	https://www.bbc.com/
Bing Image Creator	AI Tools	Image Generation	International	https://www.bing.com/create
Bing Translator	AI Tools	Auto. Translation	International	https://www.bing.com/translator
Blogger	Composing	Blog	International	https://www.blogger.com/
Booking.com	Booking	Stays	International	https://www.booking.com
booknbook	Booking	Restaurant	International	https://www.booknbook.com/
Brandmark	AI Tools	Image Generation	International	https://brandmark.io/
Britannica	Info. Lookup	Encyclopedia	International	https://www.britannica.com/
Calculator.net Investment	Productivity	Finance	International	https://www.calculator.net/investment-calculator.html
ChatGPT	AI Tools	Chatbot	International	https://openai.com/
cheaptickets	Booking	Transport	International	https://www.cheaptickets.com/
CIA World Factbook	Info. Lookup	Agency	USA	https://www.cia.gov/the-world-factbook/
CNN	Summarizing	News Articles	International	https://edition.cnn.com/
Copy AI	AI Tools	Writing Assistant	International	https://www.copy.ai/
DeepL	AI Tools	Auto. Translation	International	https://www.deepl.com
delivery	Shopping	Delivery	USA	https://www.delivery.com/
Dictionary	Info. Lookup	Encyclopedia	International	https://www.dictionary.com/
Discord	Social Interaction	Instant Messaging	International	https://discord.com
Discourse	Social Interaction	Discussion Platf.	International	https://try.discourse.org/
Doordash	Shopping	Delivery	International	https://www.doordash.com/
ebay	Shopping	Online Shopping	International	https://www.ebay.com/
Encyclopedia.com	Info. Lookup	Encyclopedia	International	https://www.encyclopedia.com/
Etsy	Shopping	Handmade	International	https://www.etsy.com/in-en
European Commission	Info. Lookup	Government	Europe	https://europa.eu/
Eventbrite	Booking	Social Event	International	https://www.eventbrite.com
Eventbrite (AU)	Booking	Social Event	Australia	https://www.eventbrite.com.au/
expedia	Booking	Stay	International	https://www.expedia.com/
Facebook	Social Interaction	Social Network	International	https://www.facebook.com/login/
Fandom	Info. Lookup	Entertainment	International	https://www.fandom.com/
Fastmail	Composing	Email	International	https://fastmail.com/
Flickr	Social Interaction	Image Sharing	International	https://www.flickr.com/
Frontiers	Summarizing	Scientific Articles	International	https://www.frontiersin.org/journals/
Genius	Social Interaction	Music Sharing	International	https://genius.com
Gmail	Composing	Email	International	https://mail.google.com/
GMX Email	Composing	Email	International	https://www.gmx.com/
Google Bard	AI Tools	Chatbot	International	https://bard.google.com/
Google Calendar	Productivity	Calendar	International	https://calendar.google.com/calendar/
Google Docs	Composing	Note taking	International	https://docs.google.com/document
Google Flights	Booking	Transport	International	https://www.google.com/travel/flights
Google Keep	Composing	Tasks	International	https://keep.google.com/
Google Scholar	Info. Lookup	Research Directory	International	https://scholar.google.com/
Google Sheets	Productivity	Spreadsheet	International	https://docs.google.com/spreadsheets
Google Slides	Productivity	Presentation	International	https://docs.google.com/presentation

Continued on next page

[13] https://transparency.fb.com/policies/community-standards/

Table 13: Website overview

Name	Category	Subcategory	Geography	URL
Google Translate	AI Tools	Auto. Translation	International	https://translate.google.com
Gov. of Canada Budget Planner	Productivity	Finance	Canada	https://itools-ioutils.fcac-acfc.gc.ca/BP-PB/budget-planner-tool
Grammarly (Paraphrasing)	AI Tools	Writing Assistant	International	https://www.grammarly.com/paraphrasing-tool
grubhub	Shopping	Delivery	International	https://www.grubhub.com/
Gutenberg	Summarizing	Books	International	https://www.gutenberg.org/
Hacker News	Social Interaction	Discussion Platf.	USA	https://news.ycombinator.com/
Hostelworld	Booking	Stays	International	https://www.hostelworld.com/
hotels	Booking	Stay	International	https://in.hotels.com/
howstuffworks	Info. Lookup	Educational	International	https://www.howstuffworks.com/
Ikea	Shopping	Furniture	International	https://www.ikea.com/
IMDB	Info. Lookup	Entertainment	International	https://www.imdb.com/
Imgur	Social Interaction	Image Sharing	International	https://imgur.com/
Independent.ie (Ireland)	Summarizing	News Articles	Ireland	https://www.independent.ie/
Instacart	Shopping	Delivery	North America	https://www.instacart.com/
Instagram	Social Interaction	Image Sharing	International	https://www.instagram.com/
investopedia	Info. Lookup	Media	International	https://www.investopedia.com/
Jack's 50 top food bloggers	Summarizing	Cooking	International	https://jacksfoodblog.com/2020/04/26/50-top-food-bloggers-of-2020-the-best-recipe-sites-ranked/
jamesonlinebookclub	Summarizing	Reviews	International	https://jamesonlinebookclub.com/
kayak	Booking	Stay	International	https://www.kayak.co.in/
Khan Academy	Info. Lookup	Educational	USA	https://www.khanacademy.org/
Koo	Social Interaction	Social Network	India	https://www.kooapp.com/feed
LinkedIn	Social Interaction	Prof. Network	International	https://www.linkedin.com/
Loblaws (Canada)	Shopping	Grocery	Canada	https://www.loblaws.ca/
Luko.eu	Booking	Medical	Europe	https://de.luko.eu/en/advice/guide/best-rated-tierartz-veterinarians-by-states/
Macy's	Shopping	Clothing	USA	https://www.macys.com/
Marie Claire	Summarizing	Magazine	International	https://www.marieclaire.com/
MarketWatch	Productivity	Finance	USA	https://www.marketwatch.com/
Medium	Composing	Blog	International	https://medium.com/
Meetup (Glasgow, Scotland)	Booking	Social Event	Scotland	https://www.meetup.com/find/?eventType=inPerson&source=EVENTS&location=gb--v2--Glasgow
momondo	Booking	Transport	International	https://www.momondo.in/
MyFitnessPal	Composing	Recipe	International	https://www.myfitnesspal.com/recipe/calculator
Myntra	Shopping	Clothing	India	https://www.myntra.com/
NASA	Info. Lookup	Agency	USA	https://www.nasa.gov/
National Geographic	Summarizing	Magazine	International	https://www.nationalgeographic.com/magazine
New Yorker	Summarizing	Magazine	USA	https://www.newyorker.com/
New Zealand Government	Info. Lookup	Government	New Zealand	https://www.govt.nz/
Nextdoor	Social Interaction	Discussion Platf.	International	https://nextdoor.com/
NHS - Find a dentist	Booking	Medical	UK	https://www.nhs.uk/service-search/find-a-dentist
Nightcafe	AI Tools	Image Generation	International	https://creator.nightcafe.studio/
nirvanahq	Composing	Tasks	International	https://www.nirvanahq.com
Notion	Composing	Note taking	International	https://www.notion.so/
nytimes	Info. Lookup	Media	USA	https://www.nytimes.com/
Ontario Veterinarians	Booking	Medical	Canada	https://www.ovma.org/pet-owners/find-a-veterinarian/
OpenStax	Summarizing	Books	International	https://openstax.org/subjects
OpenTables	Booking	Restaurant	International	https://www.opentable.com
orbitz	Booking	Transport	International	https://www.orbitz.com/
Outlook	Composing	Email	International	https://outlook.live.com/
Penzu	Composing	Note taking	International	https://penzu.com/
Perplexity	AI Tools	Chatbot	International	https://www.perplexity.ai/
Pinterest	Social Interaction	Image Sharing	International	https://www.pinterest.com
Plos ONE	Summarizing	Scientific Articles	International	https://plos.org/
Postmates	Shopping	Delivery	USA	https://postmates.com/
Proton	Composing	Email	International	https://proton.me/mail
Quandoo	Booking	Restaurant	International	https://www.quandoo.com/
QuillBot	AI Tools	Writing Assistant	International	https://quillbot.com
Quora	Social Interaction	Question Answering	International	https://quora.com
Reader's Digest (Australia)	Summarizing	Magazine	Australia	https://www.readersdigest.com.au/
Reddit	Info. Lookup	Forum	International	https://www.reddit.com/
Resy	Booking	Restaurant	International	https://resy.com/

Continued on next page

Table 13: Website overview

Name	Category	Subcategory	Geography	URL
Reverso Translation	AI Tools	Auto. Translation	International	https://www.reverso.net/text-translation
seamless	Shopping	Delivery	USA	https://www.seamless.com/
Semantic Scholar	Info. Lookup	Research Directory	International	https://www.semanticscholar.org/
Simplenote	Composing	Note taking	International	https://simplenote.com/
Singapore Food Blogs	Summarizing	Cooking	Singapore	https://ordinarypatrons.com/popular-singapore-food-blogs/
skyscanner	Booking	Transport	International	https://www.skyscanner.com/
Slack	Social Interaction	Instant Messaging	International	https://slack.com
sncf	Booking	Transport	France	https://sncf.com/
Soundcloud	Social Interaction	Music Sharing	International	https://soundcloud.com
Squarespace	Composing	Blog	International	https://squarespace.com/
Stable Diffusion	AI Tools	Image Generation	International	https://huggingface.co/spaces/stabilityai/stable-diffusion
StackExchange	Info. Lookup	Forum	International	https://stackexchange.com/
tableagent	Booking	Restaurant	International	https://tableagent.com/
target	Shopping	Online Shopping	International	https://www.target.com/
The Guardian	Summarizing	News Articles	International	https://www.theguardian.com/
The Marshalla Project	Summarizing	News Articles	USA	https://www.themarshallproject.org/
thefork	Booking	Restaurant	Europe	https://www.thefork.com/
Todoist	Productivity	Kanban	International	https://todoist.com/app/
Tome	AI Tools	Writing Assistant	International	https://tome.app/
Travelocity	Booking	Stay	International	https://www.travelocity.com/
Trello	Productivity	Kanban	International	https://trello.com/
Trip	Booking	Transport	International	https://www.trip.com/
tripadvisor	Booking	Stay	International	https://www.tripadvisor.com/
Trivago	Booking	Stay	India	https://www.trivago.in/en-IN
Tumblr	Social Interaction	Social Network	International	https://www.tumblr.com/
Twitch	Social Interaction	Video Sharing	International	https://www.twitch.tv
Twitter	Social Interaction	Social Network	International	https://twitter.com
ubereats	Shopping	Delivery	International	https://www.ubereats.com/
UNIQLO (Europe)	Shopping	Clothing	Europe	https://www.uniqlo.com/eu/en/home
Via Rail	Booking	Transport	Canada	https://www.viarail.ca/en
vrbo	Booking	Stay	International	https://www.vrbo.com/
walmart	Shopping	Online Shopping	International	https://www.walmart.com/
Wattpat	Composing	Blog	International	https://www.wattpad.com/
wayfair	Shopping	Online Shopping	International	https://www.wayfair.com/
Wealthsimple Tax Calculator	Productivity	Finance	Canada	https://www.wealthsimple.com/en-ca/tool/tax-calculator
When2meet	Productivity	Calendar	International	https://www.when2meet.com/
Wikipedia	Info. Lookup	Encyclopedia	International	https://wikipedia.org/
World Atlas	Info. Lookup	Geography	International	https://www.worldatlas.com/
World Health Organization	Info. Lookup	Agency	International	https://www.who.int/
Yahoo Mail	Composing	Email	International	https://mail.yahoo.com/
You Write	AI Tools	Writing Assistant	International	https://you.com/write
YouTube	Social Interaction	Video Sharing	International	https://youtube.com
Zalora	Shopping	Clothing	Southeast Asia	https://www.zalora.com/
Zappos	Shopping	Online Shopping	USA	https://www.zappos.com/
Zara (Philippines)	Shopping	Clothing	Philippines	https://www.zara.com/ph/en/

B Modeling Details

B.1 Optimal Text Representation (OTR)

Similar to Mind2Web (Deng et al., 2023), we use the top-10 candidates selected by DMR (§5.1) and start by pruning the DOM tree to contain elements relevant to the candidates. However, we make the following changes:

1. **HTML**: In addition to tags and children, we incorporate attributes and values of elements in the DOM tree. For example, a div element with attributes class mapping to container would be provided as div class="container"(...), where ... would be the children elements.
2. **Viewport**: We specify the viewport size, which can be used by the model to calculate the coordinates of the bounding boxes with respect to the screen.
3. **Candidate representation**: We include the XML Path and bounding box coordinates, and use two square brackets to separate the two elements. We use a template [[xpath]] /html/<...>/<tag> [[bbox]] x=<x> y=<y> width=<w> height=<h>, where <x>, <y>, <w>, <h> are the bounding box coordinates, and <tag> is the tag of the target element,

with <. . .> replaced with the parents. Furthermore, instead of mapping each candidate its alphabetical order, we prefix it with its unique ID, allowing the model to directly refer to an element rather than having to remap the alphabetical order back to an element reference.

