GuideWeb: A Benchmark for Automatic In-App Guide Generation on Real-World Web UIs

Anonymous ACL submission

Abstract

Digital Adoption Platforms (DAPs) provide web-based overlays that deliver operational guidance and contextual hints to help users navigate complex websites. Although modern DAP tools enable non-experts to author such guidance, maintaining guides remains labor-intensive because website layouts and functionalities evolve continuously. In this work, we introduce **GuideWeb**, a human-verified benchmark for automatic in-app guide proposal on real-world web UIs. GuideWeb formulates the task as selecting **guide target elements** grounded in the webpage and generating concise guide text aligned with user intent. We make the target audience and guideworthiness criteria explicit, and construct annotations through LLM-assisted candidate discovery followed by human verification, including deletion, rewriting, grounding correction, and addition of missed targets. We also propose an evaluation suite that jointly measures guide target selection, DOM grounding, and generated intent/guide-text quality. Experiments show that our proposed **GuideWeb Agent** achieves **30.79%** F1 in guide target prediction, while obtaining BLEU scores of **44.94** for intent generation and **21.34** for guide-text generation. Existing baselines perform substantially worse, highlighting that reliable automatic guide proposal remains challenging on real-world web UIs.

1 Introduction

In recent years, rapid progress in large language models (LLMs) (OpenAI, 2025; Anthropic, 2025; Comanici et al., 2025) has accelerated research on LLM-based agents that plan and act to achieve user goals (Huang et al., 2024). Among them, web agents (Ning et al., 2025) have attracted growing attention for automating website interactions and completing complex tasks. However, while many benchmarks focus on action-oriented web agents, a widely deployed form of web assistance remains underexplored: in-app guidance for helping users understand and operate unfamiliar web interfaces, as commonly provided by digital adoption platforms (DAPs).

Figure 1: Illustration of a digital adoption platform (DAP) as an in-app overlay for unfamiliar web UIs. **Top:** without guidance, the user cannot confidently identify the correct guide target element to proceed. **Bottom:** with a DAP overlay, the webpage is augmented with contextual hints and step-by-step instructions, enabling the user to complete the action efficiently.

DAPs are typically browser-based overlays that augment web services with contextual hints and step-by-step instructions. As illustrated in Figure 1, users often struggle to identify the correct UI ele-

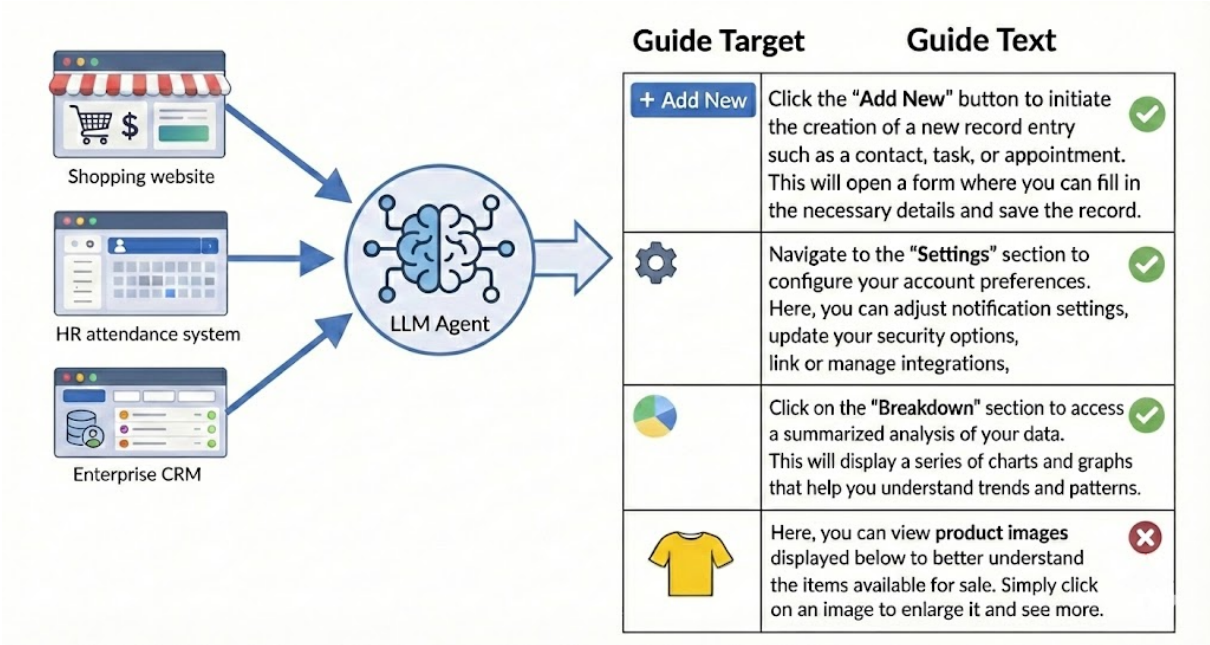

Figure 2: Overview of GUIDEWEB. Given the main page of a real-world website, an LLM-based agent identifies guide targets, namely interactive UI elements whose usage may benefit from guidance, and generates corresponding guide text grounded in visible on-page content. The examples on the right illustrate both correct guides and typical failure cases, where the agent produces low-utility guidance for elements that are already self-explanatory.

ments on unfamiliar or complex webpages, especially in enterprise systems with dense functionality and limited visual cues. This increases learning cost, reduces usage efficiency, and raises customer support burden. By annotating key workflows directly on webpages, DAPs help users locate and understand essential operations more efficiently.

Despite their practical importance, existing DAP solutions rely heavily on manual authoring. Human experts must identify guide-worthy UI elements and write corresponding instructional text. This process is costly and brittle because web layouts and functionalities evolve continuously: layout changes require re-locating guide targets, newly introduced features require additional guide creation, and outdated guides may mislead users. We therefore study automatic guide *proposal* as a human-in-the-loop component of DAP maintenance, where a model suggests candidate targets and draft guide texts while human authors remain responsible for final deployment. This setting is complementary to source-level instrumentation: directly embedding native guides can be robust when developers own the application, whereas overlay-based proposals are applicable to legacy, third-party, or frequently changing web services where DAP authors may not control the underlying code.

