WebVLN: Vision-and-Language Navigation on Websites

Qi Chen*, Dileepa Pitawela*, Chongyang Zhao*, Gengze Zhou, Hsiang-Ting Chen, Qi Wu†

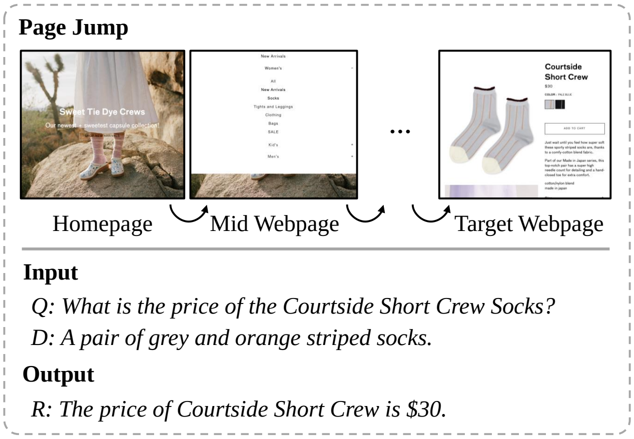
Australian Institute for Machine Learning, The University of Adelaide
{qi.chen04, dileepa.pitawela, chongyang.zhao, gengze.zhou, tim.chen, qi.wu01}@adelaide.edu.au

Abstract

Vision-and-Language Navigation (VLN) task aims to enable AI agents to accurately understand and follow natural language instructions to navigate through real-world environments, ultimately reaching specific target locations. We recognise a promising opportunity to extend VLN to a comparable navigation task that holds substantial significance in our daily lives, albeit within the virtual realm: navigating websites on the Internet. This paper proposes a new task named Vision-and-Language Navigation on Websites (WebVLN), where we use question-based instructions to train an agent, emulating how users naturally browse websites. Unlike the existing VLN task that only pays attention to vision and instruction (language), the WebVLN agent further considers underlying web-specific content like HTML, which could not be seen on the rendered web pages yet contain rich visual and textual information. Toward this goal, we contribute a dataset, WebVLN-v1, and introduce a novel approach called Website-aware VLN Network (WebVLN-Net), which is built upon the foundation of state-of-the-art VLN techniques. Experimental results show that WebVLN-Net outperforms current VLN and web-related navigation methods. We believe that the introduction of the new WebVLN task and its dataset will establish a new dimension within the VLN domain and contribute to the broader vision-and-language research community. Code is available at: https://github.com/WebVLN/WebVLN.

Introduction

Vision-and-Language Navigation (VLN) (Anderson et al. 2018) aims to seamlessly integrate visual perception and action with language understanding, to enable AI agents to navigate and interact effectively within real-world environments. Interestingly, the resemblance can be found in the virtual online environment, where users might rely on AI agents to assist them in gathering information about certain products even when they can only offer broad and vague instructions such as *"how much does a pair of grey and orange striped socks cost"*. This extends beyond the boundaries of traditional VLN tasks, including not only vision and instruction (language) but also incorporating the abundant information embedded within webpage like HTML. With this consideration, we introduce an extended VLN task, denoted as Vision-and-Language Navigation on Websites (WebVLN).



Input

Q: What is the price of the Courtside Short Crew Socks?

D: A pair of grey and orange striped socks.

Output

R: The price of Courtside Short Crew is $30.

Figure 1: An example of the WebVLN task. An agent is initiated on the homepage of a website and asked a question Q with an auxiliary description D. To respond, the agent is required to intelligently navigate and explore the website, gather information through observation, and finally provide an accurate response/answer R in a free-form sentence.

Figure 1 shows an example of the WebVLN task. In this scenario, an agent starts its journey from a website's front page, presented with a question Q accompanied by an auxiliary description D. The agent emulates genuine user behaviour and navigates through the website. It processes the current view of the webpage and engages in common web browsing activities such as reading the images and text, and clicking on the links to navigate to the next pages. The agent's objective is to efficiently traverse the website and reach a target webpage, which contains the necessary information to answer the question Q and produce an accurate response R to the question.

The new task poses several new challenges. **First**, the choices available to an AI agent navigating a website are substantially more than those in traditional discrete VLN scenarios, which are confined to adjacent navigable viewpoints in physical environments. While in WebVLN, the

[*] These authors contributed equally.