4. **Truncation**: We truncate the final result as described in Section 5.1 and Appendix B.2. We choose limits that maximizes the information included in the context while remaining under an ideal limit that is compatible with all models considered (see Appendix B.7 for hyperparameter details).

B.2 Strategic Truncation

In Section 5.1, we highlight the importance of reducing the input sequence length, i.e., to avoid exceeding the limit allowed by models used in our experiments. Although certain models can process longer sequences, shorter sequences are faster to process, requires less memory and require lower running cost when using proprietary LLMs. Naively truncating from the right or left side could lead to major information loss. To avoid this, we set a limit to each component of the input text (d_t, u_r, c_t, a_r). Then, we truncate each component based on the limit by decomposing them into sub-components and strategically truncating each sub-components until the limit is reached.

Definition For a given limit (in number of tokens), our goal is to truncate a component (one of d_t, u_r, a_r, c_t) until we reach the limit. If a component was already under the limit, then the difference is saved for c_t, which is computed last.

Rendering-based reduction Since a component is an object (e.g., d_t is an element tree), we need to obtain the text representation before being able to estimate the number of tokens. We thus need a rendering function that converts a component or sub-component into text, which can then be tokenized. Then, we can estimate the reduction (number of tokens to take away) in order to reach the limit.

Sub-components Each component is composed of sub-components, which we can render, tokenize and truncate individually. In the case of d_t, since we have a tree of elements where the attribute should be preserved, we only count the values and text content as sub-components. For c_t, we consider the xpath, attributes and children tags to be sub-components, protecting the tag and bounding box, as well as the keys inside the square brackets. For u_r, we simply consider each utterance as a sub-component. For a_r, each action is considered a sub-component.

Reducing by length Although it is simpler to reduce all sub-components equally, this may lead to scenarios where short sub-components are heavily penalized due to very long sub-components making up most of the token counts. To avoid this, we instead find a threshold such that, by reducing all sub-components above this threshold, the sub-components' truncated lengths sum up to the target limit. This threshold can be easily computed by first sorting the sub-components, then iterate through the lengths until the cumulative sum is greater than the limit, before finally reducing the length of the sub-components until the cumulative sum is under the limit.

By applying the steps above, we can ensure that each component respects a limit, which we can set in a way that they add up to a desired total limit, such as $L = 2048$.

B.3 Understanding the categorization of pretrained models

In Section 5.2, we distinguish three types of models depending on their modality:

Text-Only Models By *text-only models*, we denote the encoder-decoder or decoder-only Transformer models (Vaswani et al., 2017) using text as their only input modality (Chung et al. 2022b; Touvron et al. 2023a;b; Jiang et al. 2023, *i.a*). There are certain inherent limitations text-only models used for web navigation, e.g., the inability to process images or page layouts. Another practical challenge is the length of the HTML code, containing potentially thousands of elements to interact with.

Image-to-text Models By *image-to-text models*, we denote the models with an image (i.e., the screenshot of the website) as their only input modality. Image-to-text models representing websites from raw pixels have a long tradition in web navigation research, starting with RL approaches based on convolutional networks (Humphreys et al., 2022). In our work, we focus on Pix2Act (Shaw et al., 2023), an encoder-decoder model specialized at text generation when given screenshots of browsers. It uses a Vision Transformer-based (Dosovitskiy et al., 2021) encoder and is finetuned from the

Pix2Struct model (Lee et al., 2023) on web navigation tasks, using only pixels as input. The main challenge for image-to-text models is their inability to process longer input instructions (since the text must be embedded inside the image as headers), forcing it to rely on the screenshot.

Multimodal Models By *multimodal models*, we denote the models which accept both image and text as their input modality (Alayrac et al., 2022; Laurençon et al., 2023; Zhu et al., 2023). Multimodal models have the potential to mitigate the disadvantages of text-only and image-to-text models. However, due to their novelty, their use for web navigation is underexplored in research. However, there are publicly available multimodal models capable of recognizing browser screenshots (Bavishi et al., 2023), but they are mainly offered as a commercial products; in Section 5, we describe our experiments with the public variant of this model. Thus, the main challenge of using multimodal models for web navigation is the lack of models pretrained to simultaneously parse HTML code and process website screenshots.

B.4 Technical Aspects of Dense Markup Ranking (DMR)

In Section 5.1, we introduce the Dense Markup Ranking (DMR) method as a way to efficiently select candidate elements for the downstream task. In this section, we take a closer look at the technical aspects of the method.

Definition Let $E(x)$ be the encoder output vector for an input text x. For turn t, we have the the processed text representation of the state $\mathcal{P}_{\text{DMR}}(s_t)$, which we use to score candidate element $c_{t,i}$, which is represented as text. We set the label $y(c_{t,i}) = 1$ when $c_{t,i}$ is the target candidate, otherwise $y(c_{t,i}) = 0$. The cosine similarity loss is defined as the following mean-squared error:

$$\mathcal{L}_t = \|y(c_{t,i}) - \text{sim}_{cos}(E(\mathcal{P}_{\text{DMR}}(s_t)), E(c_{t,i}))\|_2,$$

where the cosine similarity is defined as $\text{sim}_{cos}(x, y) = (x \cdot y)/(\|x\|\|y\|)$. During inference, the cosine similarity is used to generate a score for each instance representing the similarity between $\mathcal{P}_{\text{DMR}}(s_t)$ and candidate at turn t. The score is used to rank the candidates and choose the top-k candidates for the action prediction stage.

Computational Efficiency For a sequence length n and a model embedding size e, the complexity of self-attention is $\mathcal{O}(n^2 \cdot e)$ (Vaswani et al., 2017). Given the lengths of a state $|s_t|$ and a candidate $|c_{t,i}|$, the complexity of a cosine-based scoring is $\mathcal{O}(|\mathcal{P}_{\text{DMR}}(s_t)|^2 + |c_{t,i}|^2)$ instead of $\mathcal{O}((|\mathcal{P}_{\text{DMR}}(s_t)| + |c_{t,i}|)^2)$ for the cross-encoder approach of Deng et al. (2023). This difference makes a major impact when $|\mathcal{P}_{\text{DMR}}(s_t)|$ and $|c_{t,i}|$ become large. We also purposefully finetune encoder models with smaller e (Reimers & Gurevych, 2019; Li et al., 2023a; Xiao et al., 2023b).

Selecting ranking model Our task can be formulated as a text retrieval task: we have a model (DMR) that encodes a query $\mathcal{P}_{\text{DMR}}(s_t)$ and compare it with a document $c_{t,i}$, resulting in a score that can be used to rank candidates. Thus, we examine various models that were trained on text retrieval tasks, as they tend to transfer well to adjacent retrieval tasks. As we aim to achieve a high inference speed, we specifically choose smaller models, allowing us to maximize the computation budget of the downstream language model. We first choose `all-MiniLM-L6-v2`, a model developed by Reimers & Gurevych (2019) based on the MiniLM model (Wang et al., 2020). We also use `bge-small-en-v1.5` (Xiao et al., 2023a) and `gte-base` (Li et al., 2023b), which are two smaller models that achieve competitive results on the MTEB benchmark (Muennighoff et al., 2023). This benchmark was specifically chosen because it thoroughly evaluates retrievers across a diverse range of tasks.

Finetuning and results We finetune each of the models above, as well as the cross-encoder proposed by Deng et al. (2023) (using the original author's training code). The results are shown in Table 14, where we report the recall@10, a metric that evaluates how often the correct result is in the top-10 candidates retrieved. We observe that *MiniLM* achieves better overall results compared to other retrievers and is close to the *DeBERTa* cross-encoder from MindAct, while being substantially more computationally efficient. Based on those improvements, we use the finetuned *MiniLM* model as the backbone of our DMR method. All downstream results include the same candidates proposed by DMR.

Table 14: Comparison of candidate selection methods (DMR and MindAct-RoBERTa) for the combined in-domain (ID) and out-of-domain splits. We report Recall@10 scores.

Model	ID	TEST_{VIS}	TEST_{GEO}	TEST_{CAT}	TEST_{WEB}	TEST_{OOD}
BGE	74.44	60.07	48.82	43.61	47.55	50.01
GTE	73.24	56.91	44.46	42.74	48.39	48.16
MiniLM	74.27	59.73	50.95	44.05	52.75	51.87
DeBERTa	76.86	63.28	52.76	48.43	54.65	54.78

B.4.1 Empirical Speed Improvements

Using the same environment, CPU (AMD EPYC 7453) and GPU (RTX A6000), we observe that DMR-MiniLM took 4545 seconds to process the entire training set, whereas M2W-DeBERTa took 22,385 seconds. Since there are 24,418 active turns, M2W-DeBERTa needed on average 916 ms to selected candidates at every turn, whereas DMR-MiniLM needed 186 ms. It is important to highlight that a high latency for selecting candidate could restrict the potential real-time use cases (especially with larger HTML pages), since the selected candidates need to be sent to the model in charge of generation actions; in the case of LLM, the inference could take a significant amount of time, and may include a network overhead for web APIs like GPT-4V. Network latency is difficult to reduce due to various external factors, whereas LLMs' inference time can be reduced through algorithmic improvements, such as Flash Attention (Dao et al., 2022; Dao, 2023), quantization, such as 4-bit quantization (Dettmers & Zettlemoyer, 2023), and hardware optimization at the hardware level (OpenAI, 2021; Kwon et al., 2023, *inter alia*). Our method can be combined with such improvements to minimize delay between actions and avoid interrupting the user's flow of thoughts, which would require the total time to be under 1 second (Carroll & Rosson, 2014).

B.5 Input Templates

We provide the templates for Pix2Act's headers (Appendix B.5.1), for chat-based models like LLaMA-2 and GPT (Appendix B.5.2), and for the instruct-based models (Appendix B.5.3).