We address this gap by introducing GUIDEWEB, a benchmark for automatic web guide proposal grounded in real-world web UIs, as illustrated in Figure 2. GUIDEWEB focuses on main pages of diverse websites and formulates guide generation as a two-stage task: identifying which interactive UI elements should be guided and generating concise guide text aligned with user intent. This formulation captures a central challenge of in-app guidance: not all interactive elements require explanation, and effective guides must be both selective and informative. We define guide-worthy elements as salient interactive controls that support common user goals, are non-trivial for first-time users, or reduce friction in important workflows.

In addition to the benchmark, we train a lightweight GUIDEWEB AGENT tailored to this task. Our contributions are: (1) a task definition for selective, element-grounded in-app guide proposal; (2) a 996-page benchmark with explicit annotation criteria and human verification; (3) an evaluation suite for target selection, grounding, and text quality; and (4) an efficient agent with a deterministic context-shortening mechanism. Experiments show that the trained agent substantially outperforms strong general-purpose baselines, while also revealing that reliable guide proposal remains far from solved.

2 Related Work

Web Agents and Web Interaction Benchmarks. Recent advances in large language models have accelerated research on web agents that autonomously operate websites to accomplish user goals. Early benchmarks emphasize controlled and templated environments, such as Liu et al. (2018), which enables reproducible evaluation on synthetic web interfaces. More recent benchmarks improve realism and scale: Yao et al. (2022) introduces shopping tasks grounded in structured product pages, Deng et al. (2023) collects real-world trajectories across diverse websites and user intents, and Zhou et al. (2023) hosts interactive websites requiring multi-step reasoning and form-based interactions. Multimodal extensions such as Koh et al. (2024) further incorporate visual perception to better model real browsing behavior. These benchmarks mainly study how agents perceive webpages and execute actions to complete tasks. In contrast, our work addresses a complementary problem: generating in-app guidance artifacts that help end users understand how to operate unfamiliar web interfaces.

In-App Guidance and UI-Level Assistance. Beyond action execution, prior research in human-computer interaction and intelligent assistance has explored contextual guidance over existing interfaces. Systems such as Zhong et al. (2021) generate visual tutorials aligned with UI states, while approaches like Li et al. (2017) infer task procedures from user demonstrations and UI signals. Other studies examine interactive walkthroughs, tooltips, and overlay-based explanations for reducing the learning cost of complex software. However, these works typically focus on mobile applications, scripted tutorials, or automation by demonstration, rather than web-scale guide generation as a benchmarked learning task. Existing web agent benchmarks also do not evaluate whether an agent can identify which UI elements require guidance or generate useful element-grounded explanations. GUIDEWEB fills this gap by introducing a real-world web UI benchmark that decomposes automatic guide generation into guide target identification and guide text generation, with an evaluation protocol for both stages.

3 GuideWeb: A Benchmark for Automatic In-App Guide Generation on Real-World Web UIs

We introduce GUIDEWEB, a benchmark designed to study automatic in-app guide proposal on real-world web user interfaces. GUIDEWEB focuses on the main pages of diverse websites and aims to evaluate whether an agent can identify interaction points that require guidance and generate appropriate guide text grounded in visible on-page content. Unlike prior web agent benchmarks that emphasize long-horizon task completion, GUIDEWEB targets a complementary but practically important problem: assisting users in understanding and operating unfamiliar web interfaces through lightweight, contextual guidance.

3.1 Task Definition and Output Schema

Target user and task scope. GUIDEWEB targets first-time or novice users who understand the general domain of a website but are unfamiliar with the specific interface. The goal is not to annotate every clickable object, but to propose a compact set of useful guide overlays for salient interactive elements. We treat an element as guide-worthy when it satisfies at least one of the following criteria: (i) it supports a common user goal such as search, login, signup, checkout, pricing, or support; (ii) its function is important but not fully self-explanatory from visible text, icon, layout, or surrounding context; or (iii) it serves as an entry point to a workflow where brief contextual guidance can reduce user uncertainty. Purely decorative links, redundant navigation items, and controls whose purpose is immediately obvious are excluded. The same criteria are used in the annotation prompt, the human verification protocol, and baseline inference prompts; full guidelines are summarized in Appendix D.

Input Representation. Given the main page of a website, we represent the input as the raw HTML source

$$x \in \mathcal{X}, \tag{1}$$

from which we construct a DOM tree and extract a set of interactive elements

$$\mathcal{E}(x) = \{e_1, e_2, \ldots, e_N\}. \tag{2}$$

Each element $e \in \mathcal{E}(x)$ is associated with observable attributes

$$\phi(e) = (\texttt{tag}(e), \texttt{visible_text}(e), \texttt{xpath}(e)). \tag{3}$$

The released task is therefore DOM-grounded: systems must return concrete webpage elements rather than only free-form page descriptions. During dataset construction, however, human reviewers inspect the browser-rendered page together with the DOM index, so final labels are not accepted from HTML-only LLM outputs without visual and structural checking.

Output and Task. The goal is to generate a structured guide annotation

$$y = (g, \mathcal{A}), \tag{4}$$

where $g \in \{0, 1\}$ indicates whether the page requires any guidance, and $\mathcal{A}$ is a set of element-grounded guide annotations.

Formally, a GUIDEWEB system is a mapping

$$f : \mathcal{X} \to \{0, 1\} \times \mathcal{Y}, \quad f(x) = (g, \mathcal{A}). \tag{5}$$

Stage 1: Guide Target Identification. Conditioned on the page x and its interactive set $\mathcal{E}(x)$, the model selects a subset of elements that require guidance:

$$\mathcal{E}^+ = S(x) \subseteq \mathcal{E}(x). \tag{6}$$

Elements in $\mathcal{E}^+$ are referred to as *guide targets*. If the model predicts $g = 0$, we define $\mathcal{E}^+ = \emptyset$ and $\mathcal{A} = \emptyset$.