[†] Corresponding author.

range of choices for each observed webpage is significantly broader as it contains a vast array of content, features, links, and interactive elements. Each webpage offers multiple avenues for navigation, such as clicking on various links, buttons, and dropdown menus. **Second**, due to the intrinsic diversity of available choices on each webpage, WebVLN constructs a more intricate and complex navigation graph than traditional VLN, making it nearly impossible to explore all the content on websites by a naive heuristic trial-and-error scheme. Thus, in the WebVLN task, an ideal method should seek to maximise accurate choices while minimising the need for exploration by leveraging varied information available within the webpage.

Due to the lack of an off-the-shelf dataset for WebVLN, we have collected a new WebVLN-v1 dataset to facilitate research in this field. It comprises $8,990$ records/paths with $14,825$ QA pairs derived from three different shopping websites (aliased as SA, HB and ES). Differing from other VLN datasets (Anderson et al. 2018; Qi et al. 2020b) that only consider the visual inputs of the environment, our WebVLN-v1 incorporates both visual and textual contents extracted from the websites. Furthermore, in comparison to other web-related datasets, such as web navigation (Liu et al. 2018; Xu et al. 2021; Mazumder and Riva 2020; Yao et al. 2022) and web QA (Chang et al. 2022; Hsiao et al. 2022), our WebVLN-v1 seamlessly integrates both navigation and QA environments with question-based instructions, resulting in a unified benchmark.

To tackle the challenging WebVLN task, we propose a new method called Website-aware Vision-and-Language Navigation Network (WebVLN-Net) based on the widely used VLN framework–VLN↻BERT (Hong et al. 2021). Besides the visual input (screenshot) and instruction (question & description), WebVLN-Net considers the underlying HTML of each webpage and extracts elements such as clickable buttons. Upon reaching a “stop” token, our model initiates answering the question using information from both click history and the current “stop” webpage. The evaluation of the model performance is based on the metrics from both VLN and VQA domains. Specifically, for VLN, we consider success rate (SR), oracle success rate (OSR), success rate weighted by path length (SPL), and Trajectory Length (TL), while adopting Wu-Palmer Similarity (WUPS) (Wu and Palmer 1994) for VQA evaluation due to the open-end setting, *i.e.*, generating a free-form sentence as an answer.

In summary, our contributions include:

- A new task - Vision-and-Language Navigation on Websites (WebVLN), where the agent emulates human web browsing behaviours and navigates to a specified target webpage based on the input question and its auxiliary description, subsequently answering the question using information extracted from the target webpage.
- A new WebVLN-v1 dataset, consisting of $8,990$ records/paths, and $14,825$ question-answer (QA) pairs derived from three different websites, covering both navigation and QA on the web environments.
- A new method, named Website-aware VLN Network (WebVLN-Net), which not only considers the visual input (screenshot) and linguistic instruction but also uses web-specific content (*i.e.*, HTML of the webpage) to enhance decision-making precision.

Related Work

As the WebVLN is a new task, we briefly overview several closely relevant works *w.r.t.* Vision-and-Language Navigation (VLN) and other web-related navigation and QA tasks.

Vision-and-Language Navigation (VLN) The VLN task (Anderson et al. 2018) extends the vision and language research with sequential action prediction and is one of the most influential tasks in Embodied AI. The research on VLN is dedicated to addressing the alignment of linguistic instructions with visual cues and actions, some work fine-graining the navigation instructions to achieve sub-goal planning (Hong et al. 2020; He et al. 2021a; Zhu et al. 2020), and some concentrate on utilizing object information to identify landmarks from observations (Gao et al. 2021; Qi et al. 2020a, 2021). Temporal information is specifically designed in (Hao et al. 2020; Hong et al. 2021; Chen et al. 2021, 2022; Qiao et al. 2022, 2023; Zhao, Qi, and Wu 2023) to capture long-range dependencies across past observations and actions, which are crucial during navigation. some methods incorporate external knowledge during navigation (Li et al. 2022; Gao et al. 2021). Recently, several methods leverage commonsense knowledge from LLMs and build an LLMs-as-agent pipeline to perform zero-shot VLN (Zhou, Hong, and Wu 2023). However, the VLN tasks require spatial awareness of agents and mainly focus on the photo-realistic environment, which would not consider the web-specific information (*e.g.*, descriptions denoted by “alt” in HTML) when directly applied to website navigation.