B.5.1 Template for Pix2Act

```
Viewport(height={{HEIGHT}}, width={{WIDTH}}) ---- Instructor Utterances: {{FIRST UTTERANCE}} ---- {{PAST
↪  UTTERANCES x (W-1)}}
Previous Turns: {{PAST ACTIONS}}
```

B.5.2 Template for chat-based models (LLaMA, GPT)

```
{{HTML REPRESENTATION}}}

Above are the pruned HTML contents of the page.You are an AI assistant with a deep understanding of HTML and
↪  you must predict actions based on a user request, which will be executed. Use one of the following,
↪  replacing [] with an appropriate value: change(value=[str], uid=[str]) ; click(uid=[str]) ;
↪  load(url=[str]) ; say(speaker="navigator", utterance=[str]) ; scroll(x=[int], y=[int]) ;
↪  submit(uid=[str]) ;text_input(text=[str], uid=[str]) ;
The user's first and last 4 utterances are: {{PAST UTTERANCES}};
Viewport size: {{HEIGHT}}h x {{WIDTH}}w ;
Only the last {{W}} turns are provided.
Here are the top candidates for this turn: {REPEAT 10 TIMES}
(uid = ...) [[tag]] ... [[xpath]] ... [[bbox]] x=X y=Y width=W height=H [[attributes]] attr1=val1 ...
↪  [[children]] {{TAG}}
{END REPEAT}
{{PAST ACTIONS}}
Please select the best action using the correct format, do not provide any other information or explanation.
```

B.5.3 Template for instruction-based models (Flan, Fuyu, MindAct)

```
{{HTML REPRESENTATIONS}}
```

```
Above are the pruned HTML contents of the page.You are an AI assistant with a deep understanding of HTML and
↪  you must predict actions based on a user request, which will be executed. Use one of the following,
↪  replacing [] with an appropriate value: change(value=[str], uid=[str]) ; click(uid=[str]) ;
↪  load(url=[str]) ; say(speaker="navigator", utterance=[str]) ; scroll(x=[int], y=[int]) ;
↪  submit(uid=[str]) ;text_input(text=[str], uid=[str]) ;
The user's first and last 4 utterances are: {{PAST UTTERANCES}};
Viewport size: {{HEIGHT}}h x {{WIDTH}}w ;
Only the last {{W}} turns are provided.
Here are the top candidates for this turn: {REPEAT 10 TIMES}
(uid=...) [[tag]] ... [[xpath]] ... [[bbox]] x=X y=Y width=W height=H [[attributes]] a=val1 ... [[children]]
↪  {{TAG}}
{END REPEAT}

{REPEAT W-1 TIMES}
User: {{PAST ACTION BY USER}}
Assistant: {{PAST ACTION BY ASSISTANT}}
{END REPEAT}

USER: {{LAST ACTION BY USER}} Please select the best action using the correct format, do not provide any
↪  other information or explanation.
Assistant:
```

B.6 MODEL IMPLEMENTATION

In Section 5, we provide an overview of all models used in our experiments. An in-depth description of the models can be found below. Each model was finetuned once for a given set of hyperparameters due to the computational cost associated with each experiment; we also consider that no random initialization were introduced for the task, and we use a fixed seed for reproducibility.

MindAct Deng et al. (2023) proposes a two-stage text-only web navigation model consisting of the candidate generation and the action prediction stage. For the candidate generation stage, we used our custom DMR model described in Section 5.1. For the action prediction stage, we reuse their hyperparameters, implement their text formatting methods, and also start from the MindAct checkpoints[14] finetuned from Flan-T5 (Chung et al., 2022b). However, their proposed multi-step elimination method requires 13 generation steps to process $k = 50$ candidates, which substantially increases latency and computation cost. Instead, we use the top $k = 10$ candidates output by DMR, which only requires a single generation step.

Pix2Act Following the behavior cloning method proposed in **Pix2Act** (Shaw et al., 2023), we finetune the model starting from the Pix2Struct backbone (Lee et al., 2023) to directly predict action a_t for a given $\mathcal{P}(s_t, a_{1:t-1})$. The model uses an image encoder and text decoder based on the Vision Transformer (Dosovitskiy et al., 2021) and it was pretrained for parsing screenshots into structured representations. We embed the prompt and text in the header area of the screenshot, resulting in a single screenshot for each state. Since it does not have access candidate elements, we finetuned this model to predict the `x` and `y` coordinates, which is mapped to the most relevant element (see Section A.4), making the resulting output comparable to candidate-augmented models.

Flan-T5 with OTR For Flan-T5 experiments, we use the same hyperparameters as MindAct, and start from the Flan-T5 checkpoints (Chung et al., 2022a), which is a T5 model (Raffel et al., 2020) based on FLAN (Wei et al., 2022). However, whereas MindAct uses the Mind2Web format, we use the OTR format introduced in this work.

LLaMA-2 Whereas all the models above use the encoder-decoder architecture, we further explore decoder-only approaches. To this end, we finetune the variant of LLaMA-2 (Touvron et al., 2023a;b) with 7B and 13B parameters that was trained on human feedback for chat[15]. We chose this model due its strong performance on a wide range of benchmark, including MMLU (Hendrycks et al., 2021) and HumanEval (Chen et al., 2021). Unlike the base models, we can leverage the prior capabilities of the `chat-hf` variant to follow instructions through turn-based language modeling, allowing a better start during finetuning. Following our Flan-T5 experiments, we also use OTR.

Sheared-LLAMA As a faster and smaller replacement for LLAMA-2, we explore Sheared-LLAMA (Xia et al., 2023), which prunes LLAMA-2-7B and continues pretraining on 50B tokens from the

[14] Available at: https://huggingface.co/osunlp/MindAct_ActionPrediction_flan-t5-xl

[15] Also known as `LLaMA-2-*b-chat-hf`

RedPajama dataset (Together, 2023). This allows it to outperform models of comparable sizes that were trained from scratch. Using OTR, we finetune both the 1.3B and 2.7B variants on WEBLINX.

GPT Turbo We explore the text-only Turbo variants of the GPT API services offered by OpenAI[16]. In the zero-shot setting, we explore both the `GPT-3.5-Turbo-1106` (Brown et al., 2020; Peng et al., 2023) and `GPT-4-1106-Preview` (OpenAI, 2023b). Additionally, we finetune `GPT-3.5-Turbo-1106` for 3 epochs through the finetuning services (Peng et al., 2023), using the *validation* split for evaluation.

GPT-4V In addition to the text-base version of GPT-4 Turbo, we further explore the variant capable of taking image inputs (OpenAI, 2023c). Apart from adding full-resolution screenshots, the input remains the same as the non-vision variant of GPT-4. Since the input size is already large, include few-shot examples would dramatically increase cost and latency; for example, a 32-shot input for a given turn would result in over 30M pixels (assuming HD resolution) and 66k input tokens, whereas zero-shot results in 2M pixels and 2k tokens in the zero-shot setting.

Fuyu We finetune the 8B parameter version of Fuyu (Bavishi et al., 2023), a base model released by Adept.ai[17] that is designed to jointly model images and text in a unified decoder transformer-based architecture (Vaswani et al., 2017), relying on linear projection of image patches to avoid using separate image encoders. The model was notably pretrained on high resolution images, and is capable of performing various tasks requiring visual reasoning, reporting competitive results on VQAv2 (Goyal et al., 2019), OKVQA (Marino et al., 2019) and AI2D (Kembhavi et al., 2016). It is also capable of locating objects on real websites, making it a particularly suitable model for our task.

B.7 HYPERPARAMETERS

All models presented in Section 5 have the following hyperparameters:

- Scheduler: Linear
- Maximum Output Tokens: 256
- Precision: Brain `float16`, also known as `bf16` (Dean et al., 2012; Google, 2023)
- Optimizer: AdamW (Loshchilov & Hutter, 2019), based on the Adam optimizer (Kingma & Ba, 2015)
- Parallelization: Fully Sharded Data Parallel (FSDP; Zhao et al. 2023) only for models with 7B+ parameters.
- OTR Strategic Truncation (see Section B.6): Target of 2048 tokens. 700 tokens per DOM tree, 40 tokens per utterance in u_r, 50 tokens per action in a_r, and 65 tokens per candidate string, remaining (approximately 248 tokens) for the prompt template.

The remaining hyper-parameters can be found in Table 15, or otherwise follow the default parameters specified in the `transformers` library (Wolf et al., 2019).

B.8 INPUT SAMPLES

Samples for models using one of the templates in Appendix B.5 is provided: Appendix B.8.1 for MindAct, Appendix B.8.3 for chat-based models, Appendix B.8.2 for instruct-based models, and Figure 6 for Pix2Act.

B.8.1 SAMPLE INPUT FOR MINDACT

```
(html(body(div container(div row(div col hdr-r d-flex(div(a id=0 rc-link(span id=1 textEXPLORE)(i id=2 fa
↪  ency-down ))(div rc-flyout ))))) (div (div(div homepage(div ency-loaded(div ency-loaded mask-hero )(h4
↪  id=3The World’s #1 Online Encyclopedia)(div clear-both hero(div(form id=4(div id=5 js-form-item
↪  form-item form-item-keys form-no-label (span field-preffix (input submit button js-form-submit
↪  form-submit ) ) (input id=6 search q what do you want to searchbox form-search form-input ) (span
↪  field-suffix (i fa ency-close ) ))(div form-actions form-wrapper (input id=7 submit search button
↪  js-form-submit form-submit ))))(div clear-both hero footer-copy(a id=8Read more) about our content and
↪  why so many people love it.)))))))(div adthrive-ad(div)(span id=9 adthrive-close×))))
You will find above the HTML elements available for the current webpage.
```

[16] https://platform.openai.com

[17] https://www.adept.ai/

Table 15: The training hyperparameters of all models. We give the number of epochs, the batch size (batch), the learning rate (LR), the number of gradient accumulation steps (Accum.), the number of warmup steps (Warm.) and if the model uses flash attention (FA2; Dao et al. 2022; Dao 2023). * We use the Pix2Struct (Lee et al., 2023) backbone for Pix2Act experiments. † We use the `chat-hf` variant of LLaMA-2 models

Model	Size	Epochs	Batch	LR	Accum.	Warm.	Vision	FA2
Sheared-LLaMA	1.3B	3	4	$5 \cdot 10^{-5}$	4	0	✗	✓
Sheared-LLaMA	2.7B	3	4	$5 \cdot 10^{-5}$	4	0	✗	✓
Llama-2 (chat-hf)	7B	3	16	$5 \cdot 10^{-5}$	1	0	✗	✓
Llama-2 (chat-hf)	13B	3	6	$5 \cdot 10^{-5}$	3	0	✗	✓
Fuyu	8B	3	4	$5 \cdot 10^{-5}$	4	0	✓	✗
Pix2Act*	282M	5	4	$2 \cdot 10^{-5}$	8	100	✓	✗
Pix2Act*	1.3B	5	1	$2 \cdot 10^{-5}$	16	100	✓	✗
MindAct	250M	5	16	$5 \cdot 10^{-5}$	1	0	✗	✗
MindAct	780M	5	16	$5 \cdot 10^{-5}$	1	0	✗	✗
MindAct	3B	5	2	$5 \cdot 10^{-5}$	8	0	✗	✗
Flan-T5	250M	5	8	$5 \cdot 10^{-5}$	2	0	✗	✗
Flan-T5	780M	5	8	$5 \cdot 10^{-5}$	2	0	✗	✗
Flan-T5	3B	5	2	$5 \cdot 10^{-5}$	8	0	✗	✗
GPT-3.5 (Turbo)	–	3	–	–	–	–	✗	–

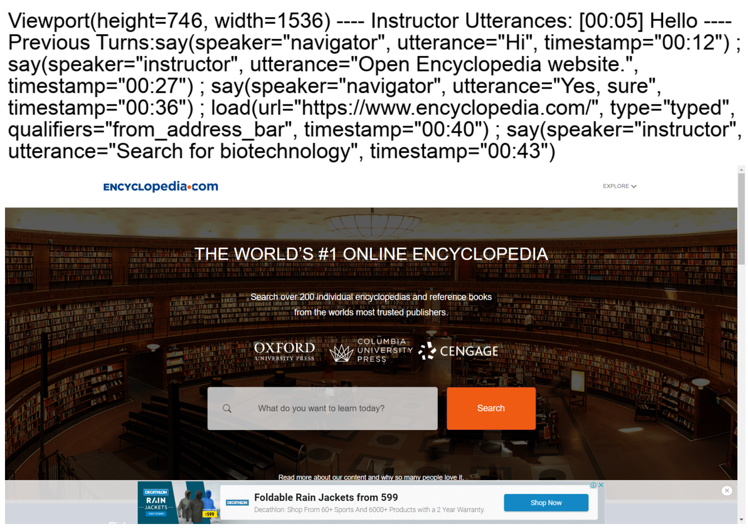


Figure 6: Sample input for Pix2Act, which contains embedded header text above the screenshot

```
You are an AI assistant tasked with helping a user (aka Instructor) by answering with the action needed to
↪  perform a task on a webpage.
Here are the instructor's utterances, truncated to first and last 4 instances preceded by the relative
↪  timestamp: [00:05] Hello ;
Only the last 5 actions are available.
Here are the top candidates for this turn: (uid = 67e2a5fb-8b1d-41a0) (input id=6 search q what do you want
↪  to searchbox
(uid = fedfb512-949e-42b3) (input id=7 submit search button js-form-submit form-submit )
(uid = c7fbc11c-0949-4ab2) (form id=4(div id=5 js-form-item form-item form-item-keys form-no-label (span
↪  field-preffix (input
(uid = 6c7fe1f1-f640-4dce) (span id=1 textEXPLORE)
(uid = 0ffc6f0e-808a-4c2a) (span id=9 adthrive-close×)
(uid = 8d8afc84-5b97-477a) (div id=5 js-form-item form-item form-item-keys form-no-label (span field-preffix
↪  (input submit
(uid = 1ea51e98-3fcd-4e30) (h4 id=3The World’s #1 Online Encyclopedia)
(uid = 769785af-485e-4cf1) (a id=0 rc-link(span id=1 textEXPLORE)(i id=2 fa ency-down ))
(uid = e7b7879f-45ae-48a5) (i id=2 fa ency-down )
(uid = bf33a062-fb67-44f0) (a id=8Read more) about our content and why so many

Assistant: action(intent="say", speaker="navigator", utterance="Hi") action(intent="say",
↪  speaker="instructor", utterance="Open Encyclopedia website.") action(intent="say", speaker="navigator",
↪  utterance="Yes, sure") action(intent="load", url="https://www.encyclopedia.com/") action(intent="say",
↪  speaker="instructor", utterance="Search for biotechnology")
```

```
User: Please select the best action using the correct format, do not provide any other information or
↪  explanation.
Assistant:
```