Stage 2: Element-Grounded Guide Generation. For each guide target $e \in \mathcal{E}^+$, the model generates a structured annotation

$$a(e) = (i(e), t(e), s(e), \phi(e)), \tag{7}$$

where $i(e)$ is a natural-language intent, $t(e)$ is an action type (e.g., search, navigation, selection), and $s(e)$ is the guide text describing how to use the element. The tuple explicitly carries $\phi(e)$ so that the annotation is grounded in the original page x.

The final output set is

$$\mathcal{A} = \{a(e) \mid e \in \mathcal{E}^+\}. \tag{8}$$

This formulation jointly captures *where* guidance is needed (target selection via $S(x)$) and *what* guidance should be provided (element-conditioned generation of $a(e)$), while grounding all outputs in the input webpage through `xpath` and visible content.

Output Schema. Each annotated webpage in GUIDEWEB is stored in a structured JSON format. The schema consists of page-level fields and a list of guide-level annotations, summarized in Appendix Table 6. Each webpage is stored in a dedicated subdirectory containing the raw HTML file and a JSON annotation file. At the page level, the JSON records whether the webpage requires guidance and includes a coarse page category. At the guide level, each annotation stores an intent, action type, guide text, HTML tag, visible text when available, and XPath. The grounding fields enable downstream DAP systems to attach overlays to concrete DOM nodes, including icon-only controls whose function must be inferred from structure or nearby context. The intent field makes explicit why a guide is needed and what user goal it supports.

```
{
  "intent": "How do I log in to my account?",
  "action_type": "login",
  "guide_text": "Choose "Login" to go to the sign-in
page.",
  "tag": "a",
  "visible_text": "Login",
  "xpath": "//a[@href='https://app.benevity.com/
login']"
}
```

Figure 3: A single guide annotation in GUIDEWEB, containing intent, action type, guide text, and DOM grounding fields.

As illustrated in Figure 3, all of the above information is unified within a single JSON file, allowing models to generate, store, and consume annotations in a consistent and structured format.

3.2 Dataset Construction Pipeline

During dataset construction, we adopt a hybrid strategy that uses LLMs only for candidate discovery and humans to determine the final gold labels. The overall pipeline is illustrated in Figure 4. This design scales annotation over many interactive elements while avoiding treating raw LLM outputs as benchmark ground truth.

The websites in GUIDEWEB are sampled from the Cisco Umbrella Popularity List,[1] which ranks domains using aggregated passive DNS usage across Umbrella's global network. Unlike browser-centric rankings such as Alexa, the Umbrella list

[1] https://umbrella-static.s3-us-west-1.amazonaws.com/index.html

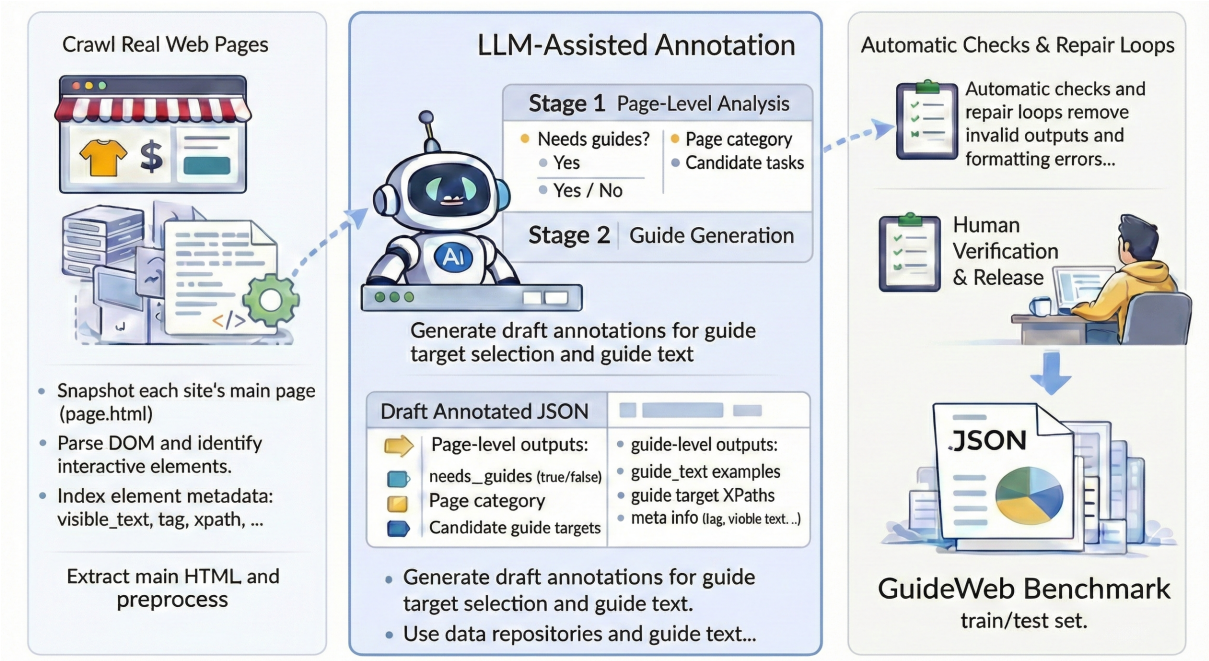

Figure 4: Overview of the GUIDEWEB construction pipeline with LLM-assisted annotation and human verification.

reflects Internet-wide activity across diverse protocols and applications, making it suitable for collecting a broad set of real-world web services. We begin with the Top 1M domains[2] and randomly sample candidates. For each domain, we use an automated browser to crawl and snapshot its main landing page, retaining only pages with sufficient interactive elements such as clickable controls, input fields, and forms. After filtering, we obtain 1,000 real-world websites spanning e-commerce platforms, enterprise systems, and general service websites.