Web Navigation and Question-Answering Web navigation task (Toyama et al. 2021; Yao et al. 2022; Burns et al. 2022) involves developing algorithms or models that enable automated agents to navigate and interact with websites on the Internet. There are some related datasets (Liu et al. 2018; Xu et al. 2021; Mazumder and Riva 2020; Yao et al. 2022; Deng et al. 2023; Zhou et al. 2023). For example, MiniWoB++ (Liu et al. 2018), RUSS (Xu et al. 2021) and FLIN (Mazumder and Riva 2020) cover sites with diverse user instructions from simple tasks to complex ones like booking flights. Many previous works use various methods on these datasets, which, however, depend on Document Object Model (DOM) structure (Jia, Kiros, and Ba 2019; He et al. 2021b) and hence hamper their flexibility. As for web QA, it mimics the human behaviour of posing a question, aggregating information on the webpage, and generating a response. Several benchmarks, such as WebQA (Chang et al. 2022) and ScreenQA (Hsiao et al. 2022), have been proposed. However, they only offer a single webpage for each question. In contrast, we break the boundary between web navigation and web QA by merging them into a unified task called WebVLN, aligning more closely with human behaviour. Moreover, we design a framework (*i.e.*, WebVLN-Net) that can be easily adapted for different websites.

WebVLN Task and Simulator

Problem Definition

As in Figure 1, the WebVLN task requires an agent to follow natural language instructions (*i.e.*, question & description) to navigate from a homepage to a target webpage and answer the question based on the information from both trajectories and the target webpage. Formally, at the beginning of each episode, the agent is given an input question Q and an auxiliary description D (*e.g.*, details of the target item) as the questions are often brief and may lack enough information for locating a unique item [1]. Then, the agent observes the current webpage $W^{(i)} = \langle I^{(i)}, \mathcal{B}^{(i)} \rangle$, where $I^{(i)}$ and $\mathcal{B}^{(i)}$ are the screenshot and the set of clickable buttons in the (i-th) current page, respectively. Here, each clickable button $b \in \mathcal{B}^{(i)}$ is represented by its description d (*i.e.*, "alt" in HTML) and image e, namely $b = \langle d, e \rangle$. Note that if only having either description or image, use $\varnothing$ for the other one. In this setting, the agent must execute a sequence of actions $\mathcal{A}$, where each action $a_t \in \mathcal{A}$ leads to a new state $s_t \in \mathcal{S}$. Each state s_t contains the information derived from both the state of current page $W^{(i)}$ that the agent locates on and the state of current action a_t that the agent performs. Besides, we define a special *End Of Action* (*i.e.*, `[EOA]`) as the "stop" token, which would be predicted if the current state refers to the target webpage. Ideally, given question Q and auxiliary description D, the agent should predict a response/answer R based on the contents in the target state $s_{\texttt{[EOA]}}$.

WebVLN Simulator

In this part, we establish a WebVLN simulator on three different shopping websites (aliased as SA, HB, and ES), covering $1,485$ products such as socks, fossils and blankets, and mirror the actions of humans. The details are as follows.

Observations To build the simulator, we enable an agent to navigate within a website by interacting with various buttons. During the i-th webpage $W^{(i)}$, the simulator generates an RGB image observation $I^{(i)}$ (*i.e.*, screenshot) that corresponds to the agent's viewpoint. It is worth noting that, to prevent any instances of mismatch, we provide the entire screenshot of the current webpage, rather than just a partial view within the display window.

Action Space The primary difficulty in simulator implementation lies in defining the action space that depends on the current state. Naturally, we hope that the agent will not jump to the currently unreachable page indiscriminately. Hence, during each step t, the simulator additionally generates a subset of next-step reachable clickable buttons $\mathcal{B}_{t+1} \subseteq B$, where B contains all clickable buttons present on the website. The agent interacts with the simulator by selecting a new button $b_{t+1} \in \mathcal{B}_{t+1}$. To establish $\mathcal{B}_{t+1}$, the simulator constructs a directed graph for each website, denoted as $G = \langle B, E \rangle$, where the existence of an edge indicates a navigable transition between two webpages, which is accessible by the agent. In each step, we also incorporate a specific button $b_{\texttt{[EOA]}}$ for the "stop" action.

WebVLN-v1 Dataset Construction

Automatic Path Generation

Following the way for generating path in R2R (Anderson et al. 2018), we sample a target webpage and subsequently construct the shortest path from the homepage to the selected target according to the aforementioned graph G. For each path, we manually check it to determine its reasonability from a human perspective and then discard the unreasonable ones (*e.g.*, clicking the advertisement icon multiple times in succession). Furthermore, we remove paths with fewer than 2 webpage transitions to maintain dataset quality and diversity, resulting in a total sample of 8,990 paths.

LLM-aided Question-Answer Generation

To alleviate the workload on humans, we seek to generate QA pairs with the assistance of Large Language Models (LLMs)[2] based on the multimodal content from the webpage. We depict the details in the following.