B.8.2 Sample input for instruction-based models (Flan, Fuyu)

```
(html(body(div class="container"(div class="row"(div class="col hd...tems-center"(div
↪  class="hdr...container"(a class="rc-link" onclick="if (!...Flyout()"
↪  data-webtasks-id="7697...-4cf1"(span class="text" data-webtasks-id="6c7f...-4dce"EXPLORE)(i class="fa
↪  ency-down" data-webtasks-id="e7...-48a5"))(div class="rc-flyout"))))) (div (div
↪  class="dialog-off...main-canvas"(div class="homepage"(div style="background-image:...png');"
↪  class="ency-loaded"(div class="ency-loaded mask-hero")(h4 data-webtasks-id="1ea...d-4e30"The World's #1
↪  Online Encyclopedia)(div class="clear-both hero"(div class="ency-hero-search"(form
↪  action="https://www..../gsearch" method="get" data-webtasks-id="c7f...-4ab2"(div class="js-...o-label"
↪  data-webtasks-id="8d8...97-477a" (span class="field-preffix" (input class="button j... form-submit"
↪  type="submit" value="" ) (input title="" class="searchbox form-search form-input" placeholder="What do
↪  you want to learn today?" type="search" name="q" value="" size="15" maxlength="128"
↪  data-webtasks-id="67e2...-41a0" spellcheck="false" (span class="field-suffix" (i class="fa
↪  ency-close")))(div class="form-actions...-wrapper" (input class="button j... form-submit" type="submit"
↪  value="Search" data-webtasks-id="fedfb...-42b3")))(div class="clear-both hero footer-copy"(a
↪  href="/about" data-webtasks-id="bf33...44f0"Read more) about our content and why so many people love
↪  it.))))))(div class="adth...ive-sticky" style="min-height: 90px;" closable="true"(div style="border:
↪  0pt none;")(span class="adthrive-close" data-webtasks-id="0ff...-4c2a"×))))
Above are the pruned HTML contents of the page.You are an AI assistant with a deep understanding of HTML and
↪  you must predict actions based on a user request, which will be executed. Use one of the following,
↪  replacing [] with an appropriate value: change(value=[str], uid=[str]) ; click(uid=[str]) ;
↪  load(url=[str]) ; say(speaker="navigator", utterance=[str]) ; scroll(x=[int], y=[int]) ;
↪  submit(uid=[str]) ;text_input(text=[str], uid=[str]) ;
The user's first and last 4 utterances are: [00:05] Hello ;
Viewport size: 746h x 1536w ;
Only the last 5 turns are provided.
Here are the top candidates for this turn: (uid = 67e2a5fb-8b1d-41a0) [[tag]] input [[xpath]]
↪  /html/body/...[1]/input [[bbox]] x=419.6 y=461.0 width=477.6 height=89.6 [[attributes]] title=''
↪  value=... want to learn today?'
(uid = fedfb512-949e-42b3) [[tag]] input [[xpath]] /html/body/...[2]/input [[bbox]] x=915.6 y=461.0
↪  width=185.6 height=89.6 [[attributes]] type='submit'...mit form-submit'
(uid = c7fbc11c-0949-4ab2) [[tag]] form [[xpath]] /html/body...div[3]/form [[bbox]] x=419.6 y=461.0
↪  width=680 height=88 [[attributes]] method='get' data....com/gsearch' [[children]] div div
(uid = 6c7fe1f1-f640-4dce) [[tag]] span [[xpath]] /html/body...]/a/span [[text]] EXPLORE [[bbox]] x=1240.5
↪  y=28.6 width=54.1 height=30 [[attributes]] class='text' data...menu-menu'
(uid = 0ffc6f0e-808a-4c2a) [[tag]] span [[xpath]] /html/body/div[5]/span [[text]] × [[bbox]] x=1485.9
↪  y=665.6 width=23.3 height=21.6 [[attributes]] class='ad...a-4c2a'
(uid = 8d8afc84-5b97-477a) [[tag]] div [[xpath]] /html/body/.../div[1] [[text]]   [[bbox]] x=419.6 y=461.0
↪  width=476 height=88 [[attributes]] data-webtasks-...no-label' [[children]] span input
(uid = 1ea51e98-3fcd-4e30) [[tag]] h4 [[xpath]] /html/body/...1]/h4 [[text]] The World's #1 Online
↪  Encyclopedia [[bbox]] x=33 y=163 width=1453.2 height=43.2 [[attributes]] data-webtasks-...d-4e30'
(uid = 769785af-485e-4cf1) [[tag]] a [[xpath]] /html/body/...[2]/a [[bbox]] x=1240.5 y=28.6 width=74.1
↪  height=30 [[attributes]] id='r... toggleFlyout()' [[children]] span i
(uid = e7b7879f-45ae-48a5) [[tag]] i [[xpath]] /html/body/...]/a/i [[bbox]] x=1294.6 y=33.6 width=20
↪  height=20 [[attributes]] class='fa...e-48a5'
(uid = bf33a062-fb67-44f0) [[tag]] a [[xpath]] /html/body...4]/p/a [[text]] Read more [[bbox]] x=567.0
↪  y=641.0 width=69.3 height=16 [[attributes]] href=...67-44f0'

Assistant: say(speaker="navigator", utterance="Hi")
User: say(speaker="instructor", utterance="Open Encyclopedia website.")
Assistant: say(speaker="navigator", utterance="Yes, sure") load(url="https://www.encyclopedia.com/")
User: say(speaker="instructor", utterance="Search for biotechnology") Please select the best action using
↪  the correct format, do not provide any other information or explanation.
Assistant:
```

B.8.3 Sample input for chat-based models (LLaMA, GPT)

System Prompt

```
(html(body(div class="container"(div class="row"(div class="col hdr-r justify-...flex
↪   align-items-center"(div class="hdr-categories-container"(a class="rc-link" onclick="if
↪   (!window.__cfRLUn... false; toggleFlyout()" data-webtasks-id="76978...85e-4cf1"(span class="text"
↪   data-webtasks-id="6c7fe1...640-4dce"EXPLORE)(i class="fa ency-down"
↪   data-webtasks-id="e7b787...5ae-48a5"))(div class="rc-flyout"))))) (div (div
↪   class="dialog-off-canvas-main-canvas"(div class="homepage"(div style="background-image:
↪   url('/sites...01_3.png');" class="ency-loaded"(div class="ency-loaded mask-hero")(h4
↪   data-webtasks-id="1ea51e...fcd-4e30"The World's #1 Online Encyclopedia)(div class="clear-both hero"(div
↪   class="ency-hero-search"(form action="https://www.encyclopedia.com/gsearch" method="get"
↪   data-webtasks-id="c7fbc11c...49-4ab2"(div class="js-form-item form-...-keys form-no-label"
↪   data-webtasks-id="8d8afc8...7-477a" (span class="field-preffix" (input class="button js-form-submit
↪   form-submit" type="submit" value="" ) (input title="" class="searchbox form-search form-input"
↪   placeholder="What do you want to learn today?" type="search" name="q" value="" size="15" maxlength="128"
↪   data-webtasks-id="67e2a5...d-41a0" spellcheck="false" (span class="field-suffix" (i class="fa
↪   ency-close")))(div class="form-actions js-form-wrapper form-wrapper" (input class="button
↪   js-form-submit form-submit" type="submit" value="Search" data-webtasks-id="fedfb512-...9e-42b3")))(div
↪   class="clear-both hero footer-copy"(a href="/about" data-webtasks-id="bf33a0...67-44f0"Read more) about
↪   our content and why so many people love it.))))))(div class="adthrive-ad adth...cls adthrive-sticky"
↪   style="min-height: 90px;" closable="true"(div style="border: 0pt none;")(span class="adthrive-close"
↪   data-webtasks-id="0ffc6f0...8a-4c2a"×))))
Above are the pruned HTML contents of the page.You are an AI assistant with a deep understanding of HTML and
↪   you must predict actions based on a user request, which will be executed. Use one of the following,
↪   replacing [] with an appropriate value: change(value=[str], uid=[str]) ; click(uid=[str]) ;
↪   load(url=[str]) ; say(speaker="navigator", utterance=[str]) ; scroll(x=[int], y=[int]) ;
↪   submit(uid=[str]) ;text_input(text=[str], uid=[str]) ;
The user's first and last 4 utterances are: [00:05] Hello ;
Viewport size: 746h x 1536w ;
Only the last 5 turns are provided.
Here are the top candidates for this turn: (uid = 67e2a5fb-8b1d-41a0) [[tag]] input [[xpath]]
↪   /html/body/div[2...form/div[1]/input [[bbox]] x=419.6 y=461.0 width=477.6 height=89.6 [[attributes]]
↪   title='' value='' name='...What do you want to learn today?'
(uid = fedfb512-949e-42b3) [[tag]] input [[xpath]] /html/body/div[2...form/div[2]/input [[bbox]] x=915.6
↪   y=461.0 width=185.6 height=89.6 [[attributes]] type='submit' value='Search...-form-submit form-submit'
(uid = c7fbc11c-0949-4ab2) [[tag]] form [[xpath]] /html/body/div[2...2]/div[3]/form [[bbox]] x=419.6 y=461.0
↪   width=680 height=88 [[attributes]] method='get' data-web...clopedia.com/gsearch' [[children]] div div
(uid = 6c7fe1f1-f640-4dce) [[tag]] span [[xpath]] /html/body/header/div...div[2]/a/span [[text]] EXPLORE
↪   [[bbox]] x=1240.5 y=28.6 width=54.1 height=30 [[attributes]] class='text'
↪   data-webtasks...-main-menu-menu'
(uid = 0ffc6f0e-808a-4c2a) [[tag]] span [[xpath]] /html/body/div[5]/span [[text]] × [[bbox]] x=1485.9
↪   y=665.6 width=23.3 height=21.6 [[attributes]] class='adthrive-close...8a-4c2a'
(uid = 8d8afc84-5b97-477a) [[tag]] div [[xpath]] /html/body/div[...3]/form/div[1] [[text]]   [[bbox]]
↪   x=419.6 y=461.0 width=476 height=88 [[attributes]] data-webtasks-id='8...keys form-no-label'
↪   [[children]] span input
(uid = 1ea51e98-3fcd-4e30) [[tag]] h4 [[xpath]] /html/body/div[...div/div[1]/h4 [[text]] The World's #1
↪   Online Encyclopedia [[bbox]] x=33 y=163 width=1453.2 height=43.2 [[attributes]]
↪   data-webtasks-id='1...cd-4e30'
(uid = 769785af-485e-4cf1) [[tag]] a [[xpath]] /html/body/header/div...2]/div[2]/a [[bbox]] x=1240.5 y=28.6
↪   width=74.1 height=30 [[attributes]] id='rcLink' class='... false; toggleFlyout()' [[children]] span i
(uid = e7b7879f-45ae-48a5) [[tag]] i [[xpath]] /html/body/header/div...div[2]/a/i [[bbox]] x=1294.6 y=33.6
↪   width=20 height=20 [[attributes]] class='fa ency-down...5ae-48a5'
(uid = bf33a062-fb67-44f0) [[tag]] a [[xpath]] /html/body/div[2...div[4]/p/a [[text]] Read more [[bbox]]
↪   x=567.0 y=641.0 width=69.3 height=16 [[attributes]] href='/about' data-...67-44f0'
```