For each website, the main page serves as the annotation target. We store the HTML snapshot, render the page in a browser, and parse the DOM to identify interactive elements. The crawler records page complexity statistics such as links, buttons, inputs, forms, scripts, and text length; pages that fail to load or expose too few meaningful controls are excluded before annotation. Each retained element is indexed in a unified `elements index`, which records its raw HTML tag (`tag`), visible text (`visible_text`), and XPath (`xpath`) for localization. Icon-only elements without visible text are indexed using structural attributes and nearby context. This index grounds all subsequent guide annotations to concrete DOM nodes.

The processed page is then provided to an LLM for *candidate* annotation. The LLM first proposes whether the page contains interaction points that may merit guidance and assigns a coarse-grained category. It then proposes candidate guide targets, intents, action types, and guide texts in the predefined JSON schema. The prompt follows the same target-user assumption and inclusion/exclusion criteria given to human reviewers: prioritize high-utility workflow entry points and avoid self-explanatory, decorative, or redundant controls. A prompt template is provided in Appendix E.

The guide text serves as an answer-like explanation, while the intent captures the implicit question a user may have when interacting with the page. If the model output does not conform to the required format, an automatic regeneration and repair loop is triggered until a valid structured output is obtained.

Human verification is then applied to the candidate JSON as the final annotation stage. Reviewers inspect the browser-rendered webpage, indexed DOM fields, and candidate annotations under the same operational criteria. They perform four operations: (1) **validate** correct guide targets and texts, (2) **delete** candidates attached to decorative, redundant, or self-explanatory elements, (3) **revise** intents, action types, guide texts, and grounding fields when the target is useful but the wording or XPath is inaccurate, and (4) **add** missing guide-worthy elements not proposed by the LLM. This

[2] http://s3-us-west-1.amazonaws.com/umbrella-static/top-1m.csv.zip

Statistic	Value	Statistic	Value
Total crawled websites	1,000	Websites removed after verification	4
Valid annotated websites	996	Websites requiring guides (`needs_guides=true`)	980
Websites without guides	16	Ratio requiring guides	98.4%
Average guides per page	3.09	Pages reaching guide cap (5 guides)	546 (54.8%)

Table 1: Overall statistics of the GUIDEWEB benchmark (guide-level annotations only).

add operation explicitly checks for LLM omissions rather than assuming the candidate pool is complete. The five-guide cap should therefore be interpreted as a top-k authoring budget for the most useful guides, not as a claim that no additional useful guides could exist. Pages with severe structural errors, degenerate element indices, or unresolved ambiguity are removed. Through verification, we remove four pages and obtain 996 valid annotated webpages, which are split into training and test sets with a 7.5:2.5 ratio used across all experiments.

3.3 Dataset Statistics

Table 1 reports high-level benchmark statistics. Starting from 1,000 crawled domains, we obtain 996 validated main-page snapshots after human verification. Most pages are guide-worthy (`needs_guides=true`), reflecting that real-world homepages typically expose multiple interactive entry points that benefit from lightweight guidance. Each page contains 3.09 guide annotations on average, and over half reach the five-guide budget. We introduce this cap to approximate practical DAP design, where excessive overlays can harm usability and visual clarity; annotators therefore prioritize valuable, frequent, and user-critical interactions. The cap also makes GUIDEWEB a top-k guide-authoring benchmark: predictions outside the capped gold set may still be useful but are counted as false positives, a limitation discussed in Section 8. Appendix B reports page-category statistics, showing variation in guide density and necessity across categories.

Table 2 summarizes the distribution of guide annotations by action type. The benchmark is dominated by information-seeking and navigation behaviors such as `search` and `navigate`, followed by account and support workflows such as `login`, `signup`, and `contact_support`. Long-tail categories are aggregated into `other` to stabilize analysis and avoid over-interpreting sparse types. A detailed breakdown of page categories and guide necessity is provided in Appendix B.

Action type	# Guides
search	728
navigate	520
login	412
contact_support	307
other (incl. <50 types)	296
signup	199
subscribe_newsletter	112
start_trial	107
checkout	93
pricing	86
filter_sort	79
download_install	75
settings_profile	64

Table 2: Distribution of guide annotations by action type (guide-level counts). Rare types with fewer than 50 guides are merged into `other`.

4 GuideWeb Agent

In addition to the benchmark, we train a lightweight GUIDEWEB AGENT on the training set for automatic in-app guide proposal. The agent uses Qwen3-8B as the backbone and is supervised to generate the same structured JSON schema as the benchmark annotations. Unlike a browser-operating web agent, it does not click, browse, call tools, or access external webpages at inference time; it receives a fixed preprocessed page representation and emits JSON guide proposals.

To handle long and noisy webpages, we apply a deterministic *Shorter* mechanism before inference. Shorter removes scripts, styles, boilerplate, and long non-interactive regions; normalizes visible text; keeps page-level excerpts and headings; extracts candidate interactive elements from native controls, links, forms, ARIA roles, and click handlers; and preserves visible text plus XPath-based grounding cues for each candidate. Detailed budgets and truncation rules are provided in Appendix G. This reduces input length and computational cost while preserving information needed for guide target identification and guide text generation.

Empirical results in Section 6 show that *Shorter* does not degrade performance and can improve it. These findings suggest that guide-relevant information is concentrated in interactive elements, visible texts, and nearby contextual descriptions,

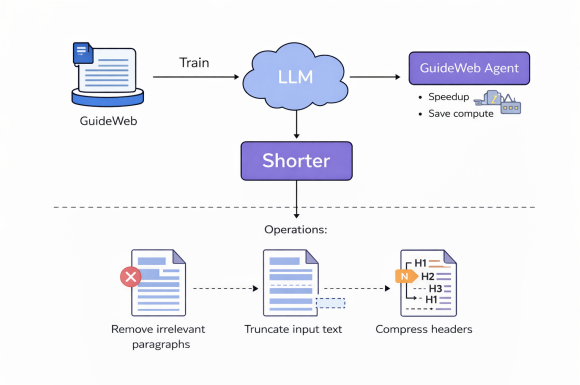

Figure 5: Overview of the GUIDEWEB AGENT with the *Shorter* mechanism. The agent is trained on the GUIDEWEB benchmark to perform automatic in-app guide proposal. During inference, the *Shorter* module reduces input length by removing irrelevant content, truncating long text, and compressing headers, enabling faster inference and reduced computational cost while preserving guide-relevant information.

while much raw webpage text and auxiliary HTML structure contributes little. This supports compact, interaction-focused representations for efficient in-app guide generation.