Image Data Handling & HTML Cleaning Firstly, we employ BLIP-2 (Li et al. 2023), a large and powerful model for image captioning, to transform the website images into captions, ensuring that the LLMs are able to capture the visual information. After that, another critical step is processing the word list, since the original texts directly from webpages are often disorganised and challenging to work with, which also include irrelevant information, such as messy code. Hence, we systematically adopt a rule-based filtering approach to manually eliminate such information, which enhances the logic and readability of these texts.

Designing Rules for Generating QA Pairs We start to design a series of rules for the LLM to guide its behaviour when generating required QA pairs. The initial rules include:

- *Provide 3 questions and their answers that can be directly found from the information provided in the text.*
- *Ask the first question about price and second about available sizes and the third about material.*
- *If precise answers cannot be found for those questions, then ask the questions on colours and availability in stock.*
- *Phrase your questions in a clear and concise manner to ensure they can be accurately answered by the given content.*
- *Answer should be to the point without additional information.*

To mitigate the impact of negative information, we introduce two additional rules as follows.

- *The provided text is all from an online shopping website, there is some disturbing information which is irrelevant to the products, such as "sign in". Make sure your questions and answers will focus on the products themselves.*

[1] Note that if the question has sufficient information to locate the target webpage, the auxiliary description will remain empty.

[2] Here, we use ChatGPT with the gpt-3.5-turbo model.

Table 1: Comparison between our WebVLN-v1 dataset and the related datasets, which cover three different types of environment, *i.e.*, embodied scene, Mobile App, and website. To ensure a more comprehensive comparison, we evaluate the datasets across three dimensions: the environment (Env.), the instruction (Ins.) and the target task. Here, "Temp." refers to whether the environment would be temporally changed. "Image", "Text" and "HTML/Code" are the components that environment covers. "Que." and "Des." are the abbreviations of question and description, respectively, referring to the type of instruction that the dataset contains. "Ins. Level" means whether the instruction is a high-level statement or a low-level step-by-step command.

Env. Type	Dataset	Environment (Env.)				Instruction (Ins.)			Task	Number
		Temp.	Image	Text	HTML/Code	Que.	Des.	Ins. Level		
Embodied	R2R (Anderson et al. 2018)	✓	✓				✓	Low	Navigation	$21,567$
	EQA (Das et al. 2018)	✓	✓			✓	✓	High	Navigation + QA	$5,281$
	REVERIE (Qi et al. 2020b)	✓	✓				✓	High	Localise Remote Object	$21,702$
Mobile App	PixelHelp (Li et al. 2020)	✓		✓	✓		✓	Low	Navigation	187
	MoTIF (Burns et al. 2022)	✓	✓	✓	✓		✓	High	Navigation	$1,125$
	META-GUI (Sun et al. 2022)	✓	✓	✓	✓	✓		High	Dialogue	$4,707$
Website	MiniWoB++ (Liu et al. 2018)	✓		✓	✓		✓	Low	Navigation	-
	RUSS (Xu et al. 2021)	✓		✓	✓		✓	Low	Navigation	741
	FLIN (Mazumder and Riva 2020)	✓		✓	✓		✓	High	Navigation	$53,520$
	WebShop (Yao et al. 2022)	✓		✓	✓		✓	High	Navigation	$12,087$
	MIND2WEB (Deng et al. 2023)	✓	✓	✓	✓		✓	High	Navigation	$2,350$
	WebQA (Chang et al. 2022)		✓	✓		✓		High	Question-Answer (QA)	$\sim 46,500$
	ScreenQA (Hsiao et al. 2022)		✓	✓		✓		High	Question-Answer (QA)	-
	WebVLN-v1 (ours)	✓	✓	✓	✓	✓	✓	High	Navigation + QA	$14,825$

- *The provided texts may contain punctuation and symbols, which are irrelevant to the products, you should be able to distinguish them and make sure they won't appear in the generated questions and answers.*

Prompt for Generating Final QA Pairs We obtain the final prompt $\mathcal{P}$ by directly concatenating the three aforementioned terms, *i.e.*, $\mathcal{P} = \{$*"There is a picture of the product with the caption of"* + `caption` + *"After that, here are all the words that appear on the website:"* + `word list` + *"Lastly, I will give the following instructions, and you will be strictly following the instructions:"* + `rules`$\}$. Moreover, due to the varying lengths of these three terms – with the word list being the longest and the caption the shortest – it becomes necessary to appropriately adjust the weighting of each component. It is particularly crucial given that the caption, despite its brevity, holds rich information.