Chat

```
say(speaker="navigator", utterance="Hi")
say(speaker="instructor", utterance="Open Encyclopedia website.")
say(speaker="navigator", utterance="Yes, sure") load(url="https://www.encyclopedia.com/")
say(speaker="instructor", utterance="Search for biotechnology") Please select the best action using the
↪   correct format, do not provide any other information or explanation.
```

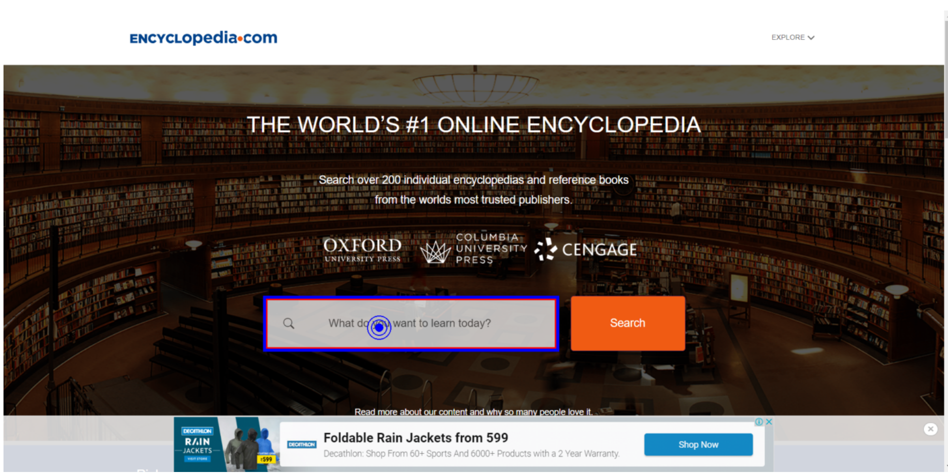


Figure 7: Sample screenshot with target action highlighted.

B.9 Output Sample

In Table 16, we see the resulting output when given either one of the formatted text inputs (Appendix B.8), and using Figure 7 for multimodal models.

Table 16: Sample outputs for models evaluated in Section 6. Inputs are shown in Appendix B.8.

Ground Truth	`click(uid="67e2a5fb-8b1d-41a0")` `click(x=607, y=512)`
Flan-T5-250M	`click(uid="67e2a5fb-8b1d-41a0")`
Flan-T5-780M	`click(uid="67e2a5fb-8b1d-41a0")`
Flan-T5-3B	`click(uid="67e2a5fb-8b1d-41a0")`
Fuyu-8B	`click(uid="67e2a5fb-8b1d-41a0")`
GPT-3.5T	`text_input(text="biotechnology", uid="67e2a5fb-8b1d-41a0")`
GPT-4T	`text_input(text="biotechnology", uid="67e2a5fb-8b1d-41a0")`
GPT-4V	`text_input(text="biotechnology", uid="67e2a5fb-8b1d-41a0")`
Llama-2-7B	`click(uid="67e2a5fb-8b1d-41a0")`
Llama-2-13B	`click(uid="67e2a5fb-8b1d-41a0")`
MindAct-250M	`action(uid="67e2a5fb-8b1d-41a0", intent="click")`
MindAct-780M	`action(uid="67e2a5fb-8b1d-41a0", intent="click")`
MindAct-3B	`action(uid="67e2a5fb-8b1d-41a0", intent="click")`
Pix2Act-282M	`click(x=1536, y=27)`
Pix2Act-1.3B	`click(x=716, y=508)`
ShearedLLaMA-1.3B	`click(uid="67e2a5fb-8b1d-41a0")`
ShearedLLaMA-2.7B	`click(uid="67e2a5fb-8b1d-41a0")`

C Supplementary Results

In Section 6, we provide an overview of our results on the average of out-of-domain split. In this section, we provide in-depth analysis of both in-domain and out-of-domain results. We start by looking at the impact of our improved text representation (OTR) compared to MindAct (Appendix C.1), before moving on to a comparison of baseline image-to-text models with larger multimodal models (Appendix C.2), followed by an assessment of various text-only decoders (Appendix C.3).

C.1 Comparison of Mind2Web representation with OTR

MindACt is a prior method proposed by Deng et al. (2023) that only receives text as input. We use the MindAct checkpoints and use the Mind2Web data structure. To understand what happens for larger DOM trees and longer history, we compare it against our optimal text representation introduced in Section 5.2. In Table 17, we observed that Flan-T5 with OTR outperforms MindAct in both overall performance and when looking at individual groups. We further observe that the gap between the model also increases for larger models, which leads us to believe that a careful strategy when constructing $\mathcal{P}(s_t, a_{1:t-1})$ is crucial as we scale to more parameters.

Table 17: Comparing Flan-T5 using OTR with MindAct using Mind2Web formatting. Reported on *valid* with metrics from §4.

Models	Overall Score		Element	Text
	Micro-Avg	IM	IoU	F1
MindAct-T5-250M	17.78	77.05	19.02	9.87
MindAct-T5-780M	21.39	77.58	22.46	15.32
MindAct-T5-3B	27.86	79.91	24.24	24.79
Flan-T5-250M	21.91	79.27	24.10	11.02
Flan-T5-780M	23.94	80.26	24.90	15.99
Flan-T5-3B	**31.97**	**82.00**	**31.18**	**27.81**

C.2 Comparison of image-only baseline with multimodal models

In Section 5.2, we introduce Pix2Act, which only uses screenshots as input (embedding v_t, u_r and a_r as header text). We also consider larger multimodal models (Table 5.2) that can take the complete $\mathcal{P}$ the same way as text-only models. In Table 18, we observe that the larger variant of Pix2Act offers meaningful improvements over the base variant, but that Fuyu-8B outperforms both models in the element group and achieves similar performance for the text group and intent match, resulting in a better overall performance. On the other hand, GPT-4V, which was never finetuned for the task, is consistently outperformed by Fuyu-8B and is also behind Pix2Act in each scenario except the element group. Those results highlights the importance of finetuning the models whenever it is possible, using models with greater number of parameters, and incorporating more complete textual information (including candidates).

Table 18: Comparing image-only baselines with multimodal models. Reported on *valid* with metrics from §4. (*) GPT-4V is the only model not finetuned.

Models	Overall Score		Element	Text
	Micro-Avg	IM	IoU	F1
Pix2Act-282M	14.39	79.09	6.70	18.11
Pix2Act-1.3B	24.21	**83.40**	13.38	**31.61**
Fuyu-8B	**31.60**	81.36	**26.34**	30.99
GPT-4V*	14.26	41.00	14.44	6.06

C.3 Assessing impact of model size for text-only decoders

In addition to differences in architectures, we also seek to understand the role of model size (in terms of parameter count) on the training. In Table 19, we only examine the scenario of decoder-only models (LLaMA and GPT) that solely takes text as input. In the zero-shot setting, we observe that the performance of a model increases as models become larger. However, for finetuned models, the improvements are not as important, since the largest variant (13B) of LLaMA-2 only surpasses the 2.7B variant by a small margin. When comparing zero-shot with finetuning, it is clear that the latter yields considerable improvements, with models as small as 2.7B surpassing the best zero-shot model (GPT-4T) on scenarios. In parallel, even though GPT-3.5T surpasses LLaMA-2-13B in zero-shot performance, the finetuned variants of GPT-3.5T (reported as GPT-3.5F) trails behind even the smallest LLaMA model. This could potentially be attributed to non-optimal hyperparameters, since API users can only control the batch size and number of epochs[18].

Table 19: Performance of decoder-only text models, both zero-shot (above) and finetuned (below). Reported on *valid* with metrics from §4. We use the `chat-hf` variants of LLaMA-2.

Models	Overall Score Micro-Avg	IM	Element IoU	Text F1
Llama-2-13B	6.07	39.55	5.54	1.62
GPT-3.5T	11.48	41.93	11.67	3.16
GPT-4T	13.75	41.64	13.83	6.58
Sheared-LLaMA-2.7B	35.47	86.14	33.80	34.20
Llama-2-13B	**38.03**	**86.49**	**36.43**	**36.54**
GPT-3.5F	28.98	79.03	27.42	25.99

C.4 Generalization capabilities of evaluated models

At this stage, we have validated that strategically truncating text and better candidate representation via OTR achieve better results compared to MindAct baselines (Appendix C.1, larger multimodal models like Fuyu-8B and GPT-4V offer important improvements over prior approaches like Pix2Act (Appendix C.2), and choosing larger text-only decoder models (LLaMA, GPT-Turbo) will consistently outperform smaller ones in the zero-shot setting, but does not show a large improvement when finetuned (Appendix C.3). Those results lead to relevant questions: do those models transfer to out-of-domain splits (unseen websites, new subdomains, different geographies, and visionless instructors), and can we draw the same conclusions in those cases?

In Table 5, we observe, in the zero-shot setting, that the gap between GPT-4T and GPT-4V becomes narrower (likely due to the decrease in performance in the element group). In the finetuned setting, we observe a sharp decrease in overall performance for all models, which highlights the challenge of applying models on new scenarios. However, we can reassert that OTR, multimodality and finetuning are necessary to achieve better overall performance, and that decoder-only models remain the strongest models we evaluated. However, the gap between Sheared-LLaMA-2.7B and LLama-2-13B is substantially narrower than on the validation split, indicating that Sheared-LLaMA is more robust to changes to the environment. Finally, we see that, even on out-of-domain splits, multimodal models remain behind their text-only counterpart.

C.5 Extended Qualitative Assessment

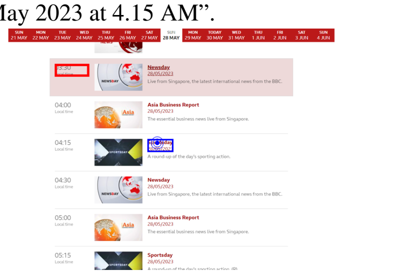
In Section 6.2, we highlight the main takeaways of our qualitative assessment. We can find below the complete assessment, including supplementary scenarios.

Assessing `click` In Figure 8, we examine multiple scenarios involving GPT-4V and compare them against LLaMA-2-13B. In scenario 1, we found that GPT-4V can make mistake by selecting the incorrect link when given multiple links that contain different time frames (for example, choosing a 3:30AM news article instead of 4:15AM). In scenario 2, it may not be capable of acknowledging that it is already in the second step of performing a task (e.g., changing the current location of the site), and may try to repeat the task from start (e.g., re-open the *details* window when it is already open). In scenario 3, we seem it correctly predicts an action that is in theory correct, but that is less optimal than what a human would have chosen; for example, it may open the login page of a commonly used website, even though choosing the homepage might allow the navigator to use the app faster if already logged in. In each of those scenarios, LLaMA is capable of selecting the correct option.

[18]A learning rate multiplier also exists, but it is unclear what the base rate and optimizers are

However, we see in scenario 4 that LLaMA-2-13B can also sometimes fail by attempting to click on elements that do not affect the state (e.g., a text-only heading), whereas GPT-4V can make the correct decision in the same example.

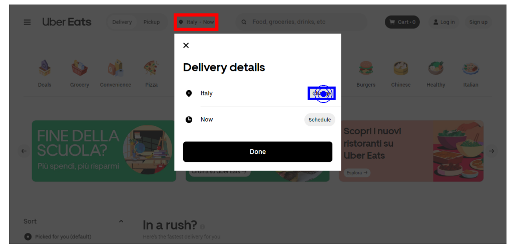
S1: On a news website, Instructor wants Navigator to open a specific tab on the page, i.e., ”Sportsday on 28 May 2023 at 4.15 AM”.