5 Experiments

5.1 Evaluation Metrics

We evaluate GUIDEWEB from three complementary aspects: (1) **guide target selection**, i.e., which UI elements should be annotated; (2) **text generation quality**, covering intent and guide text; and (3) **structured grounding accuracy**, measured by field-level exact matches. Although the underlying measures are standard, we tailor their targets to the GUIDEWEB setting.

Guide target selection (P/R/F1). For each page, let G be the set of gold guide targets and $\hat{G}$ the predicted set, where each target corresponds to a unique DOM element. We report precision, recall, and F1 over set overlap (definitions in Appendix F). Precision measures how many predicted targets are truly guide-worthy, recall measures how many gold targets are recovered, and F1 summarizes the trade-off. This metric captures *selective* guide authoring by penalizing redundant or low-utility annotations.

Intent and guide-text generation (BLEU, ROUGE-L). For matched guide targets, we compare generated intents and guide texts against references using BLEU and ROUGE-L. These metrics quantify lexical overlap and sequence-level similarity, reflecting whether the agent produces concise, faithful descriptions aligned with the intended interaction. Since multiple semantically valid guide texts may exist for the same element, we treat text-overlap metrics as diagnostic rather than complete measures of user-perceived guide quality.

Structured field correctness (Exact-match F1). We further evaluate grounding by computing exact-match F1 for key fields: `action_type`, `tag`, `visible_text`, and `xpath`. A field value is correct only if it exactly matches the reference. This provides fine-grained diagnostics: `xpath` tests precise localization, while `tag` and `visible_text` reflect structural and surface alignment.

5.2 Baselines

We evaluate GUIDEWEB with representative large language models covering both proprietary and open-source systems. We include GPT-5 (OpenAI, 2025), Claude Sonnet 4.5 (Anthropic, 2025), and Gemini 2.5 Pro (Comanici et al., 2025) as strong closed-source baselines, and Qwen3-8B (Yang et al., 2025) and LLaMA 3.1-8B (Grattafiori et al., 2024) as open-source counterparts.

All baselines are prompted to perform guide target identification and guide text generation directly from webpage content using a unified prompt template, without task-specific fine-tuning. This reflects a realistic zero-shot or instruction-following deployment setting and isolates the inherent capability of general-purpose LLMs on GUIDEWEB. Since no prior benchmark defines the same element-grounded guide proposal task, these baselines are not exhaustive future modeling choices; rather, they establish how strong generalist systems perform before task-specific supervision. We view retrieval baselines, accessibility-tree variants, and supervised element-selection-only models as important future comparisons once the task is adopted more broadly.

6 Results

As shown in Table 3, we evaluate all baseline models and the proposed GuideWeb Agent using the metric suite introduced in Section 5.1. We first focus on guide target element selection. The GuideWeb Agent achieves the highest F1 score among all models, indicating a substantially better balance between precision and recall. We intentionally report precision/recall/F1 rather than a single "accuracy" number because the task is a capped

Table 3: Main results on GuideWeb. Left: guide target element selection. Right: intent and guide-text generation quality.

	(a) Guide target element selection				(b) Text generation quality			
Model	**P**	**R**	**F1**	**Match/Pred.**	**Intent BLEU**	**Intent ROUGE-L**	**Guide BLEU**	**Guide ROUGE-L**
GPT-5	15.29	54.69	23.90	356/2328	13.84	24.99	4.30	11.69
Claude-Sonnet-4.5	15.99	**58.99**	25.16	384/2402	15.88	25.99	1.48	8.98
Gemini-2.5-Pro	14.43	54.84	22.85	357/2474	16.53	30.54	2.53	13.43
Qwen3-8B	9.11	12.33	10.48	72/790	3.69	11.54	1.93	11.47
LLaMA3.1-8B	10.98	2.77	4.42	18/164	4.13	13.57	2.05	11.45
GuideWeb Agent	**29.99**	31.64	**30.79**	206/687	**44.94**	**52.89**	**21.34**	**28.44**

top-k selection problem: useful guide candidates can exist outside the reference set, but deployed DAP overlays must remain selective. Although several baseline models exhibit much higher recall than the trained GuideWeb Agent, this does not imply superior performance. Instead, these models achieve high recall by producing a significantly larger number of predictions, effectively relying on over-generation to increase the chance of matching gold guide targets.

This behavior is clearly reflected in the *Match/Predicted* statistics. Compared to the GuideWeb Agent, baseline models generate approximately three to four times more predicted guide targets, yet they only identify about 1.5× more correct matches. Such a strategy inflates recall at the cost of precision and indicates that these models fail to selectively identify guide-worthy elements. Rather than understanding which interactive elements truly require guidance, they tend to mark many elements indiscriminately, resulting in redundant and low-utility guides. Given the five-guide cap, some unmatched predictions may still be plausible guide candidates; nevertheless, the large gap between predictions and matches shows that current general-purpose LLMs lack reliable selectivity under a compact guide-authoring setting.

In contrast, for both user intent generation and guide text generation, the trained GuideWeb Agent consistently outperforms all baselines by a large margin across BLEU and ROUGE-L metrics. This demonstrates that task-specific training enables the agent to better align generated intents and guide texts with the underlying page semantics and user needs. Overall, these results highlight that existing general-purpose models struggle to solve the GuideWeb task effectively, and that dedicated modeling and training are crucial for accurate and practical in-app guide generation.