Quality Checking Due to the inherent uncertainty introduced by LLMs, it is vital to perform quality checks on every generated question-answer (QA) pair. Concretely, we randomly select 100 QA samples for each website to assess their quality. Our stringent quality checks adhere to the following criteria *w.r.t.* question and answer, respectively:

- The generated questions by LLM should relate to the actual products visible on the website.
- The generated answers should be correct, brief, and correspond to the questions. Any answer surpassing the scope of the question will be considered invalid.

We have undertaken multiple iterations of the constructed prompt until all the samples are reasonable and correct. In each iteration, a subset of samples is generated for manual quality assessment. Each sample has undergone evaluation by at least two assessors for a reliable evaluation.

WebVLN-v1 Dataset Analysis

WebVLN-v1 Dataset vs. Related Datasets

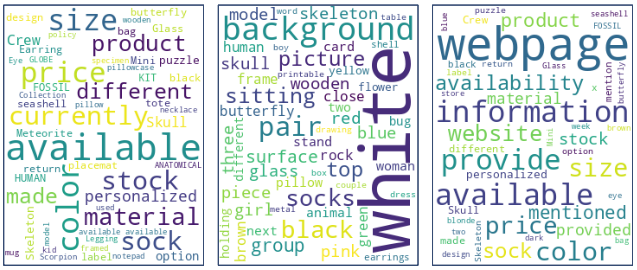
We compare our WebVLN-v1 dataset with the most relevant datasets *w.r.t.* Embodied AI, Mobile App, and Website. Specifically, for Embodied AI datasets, we consider R2R (Anderson et al. 2018), REVERIE (Qi et al. 2020b) and EQA (Das et al. 2018), where the first two are the widely used vision-and-language navigation (VLN) datasets while the last one is a famous embodied question answering dataset. Regarding App-based datasets, we compare PixelHelp (Li et al. 2020), MoTIF (Burns et al. 2022) and META-GUI (Sun et al. 2022). As for the website, we consider seven datasets for a comprehensive comparison, including MiniWoB++ (Liu et al. 2018), RUSS (Xu et al. 2021), FLIN (Mazumder and Riva 2020), WebShop (Yao et al. 2022), MIND2WEB (Deng et al. 2023), WebQA (Chang et al. 2022), and ScreenQA (Hsiao et al. 2022).

Table 1 demonstrates that our WebVLN-v1 dataset is unique, covering rich information (*i.e.*, temporal sequence, image, text and HTML) from the environment and two different types of instructions (*i.e.*, questions and descriptions/statements), while others only focus on them partially. Moreover, our WebVLN-v1 is able to support both navigation and question-answering tasks, rather than other website-related datasets that can only support one of them.

WebVLN-v1 Statistics

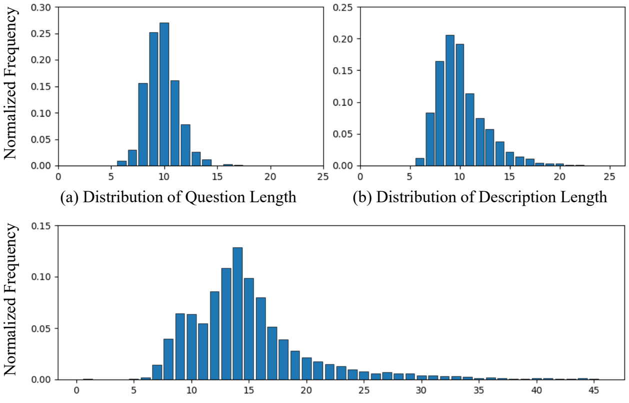
In this part, we provide statistics of the WebVLN-v1 dataset, including the word cloud, the distribution of textual lengths (question, description and answer), and the data split.

Word Cloud As shown in Figure 2, we visualise the questions, auxiliary descriptions, and answers of our WebVLN-v1 dataset as Venn-style word cloud (Coppersmith and Kelly 2014). Here, the size of each word corresponds to the harmonic mean of its count.



(a) Questions (b) Descriptions (c) Answers

Figure 2: Word cloud of (a) questions, (b) descriptions, and (c) answers on the proposed WebVLN-v1 dataset. The larger the font size, the greater percentage in the corpus.



(a) Distribution of Question Length (b) Distribution of Description Length

(c) Distribution of Answer Length

Figure 3: Distributions of (a) question length, (b) description length, and (c) answer length on the WebVLN-v1 dataset.