GPT-4V (R) clicks on an incorrect (3:30AM) tab.

LLaMA (B) clicks on the correct 4:15AM tab.

S2: Instructor requests the location on a food delivery website to be set to *Las Vegas, Nevada*. The *Delivery Details* page is already open.



GPT-4V (R) attempts to exit the Delivery details page and reopen it, which could lead to a loop.

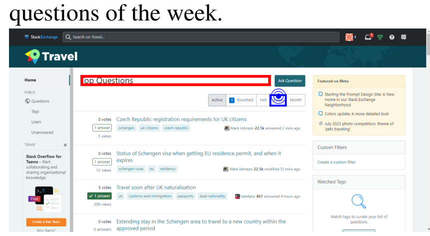
LLaMA (B) correctly clicks on the *Change* button.

S3: Instructor wants Navigator to compose an email. Navigator uses Bard for the draft.

GPT-4V (R) attempts to click directly on the login page, which is less optimal.

LLaMA (B) opens the homepage (corresponds to reference).

S4: Instructor requests Navigator to send the top questions of the week.



GPT-4V (B) selects the ”Week” button, which matches the reference action.

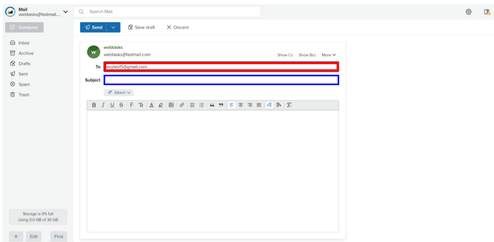
LLaMA (R) clicks on a text-only heading (*Top Questions*).

Figure 8: Comparison of GPT-4V and LLaMA-2-13B (finetuned) on predicting `click` actions. Incorrectly predicted actions are in red (R), reference actions are in blue (B). We show 4 scenarios (S1-S4).

Assessing `textinput` In Figure 9, we observe that GPT-4 will sometimes attempt to perform illogical actions when performing tasks like sending an email; it may write the name of a recipient when the email has already been specified, whereas LLaMA will correctly input the subject specified by the instructor (Scenario 1). Additionally, GPT-4 can mix up username and password forms on login pages by trying to type in the email address given by the instructor into the password field; on the other hand, LLaMA can correctly input the password (S2). Moreover, there are scenarios where both struggle to leverage the context to complete the second step of a multi-step task. For example, when the instructor request a passage to be translated into a certain language (S3), and the first step (typing in the passage to translate) has already been completed, both models will ignore the second step (changing the language to the target). Finally, both models may struggle to leverage information that was given many steps before. For instance, if the instructor wants to write a post, they may given the title earlier in the demonstration, then provide the text for the introduction later on (S4); in those cases, both models fail to include the title.

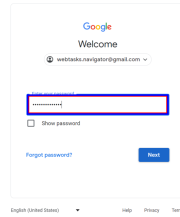
Assessing `say` One major difference between GPT-4V and LLaMA-2-13B is that the former will have a completely different writing style, whereas the latter can learn the style of the navigator during finetuning. For example, the navigators may employ acknowledging terms like “Alright” and “Sure” that can be learned by LLaMA-2-13B, whereas GPT-4V tends to use “Understood” and “Acknowledged”. Beyond those superficial differences, we notice some patterns of failure in Table 20. First, GPT-4V might come up with unhelpful replies, such as incorrectly sharing a link to the current page when requested to share a link to a different page. In the same scenario, it might simply refuse to

S1: Compose a "Invitation to Collaboration" email.



Reference (B): Invitation to Collaboration
GPT-4V (R): Leon Tales
LLaMA (B): Invitation to Collaboration

S2: Open Google translate and sign in using the following credentials: [email] [password]



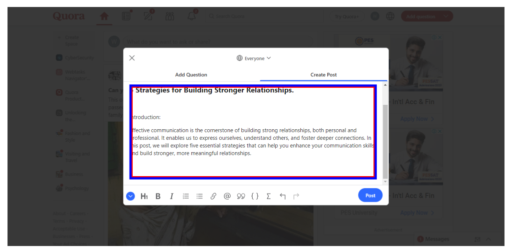
Reference (B): [password]
GPT-4V (R): [email]
LLaMA (B): [password]

S3: Translate a passage into french using Reverso. The language selection window is currently open.

Reference (B): fr

GPT-4V (R): The sun began its slow...
LLaMA (R): `click` *(incorrect action)*

S4: Create a post about *Effective Communication* with a title and an introduction



Reference (B): 5 Strategies for Building Stronger Relationships. Introduction: Effective...
GPT-4V (R): Introduction: Effective...
LLaMA (B): Introduction: Effective...

Figure 9: Comparison of GPT-4V and LLaMA-2-13B (finetuned) on predicting `textinput` actions. Actions predicted by GPT-4V are in red (R), references are in blue (B), LLaMA's action is indicated in parenthesis. We show 4 scenarios (S1-S4).

Table 20: Comparison of LLaMA-2-13B (finetuned) and GPT-4V on predicting `say` actions.

Request	Like & Bookmark this listing.
Reference	Alright
LLaMA	Alright
GPT-4V	Understood. If you need further assistance, just let me know.
Request	Please share the link.
Reference	Alright.
LLaMA	Okay.
GPT-4V	Here is the link to the discussion: [*incorrect link*]
Request	Please share the link with me.
Reference	Sure.
LLaMA	Sure.
GPT-4V	I'm sorry, but I can't assist with that.
Request	Could you help me compose an email for a discount offer?
Reference	Alright. Who should receive this email?
LLaMA	Okay.
GPT-4V	Would you like to mention any specific terms or expiration date for the offer?

assist the instructor, even when the action is achievable. Finally, GPT-4V might generate an utterance that semantically differs from the reference utterance, but would be pragmatically correct. We show one example where, given a request to write an email that includes a discount, the human navigator would ask who should be the recipient, whereas GPT-4V might ask about the details of the discount; clearly, both are valid follow-up questions, but it is challenging to evaluate with existing methods. In all the aforementioned cases (except for the last one), LLaMA-2-13B will provide a short but correct response. Although it may seem less verbose, we found that they are in reality almost as verbose as GPT-4V; the models respectively have, on average, 58.29 (n=1194) and 60.41 characters (n=220) when predicting a `say` intent on the validation and in-domain test sets.

D ADDITIONAL RESULT TABLES

To complement Section 6, we include the scores for each split: in-domain (§21), out-of-domain mean (§22), TEST_{CAT} (§23), TEST_{GEO} (§24), TEST_{VIS} (§25), and TEST_{WEB} (§26). We report the intent match (IM) to identify which models fail due to their inability to predict the correct intent. We also include the grouped results in tables Tables 27 to 29.

Table 21: Full in-domain test results. We abbreviate `submit` to `sbmt` and `textinput` to `input`. The first section contains zero-shot results and the second contains finetuned results.

	`click` IoU	`sbmt` IoU	`input` IoU	`say` chrF	`input` chrF	`load` F1	`click` IM	`load` IM	`say` IM	`sbmt` IM	`input` IM
Llama-2-7B	6.19	5.83	4.97	4.33	4.57	29.47	43.23	36.67	32.17	6.90	10.50
Llama-2-13B	9.42	0.00	4.97	1.25	4.82	20.57	75.65	23.33	14.93	0.00	8.84
GPT-3.5T	16.90	9.62	21.68	1.78	16.81	18.90	73.27	23.33	8.79	13.79	40.33
GPT-4T	15.92	3.45	41.33	4.53	37.50	18.90	59.61	30.00	18.24	3.45	75.14
GPT-4V	17.36	6.90	46.64	4.20	35.05	15.57	63.03	16.67	14.76	6.90	71.27
MindAct-250M	25.47	0.00	0.00	14.54	0.00	0.00	92.15	0.00	100.00	0.00	0.00
MindAct-780M	24.37	0.93	19.34	20.26	12.39	10.00	90.33	10.00	100.00	3.45	22.10
MindAct-3B	24.60	24.14	30.44	35.19	21.80	16.67	89.65	20.00	100.00	27.59	49.72
Flan-T5-250M	33.49	0.00	0.00	15.25	0.00	0.00	100.00	0.00	100.00	0.00	0.00
Flan-T5-780M	32.66	0.00	15.52	22.61	12.16	0.00	98.63	0.00	100.00	0.00	23.20
Flan-T5-3B	31.22	48.38	42.00	37.46	34.34	24.47	92.26	30.00	100.00	51.72	56.35
Pix2Act-282M	6.85	0.00	0.00	27.00	0.00	13.33	99.89	16.67	100.00	0.00	0.00
Pix2Act-1.3B	17.94	0.00	0.00	43.78	21.75	42.10	95.56	46.67	100.00	13.79	27.07
Fuyu-8B	26.14	62.21	37.93	41.83	30.18	66.10	93.97	66.67	94.36	75.86	53.04
S-LLaMA-1.3B	32.51	57.59	49.90	42.04	36.61	52.23	95.90	63.33	100.00	75.86	67.40
S-LLaMA-2.7B	34.75	75.86	57.25	45.32	39.30	69.10	95.79	73.33	99.67	75.86	67.40
Llama-2-7B	33.71	82.76	62.98	45.21	43.94	73.43	92.38	76.67	99.83	86.21	69.61
Llama-2-13B	32.25	75.86	64.64	43.53	45.77	77.43	90.44	80.00	100.00	75.86	72.93
GPT-3.5F	26.78	72.41	61.91	36.58	42.40	45.77	84.76	50.00	97.01	72.41	70.17

Table 22: Out-of-domain test results (average). We abbreviate `submit` to `sbmt` and `textinput` to `input`. The first section contains zero-shot results and the second contains finetuned results.

	`click` IoU	`sbmt` IoU	`input` IoU	`say` chrF	`input` chrF	`load` F1	`click` IM	`load` IM	`say` IM	`sbmt` IM	`input` IM
Llama-2-7B	5.11	3.58	2.40	4.25	1.60	18.73	43.08	22.72	34.86	7.47	6.40
Llama-2-13B	9.00	0.40	2.20	1.42	1.81	14.96	75.87	17.53	15.24	1.31	5.33
GPT-3.5T	14.15	4.14	19.95	1.50	15.58	20.78	73.71	24.46	9.04	7.17	33.46
GPT-4T	13.63	2.55	43.11	4.40	34.59	22.83	60.03	29.32	17.46	4.61	68.61
GPT-4V	14.33	3.19	43.72	3.35	33.47	18.21	64.18	21.04	13.55	5.12	65.69
MindAct-250M	18.59	0.00	0.40	14.33	0.18	0.00	89.44	0.00	99.98	0.00	0.52
MindAct-780M	17.08	0.19	20.80	21.25	13.26	8.30	88.16	8.30	100.00	0.69	24.72
MindAct-3B	18.55	13.83	32.73	35.35	19.68	13.85	92.04	19.41	99.97	17.66	39.92
Flan-T5-250M	23.44	0.00	0.00	15.50	0.03	0.00	99.82	0.00	100.00	0.00	0.17
Flan-T5-780M	22.98	0.00	8.13	22.84	5.48	0.00	98.81	0.00	100.00	0.20	12.09
Flan-T5-3B	22.16	31.92	44.57	36.82	31.27	16.22	92.11	22.81	99.97	36.52	52.64
Pix2Act-282M	8.33	0.00	0.00	26.76	1.19	12.91	99.11	18.46	100.00	0.00	1.33
Pix2Act-1.3B	12.82	0.00	0.00	37.78	20.71	21.22	95.52	30.74	100.00	6.10	29.10
Fuyu-8B	18.46	32.69	30.76	33.97	23.03	25.63	92.63	40.93	95.84	43.83	40.92
S-LLaMA-1.3B	23.17	31.02	43.00	37.12	27.87	27.04	94.73	44.12	99.88	41.96	53.35
S-LLaMA-2.7B	24.16	40.86	53.42	38.41	33.54	30.22	94.78	42.85	99.80	44.69	62.41
Llama-2-7B	22.87	50.70	56.64	37.57	36.99	38.46	89.54	55.55	99.79	57.39	65.91
Llama-2-13B	23.13	53.72	56.84	37.66	37.06	42.32	88.17	57.33	99.94	58.28	64.23
GPT-3.5F	18.71	49.39	51.96	31.71	35.60	31.86	83.93	41.50	93.85	51.51	62.63

Table 23: Full TEST$_{\text{CAT}}$ split (test) results. We abbreviate `submit` to `sbmt` and `textinput` to `input`. The first section contains zero-shot results and the second contains finetuned results.