Error patterns. Manual inspection of predictions reveals four common failure modes. First, models often over-annotate self-explanatory controls such as ordinary menu links. Second, they sometimes generate semantically plausible guides for elements outside the capped reference set, which are penalized by exact target matching. Third, text generation may be useful but lexically different from the reference, lowering BLEU/ROUGE. Finally, grounding errors occur when the generated explanation matches the intended function but the XPath points to a nearby parent, icon, or container rather than the precise interactive node. These patterns motivate evaluating both selection and grounding rather than treating the task as ordinary UI captioning.

Shorter ablation. Appendix A reports an ablation of the *Shorter* mechanism. It improves Qwen3-8B guide-target F1 from 10.48 to 14.42 without changing model capacity or training objectives, supporting interaction-focused context reduction for this task.

7 Conclusion and Future Work

We introduce GUIDEWEB, a benchmark for human-in-the-loop in-app guide proposal on real-world web UIs. GuideWeb formalizes guide generation as selecting guide-worthy interactive elements and generating concise, user-aligned guide text, with final annotations produced through LLM-assisted candidate discovery and human verification. Our evaluation shows that general-purpose LLMs tend to over-generate targets, while task-specific training and input structuring enable more selective and informative guide generation. Future work includes multi-step workflows, user feedback, and tighter integration with human-in-the-loop DAP authoring systems.

8 Limitations

This work focuses on main pages and does not address multi-page or stateful workflows that require long-horizon planning, such as checkout flows, settings pages, modal dialogs, and search-result refinement. The released task is primarily DOM-grounded and does not fully model visual salience from screenshots, even though human verification uses rendered pages. Although GuideWeb leverages human verification, the initial candidate pool is LLM-assisted; we mitigate this by allowing human deletion, revision, and addition, but future versions should include larger blind multi-annotator studies and report inter-annotator agreement. The five-guide cap reflects practical overlay design, but it may cause useful predictions beyond the capped reference set to be counted as false positives. Our baselines emphasize strong general-purpose LLMs rather than all possible task-specific or retrieval-based alternatives. Finally, BLEU and ROUGE-L provide reproducible diagnostics for generated text but cannot fully capture semantic equivalence or user-perceived helpfulness. Future work should add human preference evaluation, semantic metrics, category-specific analyses, broader page coverage, and online adaptation to changing websites.

References

Anthropic. 2025. Claude sonnet 4.5 system card. Technical report, Anthropic.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Sri-

vastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905.

Toby Jia-Jun Li, Amos Azaria, and Brad A Myers. 2017. Sugilite: creating multimodal smartphone automation by demonstration. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 6038–6049.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations (ICLR)*.

Liangbo Ning, Ziran Liang, Zhuohang Jiang, Haohao Qu, Yujuan Ding, Wenqi Fan, Xiao-yong Wei, Shanru

Lin, Hui Liu, Philip S Yu, et al. 2025. A survey of web-agents: Towards next-generation ai agents for web automation with large foundation models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6140–6150.

OpenAI. 2025. Gpt-5 system card. Technical report, OpenAI.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Mingyuan Zhong, Gang Li, Peggy Chi, and Yang Li. 2021. Helpviz: Automatic generation of contextual visual mobile tutorials from text-based instructions. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 1144–1153.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

A Ablation of the Shorter Mechanism

Table 4: Ablation of the Shorter mechanism.

Qwen3-8B				
Setting	**P**	**R**	**F1**	**Match/Pred.**
w/o Shorter	9.11	12.33	10.48	72/790
Shorter	**9.52**	**29.67**	**14.42**	181/1901

Table 4 reports an ablation study on the proposed *Shorter* mechanism using Qwen3-8B as the backbone model. Without Shorter, the model processes raw webpage inputs with substantially longer HTML context, resulting in limited recall and F1. After enabling Shorter, which preserves interactive elements and nearby informative text, recall improves from 12.33 to 29.67 and F1 improves from 10.48 to 14.42. This gain is achieved without increasing model capacity or modifying training objectives, indicating that long-form raw HTML can hinder accurate guide prediction when much of the task-relevant signal is concentrated around interactive elements.

B Page Category Analysis

This appendix provides additional quantitative analyses of page categories in the GUIDEWEB benchmark. Table 5 and Figure 6 present the distribution of page categories, highlighting the dominance of landing pages alongside a diverse set of real-world entry points. Figure 7 further breaks down each category by whether pages require in-app guides, showing that only a small fraction of pages are labeled as not guide-worthy, and that such cases are concentrated in a few content-oriented categories. These statistics suggest that page categories differ in guide density, but the present work uses a single shared model rather than category-specific agents.

C Output Schema

D Annotation Guidelines and Human Verification

Table 7 summarizes the operational criteria used to decide whether an element should receive a guide. Human reviewers were instructed to treat LLM outputs as candidates rather than gold labels. They checked the rendered page, the visible text, and the XPath grounding, and then validated, deleted, revised, or added annotations according to these criteria. For each page, reviewers followed the same target-user assumption as the model prompt: a first-time user who understands the website domain but not the local interface. Verification proceeded in three passes: first removing candidates that were decorative, duplicated, or unsupported; then correcting useful candidates whose intent, guide text, action type, or XPath was wrong; and finally scanning the rendered page and element index for missing high-utility targets. The final pass is used to reduce LLM-candidate bias. When more than five elements were useful, reviewers retained the five highest-priority workflow entry points, following the practical overlay budget used in DAP authoring. We do not claim this protocol provides blind inter-annotator agreement; instead, it produces a human-