Lengths of Question, Description, Answer and Path Figure 3 exhibits the distributions of question length, description length and answer length on the WebVLN-v1 dataset. Specifically, the majority of questions consist of $8 \sim 12$ words (Figure 3(a)) while most descriptions have $7 \sim 14$ words (Figure 3(b)). The answers, in Figure 3(c), exhibit a broader range, with some extending to $30 \sim 40$ words, as they derive content from the question, description and webpage. The average length of the paths is 3.32.

Data Splits We split 60% samples as training data, 10% samples as validation data and 30% samples as testing data (*i.e.*, $8,960/1,262/4,603$). Notably, to prevent any information leakage, we carefully check that the training, validation, and testing sets cover all three websites, but their records/paths remain distinct and non-overlapping.

WebVLN-Net

For the WebVLN task, we propose a new model, called Website-aware Vision-and-Language Navigation Network (WebVLN-Net), which is based on a widely used vision-and-language navigation framework VLN↻BERT (Hong et al. 2021). As shown in Figure 4, our model contains three main components: initialisation, navigation and answering. Specifically, we first initialise the state and context tokens by a pre-trained BERT model. Subsequently, we input these initialised language tokens, along with the screenshot and button tokens extracted from the current webpage, into the navigation component. This process iterates until the target webpage is reached. Last, the answering head in the answering component generates the final answer.

State and Context Initialisation During initialisation ($t = 0$), our model receives a word sequence comprising the classification token `[CLS]`, the separation token `[SEP]`, and the language tokens $\mathcal{V}$ extracted from both the question Q and the auxiliary description D. Here, `[CLS]` and `[SEP]` are predefined tokens in BERT models. Similar to VLN↻BERT (Hong et al. 2021), the `[CLS]` token is used to aggregate relevant vision-language cues from the input sequence. In this context, we define the embedded `[CLS]` token as the initial state representation s_0. We update it during the whole training phase, ensuring it could be aware of the entire navigation and question-answering tasks. Formally, the process can be defined as

$$s_0, \mathcal{V} = \texttt{Init}(\texttt{[CLS]}, Q, \texttt{[SEP]}, D), \quad (1)$$

where the $\texttt{Init}(\cdot)$ indicates the initialisation process.

Web Navigation We adapt the model proposed in (Hong et al. 2021) to incorporate the learning of navigation and the concurrent selection of clickable buttons. As shown in Figure 4, at each time step, the network takes four different token sets as input: the preceding state token s_{t-1}, the language tokens $\mathcal{V}$, the screenshot tokens $\mathcal{I}_t$, and the button tokens $\mathcal{B}_t$. Specifically, we "patchify" the screenshot as a set of image patches and convert them to the tokens $\mathcal{I}_t$ by using a Transformer-based image encoder. Likewise, as each button consists of an image and a description (*i.e.*, "alt" in HTML), we introduce a button encoder that contains an image encoder and a text encoder to derive their corresponding tokens. After that, we concatenate tokens associated with the same button, followed by projecting the concatenated token back to its original dimension using a linear projection layer. Subsequently, we put all the tokens into a multi-layer Transformer to obtain an action probability p_t:

$$s_t, p_t = \texttt{Nav}(s_{t-1}, \mathcal{V}, \mathcal{I}_t, \mathcal{B}_t). \quad (2)$$

Here, the $\texttt{Nav}(\cdot)$ refers to the navigation process in each step. Notably, the set of button tokens $\mathcal{B}_t$ involves an *End Of Action* (`[EOA]`) token, selected when the agent reaches the target webpage. The state would be updated to $s_{\texttt{[EOA]}}$.

In navigation steps ($t > 0$), the state token s_t, screenshot tokens $\mathcal{I}_t$, and button tokens $\mathcal{B}_t$ employ self-attention across the entire input sequence, while the language tokens $\mathcal{V}$ only serve as the keys and values in the Transformer. We regard the language tokens generated during the model's initialisation step as a good representation of both the question and auxiliary description ($Q\&D$), obviating the necessity for subsequent encoding in later stages and saving computational resources.

Question Answering To generate the final answer, we introduce an Answering Head, which is a M-layer Transformer decoder. Considering an open-ended question-answering setting, our objective is to generate the answer