	`click` IoU	`sbmt` IoU	`input` IoU	`say` chrF	`input` chrF	`load` F1	`click` IM	`load` IM	`say` IM	`sbmt` IM	`input` IM
Llama-2-7B	5.45	7.58	0.57	4.06	0.57	17.15	47.89	22.94	39.02	15.15	2.28
Llama-2-13B	10.55	0.00	0.85	1.48	1.01	18.97	77.63	22.02	17.62	3.03	1.71
GPT-3.5T	11.92	0.03	25.20	1.43	15.67	21.57	75.24	24.77	10.14	3.03	33.90
GPT-4T	11.23	0.00	48.03	3.08	35.26	15.61	61.01	23.85	14.97	0.00	67.52
GPT-4V	11.42	1.52	46.01	2.61	33.18	14.07	65.42	17.43	12.87	1.52	65.53
MindAct-250M	15.17	0.00	0.28	12.72	0.55	0.00	83.57	0.00	100.00	0.00	0.57
MindAct-780M	13.62	0.00	27.16	19.78	19.03	9.17	78.36	9.17	100.00	0.00	32.76
MindAct-3B	16.62	27.27	37.34	36.57	24.98	11.17	93.53	15.60	99.93	31.82	40.17
Flan-T5-250M	18.37	0.00	0.00	14.82	0.16	0.00	99.39	0.00	100.00	0.00	0.85
Flan-T5-780M	17.90	0.00	6.01	20.71	3.01	0.00	99.13	0.00	100.00	0.00	7.98
Flan-T5-3B	18.35	34.85	46.44	38.70	31.32	13.46	91.52	19.27	99.93	39.39	51.00
Pix2Act-282M	9.33	0.00	0.00	28.00	3.16	15.52	98.36	19.27	100.00	0.00	3.70
Pix2Act-1.3B	11.80	0.00	0.00	37.21	21.83	15.32	97.60	20.18	100.00	10.61	30.48
Fuyu-8B	15.27	42.52	28.50	34.15	22.85	14.80	94.90	35.78	96.50	48.48	32.48
S-LLaMA-1.3B	18.44	34.85	41.57	38.23	30.14	19.95	95.97	38.53	99.86	43.94	45.87
S-LLaMA-2.7B	20.45	39.48	51.44	37.96	32.84	19.14	95.70	33.03	99.86	42.42	56.41
Llama-2-7B	18.58	42.44	57.08	37.76	36.61	27.01	90.26	46.79	100.00	53.03	61.54
Llama-2-13B	18.12	51.53	57.11	37.00	35.05	31.71	84.98	47.71	100.00	57.58	61.25
GPT-3.5F	15.97	43.94	47.21	29.79	30.27	21.26	85.89	32.11	91.96	45.45	55.27

Table 24: Full TEST$_{\text{GEO}}$ split (test) results. We abbreviate `submit` to `sbmt` and `textinput` to `input`. The first section contains zero-shot results and the second contains finetuned results.

	`click` IoU	`sbmt` IoU	`input` IoU	`say` chrF	`input` chrF	`load` F1	`click` IM	`load` IM	`say` IM	`sbmt` IM	`input` IM
Llama-2-7B	4.21	4.00	2.54	4.45	1.58	14.04	43.35	17.11	34.58	7.00	11.61
Llama-2-13B	7.21	2.00	2.27	1.25	1.50	10.62	77.83	13.16	12.93	2.00	8.75
GPT-3.5T	14.58	4.00	14.98	1.90	15.22	19.97	73.55	25.00	10.24	5.00	33.75
GPT-4T	13.20	4.00	36.16	5.78	26.16	25.32	57.30	30.26	21.43	6.00	69.11
GPT-4V	14.56	5.00	36.09	4.07	26.50	17.86	62.16	21.05	16.00	7.00	65.00
MindAct-250M	16.58	0.00	0.54	18.08	0.01	0.00	86.32	0.00	100.00	0.00	0.71
MindAct-780M	14.74	0.00	19.29	30.93	10.39	7.89	90.73	7.89	100.00	0.00	23.04
MindAct-3B	15.68	7.00	30.40	41.64	17.88	14.05	91.04	23.68	99.94	8.00	34.64
Flan-T5-250M	20.41	0.00	0.00	20.10	0.00	0.00	99.86	0.00	100.00	0.00	0.00
Flan-T5-780M	19.77	0.00	2.37	32.25	1.67	0.00	98.91	0.00	100.00	1.00	4.29
Flan-T5-3B	17.92	25.77	41.82	42.03	27.16	13.17	90.32	25.00	99.94	33.00	49.29
Pix2Act-282M	9.05	0.00	0.00	31.90	0.18	14.49	99.93	21.05	100.00	0.00	0.36
Pix2Act-1.3B	8.80	0.00	0.00	42.42	20.91	13.39	92.82	22.37	100.00	0.00	29.82
Fuyu-8B	14.92	22.46	30.36	35.50	18.87	9.87	86.83	27.63	97.82	36.00	45.36
S-LLaMA-1.3B	18.79	26.29	36.46	41.14	22.89	12.50	90.56	32.89	99.78	44.00	48.93
S-LLaMA-2.7B	18.85	32.00	54.14	41.75	31.52	13.71	91.11	30.26	99.72	32.00	66.25
Llama-2-7B	17.73	51.00	52.21	40.42	32.23	21.91	85.63	43.42	99.78	53.00	64.64
Llama-2-13B	19.68	56.00	52.98	41.87	33.52	29.72	86.45	50.00	100.00	58.00	61.07
GPT-3.5F	14.90	45.00	49.71	35.34	35.53	21.14	81.05	34.21	94.57	45.00	59.64

Table 25: Full TEST_{VIS} split (test) results. We abbreviate `submit` to `sbmt` and `textinput` to `input`. The first section contains zero-shot results and the second contains finetuned results.

	`click` IoU	`sbmt` IoU	`input` IoU	`say` chrF	`input` chrF	`load` F1	`click` IM	`load` IM	`say` IM	`sbmt` IM	`input` IM
Llama-2-7B	4.35	0.01	1.16	4.15	0.87	10.61	38.04	12.86	33.41	3.05	3.85
Llama-2-13B	8.82	0.00	1.15	1.75	0.97	8.19	74.76	11.43	14.70	1.53	4.33
GPT-3.5T	14.93	5.34	16.44	0.53	14.97	10.48	74.03	15.00	4.60	6.11	27.77
GPT-4T	15.04	2.31	44.22	2.96	37.58	16.15	60.67	20.71	13.84	3.05	67.42
GPT-4V	15.26	0.82	44.73	1.66	36.59	13.10	65.03	17.14	8.90	2.29	62.76
MindAct-250M	18.94	0.00	0.16	11.65	0.08	0.00	92.28	0.00	100.00	0.00	0.32
MindAct-780M	17.57	0.00	16.33	15.31	9.80	4.29	89.95	4.29	100.00	0.00	22.47
MindAct-3B	19.36	10.74	29.83	24.41	13.78	9.64	93.01	15.00	100.00	12.98	36.44
Flan-T5-250M	23.12	0.00	0.00	11.70	0.00	0.00	99.89	0.00	100.00	0.00	0.00
Flan-T5-780M	22.77	0.00	6.18	17.04	4.19	0.00	98.84	0.00	100.00	0.00	11.08
Flan-T5-3B	22.86	30.76	42.78	26.32	29.75	11.43	93.92	15.71	99.96	37.40	51.52
Pix2Act-282M	6.86	0.00	0.00	16.60	0.32	6.43	99.68	10.00	100.00	0.00	0.32
Pix2Act-1.3B	12.31	0.00	0.00	25.93	15.44	16.09	96.29	32.86	100.00	6.11	25.52
Fuyu-8B	17.56	22.78	27.10	23.43	18.64	20.12	93.20	37.86	93.64	35.11	36.12
S-LLaMA-1.3B	23.65	24.79	40.19	26.03	20.76	23.71	96.18	47.86	99.85	32.82	51.52
S-LLaMA-2.7B	24.14	35.88	52.27	26.26	30.36	21.55	95.62	37.14	99.81	38.93	59.87
Llama-2-7B	23.23	40.46	58.71	26.72	36.19	34.74	90.65	56.43	99.51	47.33	68.86
Llama-2-13B	23.03	40.46	57.31	27.87	35.02	33.63	88.98	50.71	99.89	47.33	64.04
GPT-3.5F	17.97	43.51	50.27	22.99	32.31	29.91	84.09	39.29	91.32	47.33	59.39

Table 26: Full TEST_{WEB} split (test) results. We abbreviate `submit` to `sbmt` and `textinput` to `input`. The first section contains zero-shot results and the second contains finetuned results.

	`click` IoU	`sbmt` IoU	`input` IoU	`say` chrF	`input` chrF	`load` F1	`click` IM	`load` IM	`say` IM	`sbmt` IM	`input` IM
Llama-2-7B	5.33	0.47	2.78	4.27	0.41	22.38	42.92	24.05	35.12	5.26	3.79
Llama-2-13B	8.98	0.00	1.77	1.36	0.76	16.47	73.48	17.72	16.04	0.00	3.03
GPT-3.5T	12.41	1.71	21.46	1.88	15.21	33.00	72.46	34.18	11.43	7.89	31.57
GPT-4T	12.75	2.97	45.83	5.64	36.43	38.15	61.56	41.77	18.83	10.53	63.89
GPT-4V	13.03	1.71	45.12	4.19	36.06	30.47	65.26	32.91	15.21	7.89	63.89
MindAct-250M	16.79	0.00	1.01	14.68	0.26	0.00	92.90	0.00	99.92	0.00	1.01
MindAct-780M	15.09	0.00	21.86	19.99	14.69	10.13	91.44	10.13	100.00	0.00	23.23
MindAct-3B	16.50	0.00	35.63	38.93	19.97	17.72	92.99	22.78	100.00	7.89	38.64
Flan-T5-250M	21.79	0.00	0.00	15.64	0.00	0.00	99.95	0.00	100.00	0.00	0.00
Flan-T5-780M	21.78	0.00	10.55	21.60	6.36	0.00	98.54	0.00	100.00	0.00	13.89
Flan-T5-3B	20.48	19.84	49.79	39.59	33.80	18.57	92.51	24.05	100.00	21.05	55.05
Pix2Act-282M	9.57	0.00	0.00	30.30	2.27	14.77	97.71	25.32	100.00	0.00	2.27
Pix2Act-1.3B	13.24	0.00	0.00	39.57	23.63	19.20	95.33	31.65	100.00	0.00	32.58
Fuyu-8B	18.44	13.50	29.89	34.95	24.58	17.24	94.26	36.71	96.88	23.68	37.63
S-LLaMA-1.3B	22.46	11.58	46.89	38.14	28.92	26.78	95.04	37.97	99.92	13.16	53.03
S-LLaMA-2.7B	22.61	21.05	51.99	40.77	33.68	27.62	95.67	40.51	99.92	34.21	62.12
Llama-2-7B	21.11	36.84	52.23	37.74	35.99	35.20	88.76	54.43	99.84	47.37	64.90
Llama-2-13B	22.58	44.74	52.14	38.03	35.93	39.13	89.98	58.23	99.84	52.63	61.87
GPT-3.5F	17.91	42.11	50.68	33.88	37.47	41.20	83.89	51.90	94.41	47.37	68.69

Table 27: Element Group (EG), Text Group (TG) and overall results for $\textsc{Test}_{\textsc{iid}}$ (left) and $\textsc{Test}_{\textsc{ood}}$ (right) splits. The top section contains zero-shot results and the bottom contains finetuned results.