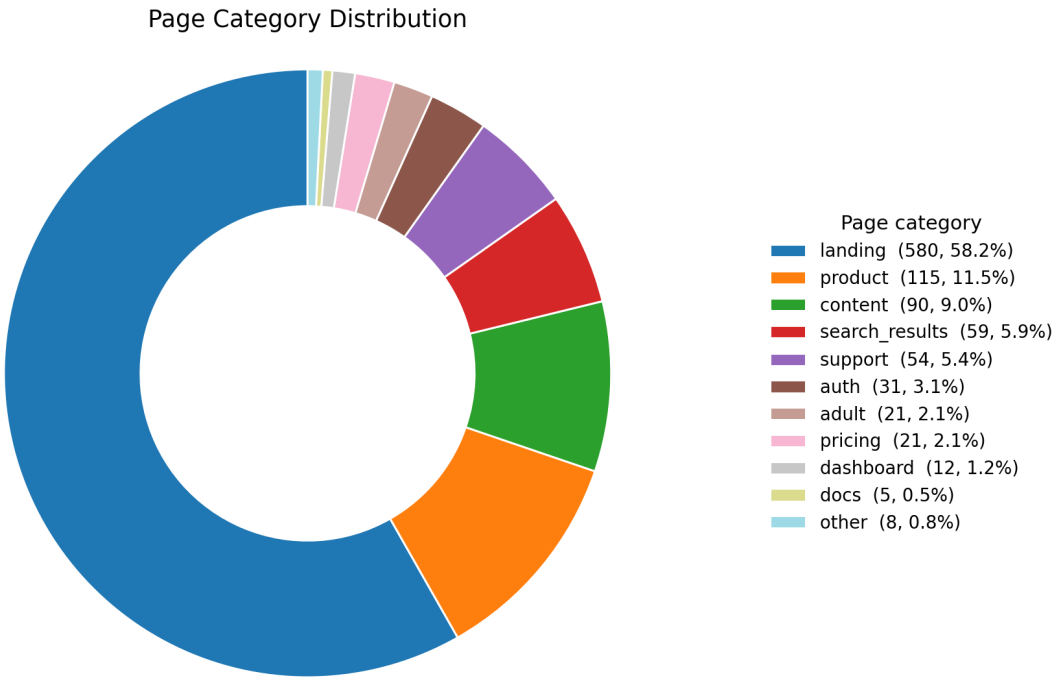

Figure 6: Page category distribution in GUIDEWEB.

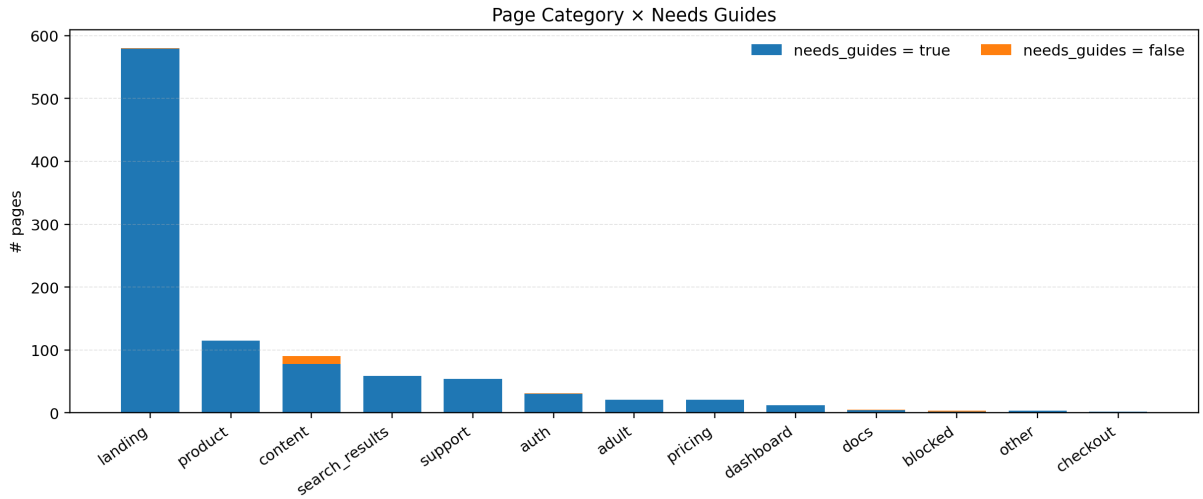

Figure 7: Page category by whether in-app guides are needed in GUIDEWEB.

verified top-k benchmark, and larger independent annotation studies are left for future versions.

E Prompt Templates

For reproducibility, all LLM-based annotation and baseline inference use a structured prompt with the same output schema. The full prompt is released with the dataset; below we show its essential instruction template.

> You are given a webpage represented by indexed interactive elements, visible text, headings, and DOM grounding fields. The target user is a first-time visitor who understands the website domain but is unfamiliar with this interface. Decide whether the page needs in-app guides. Select at most five high-utility guide targets. Prefer elements that support common user goals or important workflows, especially when the function is not obvious from visible text, icon, layout, or surrounding context. Avoid self-explanatory, decorative, legal/footer, social-media, duplicated, or unsupported elements. For each selected target, output a JSON object with `intent`, `action_type`, `guide_text`, `tag`, `visible_text`, and `xpath`. Use only elements present in the input; do not invent new UI elements. The output must be valid JSON and end with `</JSON>`.

Page category	# Pages	# Needs guides	Needs ratio	Avg. guides
landing	580	579	99.8%	3.20
product	115	115	100.0%	3.15
content	90	78	86.7%	2.64
search_results	59	59	100.0%	3.12
support	54	54	100.0%	2.93
auth	31	30	96.8%	2.52
adult	21	21	100.0%	3.10
pricing	21	21	100.0%	3.10
dashboard	12	12	100.0%	3.25
docs	5	4	80.0%	2.60
blocked	3	2	66.7%	1.67
other	3	3	100.0%	3.00
checkout	2	2	100.0%	3.00

Table 5: Page-category statistics in GUIDEWEB. Average guides are computed over task-level guide annotations per page.

Field	Description
`site`	Unique identifier of the website.
`html_file`	Raw HTML of the main page (`page.html`).
`needs_guides`	Boolean indicating whether the page requires guidance.
`page_category`	Coarse category of the webpage (e.g., landing, listing).
`annotations`	List of guide annotations for selected guide targets.
`intent`	Natural language description of user intent.
`action_type`	High-level action type (e.g., search, navigation).
`guide_text`	Generated textual guidance for the target element.
`tag`	HTML tag of the target element.
`visible_text`	Visible text associated with the element (if any).
`xpath`	XPath locating the target element in the DOM.