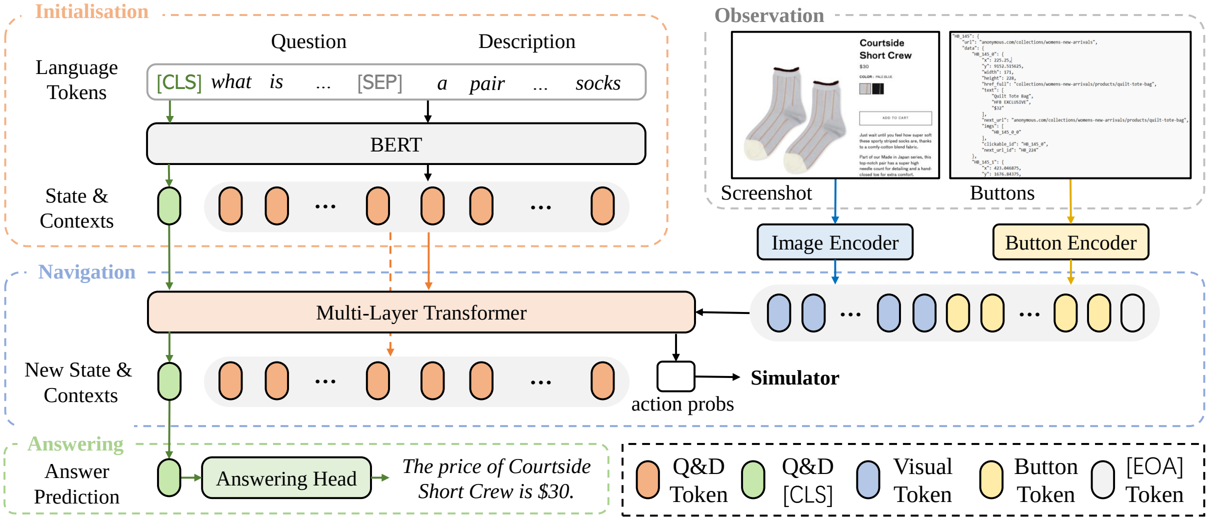


Figure 4: Overall architecture of WebVLN-Net. We take an example that the question Q is *"What is the price of the Courtside Short Crew Socks?"* with an auxiliary description D as *"a pair of grey and orange striped socks"*. The observation at each step contains a screenshot of the current webpage and all the clickable buttons derived from its HTML.

as a free-form sentence autoregressively. Mathematically,

$$R = \texttt{Ans}(s_{\texttt{[EOA]}}), \tag{3}$$

where $\texttt{Ans}(\cdot)$ refers to the answering process. Here, R is the predicted answer consisting of L words (*i.e.*, $R = \{w_l\}_{l=1}^{L}$) and $s_{\texttt{[EOA]}}$ denotes the last state aforementioned above.

Training For navigation, we train our network using an imitation learning (IL) objective. Concretely, our agent navigates on the ground-truth trajectory by following the teacher actions and computes a cross-entropy loss for each decision made. Formally, we minimise the navigation loss function, which can be formulated for each specific sample by

$$\mathcal{L}_{nav} = -\sum_{t} a_t \log(p_t) - \eta \sum_{t} a_t^* \log(p_t), \tag{4}$$

where a_t is the sampled action and a_t^* is the teacher action. Here, η represents a coefficient used to weigh the IL loss.

The training of the Answering Head employs the autoregressive Teacher Forcing (Williams and Zipser 1989) approach, wherein the prediction at each step should maximise the likelihood of the subsequent token:

$$\mathcal{L}_{ans} = \sum_{l=1}^{L} -\log p(w_l | w_{<l}, s_{\texttt{[EOA]}}), \tag{5}$$

where w_l indicates the l-th token in the answer, and $w_{<l}$ denotes the tokens before w_l. L is the total number of tokens. The final loss function can be defined as

$$\mathcal{L} = \mathcal{L}_{nav} + \lambda \mathcal{L}_{ans}, \tag{6}$$

where λ is the weighting hyper-parameter.

Experiments

Evaluation Metrics and Baselines

Metrics for Navigation Following the evaluation metrics in R2R (Anderson et al. 2018), we assess the shortest path distance in the navigation graph G between the finally located webpage of the agent and the target webpage. We consider an episode to be a *success* if the agent stops on the target webpage, *i.e.*, success rate (SR), and its variant – oracle success rate (OSR). We also measure the navigation performance by considering the path length, *i.e.*, the success rate weighted by Path Length (SPL) and Trajectory Length (TL).

Metrics for QA For question-answering, we follow the open-ended setting that seeks to generate a free-form natural language sentence to answer the given question, which is more flexible and practical than regarding QA as a classification problem. Thus, rather than an exact accuracy between predicted and ground-truth answers, we adopt Wu-Palmer Similarity (WUPS) (Wu and Palmer 1994), aiming to quantify the semantic differences between a predicted answer and the ground truth. WUPS assigns a value ranging between 0 and 1, reflecting their degree of similarity. Following (Malinowski and Fritz 2014), we set thresholds of the WUPS as 0.9 and 0.0 separately with a scaling factor of 0.1, whereby scores below the threshold are proportionally adjusted.