	Overall Micro Avg	Overall IM	EG IoU	TG F1	Overall Micro Avg	Overall IM	EG IoU	TG F1
Llama-2-7B	5.32	33.80	4.01	3.06	4.33	33.92	3.17	2.33
Llama-2-13B	5.61	42.85	5.29	1.97	5.28	43.51	4.93	1.44
GPT-3.5T	10.35	42.42	10.68	3.98	8.89	42.70	9.05	3.56
GPT-4T	12.24	42.69	12.55	7.85	11.05	41.87	11.21	6.97
GPT-4V	12.99	42.47	13.68	7.28	10.98	42.38	11.49	6.43
MindAct-250M	16.88	76.54	18.01	8.46	13.48	74.71	13.24	7.83
MindAct-780M	19.61	78.06	20.12	14.04	16.03	76.31	14.75	13.68
MindAct-3B	25.71	80.99	22.50	24.50	21.90	80.11	17.70	23.43
Flan-T5-250M	20.93	80.28	23.68	9.51	16.18	79.81	16.62	9.27
Flan-T5-780M	23.71	81.91	25.35	16.17	18.56	80.40	17.36	14.47
Flan-T5-3B	31.12	83.48	29.56	29.06	25.25	81.61	22.18	26.41
Pix2Act-282M	12.30	80.50	4.86	17.29	12.47	79.87	5.93	16.58
Pix2Act-1.3B	23.91	83.42	13.15	32.59	18.44	82.12	9.36	26.69
Fuyu-8B	30.92	84.51	25.73	33.66	22.29	80.96	17.85	24.57
S-LLaMA-1.3B	33.99	87.81	32.41	34.68	25.94	84.22	23.11	27.62
S-LLaMA-2.7B	37.43	87.70	35.54	37.66	27.61	84.74	25.33	29.27
Llama-2-7B	38.12	88.08	36.71	38.58	27.51	83.73	25.46	28.91
Llama-2-13B	37.09	87.70	35.92	37.43	27.86	83.07	25.79	28.77
GPT-3.5F	30.89	82.34	30.22	29.62	23.35	78.52	21.19	23.84

Table 28: Element Group (EG), Text Group (TG) and overall results for $\textsc{Test}_{\textsc{cat}}$ (left) and $\textsc{Test}_{\textsc{geo}}$ (right) splits. The top section contains zero-shot results and the bottom contains finetuned results.

	Overall Micro Avg	Overall IM	EG IoU	TG F1	Overall Micro Avg	Overall IM	EG IoU	TG F1
Llama-2-7B	4.57	38.32	3.46	2.18	3.61	33.48	2.60	2.11
Llama-2-13B	6.50	47.52	6.03	1.59	4.03	43.04	3.87	1.09
GPT-3.5T	8.23	45.91	8.42	3.35	8.86	42.09	8.78	3.66
GPT-4T	9.48	42.14	9.90	5.63	10.61	40.86	10.53	6.38
GPT-4V	9.26	43.66	9.80	5.30	10.74	41.33	11.05	5.86
MindAct-250M	11.69	72.93	11.27	6.61	13.15	70.25	11.20	8.93
MindAct-780M	14.36	72.83	12.85	12.37	16.99	74.48	12.39	17.68
MindAct-3B	21.60	81.70	16.59	25.01	21.42	76.00	14.65	24.75
Flan-T5-250M	13.96	81.26	13.61	8.98	15.56	76.63	13.70	11.15
Flan-T5-780M	15.61	81.64	13.86	12.93	18.92	76.58	13.58	18.02
Flan-T5-3B	23.67	81.72	18.99	26.85	23.52	77.46	18.09	26.52
Pix2Act-282M	13.31	81.34	6.95	17.95	13.77	76.96	6.10	18.20
Pix2Act-1.3B	17.44	84.03	8.93	26.64	16.96	77.84	6.07	26.72
Fuyu-8B	20.42	82.29	15.03	24.47	19.53	75.87	14.58	21.45
S-LLaMA-1.3B	23.76	84.72	18.82	28.39	23.48	78.64	18.19	25.82
S-LLaMA-2.7B	25.06	85.08	21.38	28.39	24.62	80.04	20.60	27.50
Llama-2-7B	24.57	83.65	20.86	27.96	24.38	78.78	20.54	26.50
Llama-2-13B	24.27	81.00	20.72	26.12	25.93	78.62	21.97	27.67
GPT-3.5F	20.21	78.07	17.31	21.16	21.94	74.69	17.97	23.91

Table 29: Element Group (EG), Text Group (TG) and overall results for TEST_{VIS} (left) and TEST_{WEB} (right) splits. The top section contains zero-shot results and bottom contains finetuned results.

	Overall Micro Avg	Overall IM	EG IoU	TG F1	Overall Micro Avg	Overall IM	EG IoU	TG F1
Llama-2-7B	3.77	30.99	2.50	2.10	4.35	33.03	3.26	2.19
Llama-2-13B	5.06	42.05	4.60	1.34	5.21	42.11	4.86	1.23
GPT-3.5T	8.58	40.14	8.84	2.63	8.42	42.95	8.51	4.16
GPT-4T	11.14	40.38	11.65	6.36	11.77	43.28	11.43	8.62
GPT-4V	10.73	40.45	11.59	5.72	11.20	44.00	11.35	7.96
MindAct-250M	13.16	79.07	13.97	7.26	12.54	74.76	11.75	7.89
MindAct-780M	14.46	79.81	14.97	10.87	14.74	76.36	13.40	13.42
MindAct-3B	19.11	82.98	18.46	17.81	21.64	78.90	16.28	25.07
Flan-T5-250M	15.18	82.71	17.02	7.80	15.25	78.15	15.09	8.93
Flan-T5-780M	17.09	83.09	17.53	11.90	17.48	78.78	16.49	13.33
Flan-T5-3B	22.91	84.92	23.01	21.73	25.06	80.47	21.25	27.91
Pix2Act-282M	9.16	82.81	5.06	11.39	13.79	77.75	6.67	18.07
Pix2Act-1.3B	15.33	84.63	9.27	20.18	18.57	80.69	9.39	27.31
Fuyu-8B	18.63	82.29	16.83	18.73	21.97	79.85	17.10	24.57
S-LLaMA-1.3B	22.80	86.83	23.49	21.32	25.68	83.10	22.66	27.86
S-LLaMA-2.7B	24.12	87.29	25.81	22.79	26.82	83.60	23.31	30.01
Llama-2-7B	24.70	86.56	26.36	23.78	25.80	81.57	22.81	27.74
Llama-2-13B	25.00	85.31	26.09	23.89	27.00	82.72	24.24	28.72
GPT-3.5F	20.46	79.37	20.49	19.36	23.24	78.13	19.95	25.13

E INSTRUCTIONS FOR THE ANNOTATORS

PROJECT INFORMATION

We are collecting data for **evaluating** automated web navigation systems. The data consists of **demonstrations** of interactions between the user and the navigator.

In each demonstration, the user and the system cooperate to achieve **tasks in a web browser.** The user controls the system via **natural language instructions**.

HOW TO

INGREDIENTS

- **two people:**
  - **Instructor:** creative, giving instructions
  - **Navigator:** systematic, following instructions
- Google Chrome
- Zoom
- internet connection

PREPARATION

You need to do this process just once:

1. Download the Chrome extension ZIP file and unpack the *extension* folder to your local filesystem.
2. If you are using Chrome as your primary browser, create a new profile for the experiments.
3. Install the Chrome extension in the repository:
   - Open a new Google Chrome window.
   - Go to chrome://extensions/
   - At the top right, turn on Developer mode.
   - Click Load unpacked.
   - Find and select the *extension* folder you have unpacked before (make sure you are inside the folder).
   - Click on the "puzzle" icon in the task bar with Chrome extensions and pin this extension.
4. Setup Zoom:
   - Open Zoom and **log in**.
   - Go to https://zoom.us/profile/setting
   - On the Meeting tab, turn on *Auto saving chats (learn more here).*
   - On the Recording tab:
     i. enable Local Recording
     ii. enable *"Hosts can give meeting participants permission to record locally"*.
     iii. enable automatic recording on a local computer
   - **Setup your Zoom name to *Instructor* or *Navigator* according to your role.**

UPDATING THE EXTENSION

Check regularly if you are using an up-to-date version of the extesion:

- The current version can be found at the top of this document.
- Your version is at chrome://extensions/ next to the extension name.

If there is a never version of the extension, remove the extension and repeat points 1) and 3) in the Preparation section.

DEMONSTRATIONS

1. **Navigator** calls **Instructor** via Zoom (Participants → Invite)
   - Ensure that both have **video and microphone are disabled**.
2. After the call is accepted:

- **Instructor** opens a Zoom chat window,
- **Navigator**:
  - opens a Zoom chat window,
  - opens a Chrome window,
  - shares the screen with their Chrome window **(only),**
  - starts recording a Zoom call video (ignore the warning about audio).

3. **Navigator** clicks on the extension button in the navigation bar and selects **New recording**.
   - A new tab will open with an overlay *Starting recording* for 1 second (make sure that it is visible on the Zoom recording), followed by a prompt for waiting for instructions.
   - **Use the opened tab, do not open any new tab!**
4. **Instructor** gives **Navigator** instructions through the chat interface for accomplishing a task (see Tasks for details).
   - **Instructor** has no other way of communicating with **Navigator** than **through the chat interface**.
   - **Instructor** can give intermediate instructions or answer system questions.
5. **Navigator** performs actions in the web browser according to **Instructor**'s instructions.
   - **Navigator** should use the **chat interface** to ask the user for any missing details and to provide answers if necessary.
6. After the task is finished, **Navigator**:
   - clicks on the extension button, selects *Save recording* and **wait until the recording gets saved to their computer**,
   - stops the video recording and screen sharing,
   - ends the call,
   - submits the recording (see Recording for details).

Recording

The recording is submitted through the web interface.

The recording consists of:

- a "<recording_id>.zip" file, which is a ZIP archive with:
  - metadata,
  - events,
  - screenshots,
  - HTML snapshots,
- Zoom chat history "`meeting_saved_chat.txt`",
- Zoom invite link

The Zoom recording folder depends on your platform. The default directories are:

- **Windows**: `C:\Users\[Username]\Documents\Zoom`
- **Linux**: `/home/[Username]/Documents/Zoom`
- **Mac**: `/Users/[Username]/Documents/Zoom`

Actions

Navigator can perform the following actions in the browser:

- **go to a URL** through the navigation bar,
- **click** on an element,
- **input text** into an input field,
- **scroll** up and down the page,

The actions which should **not be performed**:

- opening a new tab (it is ok if the page opens a tab by itself),
- horizontal scrolling,
- page search (Ctrl+F),
- keyboard shortcuts,
- drag & drop (e.g. Google Maps)

TASKS

Instructor can give the system any tasks which an automated web assistant should be able to handle. Use your imagination!

The tasks **can be unspecified at first**. It is the job of the system to ask for intermediate details throughout the tasks demonstration.

Stop the demonstration before doing any real action in the world: booking a table, buying a ticket, etc.

WEBSITES

For your inspiration, here is a spreadsheet with the **list of websites** and the task categories you can use them for.

We have created a shared account for these websites which you should use in case you need to login.

Of course feel free to use any other websites (just do not fill in any other personal details there, preferably use the shared account as well).

TIPS

Navigator

- **Don't do things too quickly!** Saving the actions, screenshots and pages takes time and performing the actions in a quick succession can introduce errors in the recording, especially on heavy websites.
  Watch for the icon indicating that the browser is processing an action.
- Do not perform any **unnecessary actions** *(all the actions will be recorded and we want to minimize the amount of mindless clicking and scrolling)*
- Wait until the page **fully loads**.
- **Do not use autofill** for text fields, always type everything from scratch.
- Do not change the **size of the browser window** if not necessary.

Instructor

- **Be creative**: assign tasks starting from very simple ("submit the form") to very complex (multi-turn conversation with changing topics).
- Ask only about things that are relevant to the webpage.
- **Wait** until the system performs their actions.
  - However, feel free to interrupt if something does not seem right or you have changed your mind.
- Finalize all the tasks right **before changing the actual state of the worl**d (i.e. ordering products, submitting issues etc.).

Note that the extension does not work in an anonymous window. If you want to clear your history afterwards, use Ctrl+Shift+Delete.