Table 6: Simplified output schema of GUIDEWEB annotations.

Decision	Include as guide target	Exclude from guide target
User utility	Supports common goals such as search, signup, login, pricing, checkout, support, download, filtering, or configuration.	Decorative, legal, social-media, footer, or duplicated navigation links with little operational value.
Clarity	Important function is not obvious from visible text, icon, layout, or surrounding context.	Purpose is immediately self-explanatory and adding a guide would repeat the label.
Workflow role	Acts as an entry point to a meaningful workflow or reduces uncertainty for first-time users.	Element only triggers trivial page movement or duplicates another selected guide target.
Grounding	XPath points to the concrete interactive node or the smallest stable clickable container.	XPath refers to an overly broad section, unrelated parent container, or non-interactive text node.
Guide text	Concise, action-oriented, and grounded in visible page content.	Vague, promotional, hallucinated, or unsupported by the page.

Table 7: Operational annotation guidelines for selecting guide-worthy elements and validating guide annotations.

F Metric Definitions

Guide target selection. For a given webpage, let G be the set of gold guide targets and $\hat{G}$ the predicted targets, where each target corresponds to a unique DOM element. We define precision, recall, and F1 as follows:

$$\text{Precision} = \frac{|G \cap \hat{G}|}{|\hat{G}|}. \tag{9}$$

$$\text{Recall} = \frac{|G \cap \hat{G}|}{|G|}. \tag{10}$$

$$\text{F1} = \frac{2\,\text{Precision}\,\text{Recall}}{\text{Precision} + \text{Recall}}. \tag{11}$$

Field-level exact-match F1. For each structured field (e.g., `action_type`, `tag`, `visible_text`, `xpath`), we compute exact-match precision, recall, and F1 over field values aggregated across anno-

tations. Let S denote the (multi)set of gold field values and $\hat{S}$ the (multi)set of predicted values.[3] We define:

$$\text{Precision} = \frac{|S \cap \hat{S}|}{|\hat{S}|}. \quad (12)$$

$$\text{Recall} = \frac{|S \cap \hat{S}|}{|S|}. \quad (13)$$

$$\text{F1} = \frac{2\,\text{Precision}\,\text{Recall}}{\text{Precision} + \text{Recall}}. \quad (14)$$

G Implementation Details

Baseline Inference Settings. All baseline models are evaluated in an inference-only setting without any task-specific fine-tuning. We use the original model parameters and default decoding configurations provided by each model implementation. To ensure fair comparison across models and to cover the majority of real-world webpages, we set the maximum input length to 130K tokens for all baselines. This context window is sufficient to accommodate full HTML source code for most main pages in the GUIDEWEB benchmark after preprocessing, avoiding aggressive truncation that may remove critical interactive elements. All baselines are prompted using a unified instruction template and generate guide target predictions and guide text in a single forward pass.

GuideWeb Agent Training. The GUIDEWEB AGENT is trained using the GUIDEWEB training split with full-parameter fine-tuning. Training is conducted on an NVIDIA GB10 GPU with 120 GB of memory, which allows stable optimization with long-context inputs and structured output constraints. We adopt a supervised fine-tuning (SFT) paradigm, where the model is trained to jointly predict guide targets and corresponding guide text conditioned on preprocessed webpage representations.

To reduce unnecessary computation caused by redundant or non-informative HTML content, we employ a *Shorter* mechanism that deterministically constructs the model input. Shorter first removes scripts, styles, comments, tracking code, hidden nodes, and long repeated boilerplate. It then keeps a compact global page excerpt, title/headings, and indexed interactive candidates from links, buttons, inputs, selects, textareas, forms, ARIA button-like roles, and click handlers. For each candidate, it preserves the tag, visible text, selected attributes, nearby text context, and a stable XPath when available. This mechanism shortens the effective input length while preserving task-relevant information, resulting in faster training and inference without degrading accuracy, as confirmed by our experimental results.

Training Hyperparameters. We fine-tune the GUIDEWEB AGENT with full-parameter supervised learning using AdamW. We train for **3 epochs** with a learning rate of 1×10^{-5}, warmup ratio **0.03**, and weight decay **0.0**. The per-device batch size is **1** with gradient accumulation of **4** (effective batch size **4**). We log every **10** steps and save checkpoints every **2000** steps. For numerical stability and memory efficiency, we load the base model in **FP16** and train with **BF16** mixed precision, enabling **gradient checkpointing**. We set the maximum prompt length to **3950** tokens and the maximum generation length to **2000** tokens, resulting in a maximum sequence length of **6014** tokens (including a small buffer). All runs use a fixed random seed (**13**).

Input Construction and Shorter Configuration. We render each webpage using a fixed viewport of **1280×720**. For visible text, we keep up to **800** text blocks, truncating each block to at most **400** characters. We additionally include a short global excerpt of the full-page text using an excerpt ratio of **0.02**, bounded to **[100, 200]** characters. For structural cues, we keep headings (enabled) with at most **20** headings and at most **40** characters per heading. For interactive candidates, we extract up to **2000** interactive elements, truncate each element's visible text to **120** characters, and enforce an overall interactive-text budget of **6500** characters. To represent targets robustly, we provide XPaths using the `stable_then_abs` mode, limiting XPath textual fields to **80** characters and attribute fields to **200** characters. Model outputs are required to end with a fixed end marker (`</JSON>`).

Filtering Long Samples. To avoid near-limit truncation and unstable supervision, we filter overly long samples during training. Specifically, we drop samples whose estimated total tokens exceed **5200**, or whose prompt/output lengths are within **98%** of the configured maximums. Dropped instances are recorded in `dropped_samples.jsonl` for traceability.

[3] Our implementation handles duplicates consistently between S and $\hat{S}$.