Baselines We evaluate the performance of both navigation and QA by comparing the results with baselines. For a comprehensive comparison, we consider two different types of baselines, *i.e.*, traditional VLN and web-related navigation. Specifically, we employ VLN↻BERT (Hong et al. 2021) (randomly initialised and initialised by LXMERT (Tan and Bansal 2019)) as the VLN baseline, which is widely used in the VLN task. Notably, to well evaluate the performance, we seek to adapt VLN↻BERT to our task and dataset with minimal changes. Specifically, following its original design, we take as inputs the linguistic instruction (question & description) and screenshot. The model would predict the intermediate navigation action and finally answer the question by incorporating the same QA head as ours. As for the web-

Table 2: Comparison with baseline methods. We compare our method with VLN↻BERTand WebGUM, which are the widely used VLN model and state-of-the-art (SoTA) multimodal web navigation foundation model, respectively. VLN↻BERT and VLN↻BERT* are randomly initialised and initialised by LXMERT, respectively. WebGUM and WebGUM† denote the models based on T5-small and T5-base separately.

Method	Val						Test					
	SR ↑	OSR ↑	SPL ↑	TL ↓	WUPS0.9 ↑	WUPS0.0 ↑	SR ↑	OSR ↑	SPL ↑	TL ↓	WUPS0.9 ↑	WUPS0.0 ↑
Random	0.05	0.17	0.02	6.49	0.00	0.00	0.04	0.17	0.02	6.50	0.00	0.00
VLN↻BERT	17.59	17.59	16.73	6.81	9.99	13.91	11.28	12.04	10.75	7.55	7.12	9.26
VLN↻BERT*	18.62	18.62	18.14	6.96	11.23	14.98	12.23	12.23	11.74	7.72	8.50	10.36
WebGUM	6.02	6.02	6.02	**2.99**	1.84	4.08	9.71	9.71	9.71	**3.15**	3.57	6.98
WebGUM†	31.22	31.78	31.22	3.44	18.26	24.88	29.29	29.39	29.26	3.44	17.34	23.48
Ours	**39.46**	**39.54**	**39.46**	3.71	**24.26**	**31.87**	**34.76**	**34.80**	**34.59**	4.34	**22.13**	**28.58**

Table 3: Ablation study on the test set of WebVLN-v1 dataset. Q, D and I are the input question, auxiliary description and screenshot, respectively. The notations d and e are the button description/text and the button image separately.

Q	D	I	d	e	SR ↑	OSR ↑	SPL ↑	TL ↓	WUPS0.9 ↑	WUPS0.0 ↑
✓		✓			6.32	6.34	6.13	6.89	4.74	5.52
✓	✓	✓			12.23	12.23	11.74	7.72	8.50	10.36
✓	✓	✓	✓		28.63	28.72	28.62	**3.22**	15.97	22.51
✓	✓	✓	✓	✓	**34.76**	**34.80**	**34.59**	4.34	**22.13**	**28.58**

Table 4: Zero-shot by using LLMs. * denotes using GPT4 while others are ChatGPT (gpt-3.5-turbo).

Methods	SR ↑	OSR ↑	SPL ↑	TL ↓	WUPS0.9 ↑	WUPS0.0 ↑
Humans	93.94	93.94	80.92	3.88	60.55	79.94
AgentBench	6.97	11.94	4.13	5.39	1.64	4.65
NavGPT	7.46	12.94	4.53	4.98	2.43	5.23
NavGPT*	16.92	21.89	11.61	5.43	5.97	12.06

related navigation, we consider a state-of-the-art (SoTA) instruction-finetuned foundation model WebGUM (Furuta et al. 2023), which is built upon the T5 model (Raffel et al. 2020). Moreover, we evaluate the performance of a random strategy and humans, helping to understand the lower and upper bounds of the WebVLN task, respectively.

Comparison with Baselines We evaluate the performance of the proposed WebVLN-Net compared with the baseline methods. In all the experiments, we set the weighting hyperparameters η and λ equal to 1. In Table 2, for navigation, we obtain the best results in SR, OSR and SPL, and comparable results in TL, both when compared to VLN methods and web-related navigation techniques. As for QA metrics, our method consistently outperforms all the baselines. All the results demonstrate the effectiveness of our WebVLN-Net. Note that the QA metrics for the random method are 0 since generating a free-form answer randomly is nearly impossible to overlap with ground truth. WebGUM† achieves better performance than WebGUM mainly due to the larger number of parameters (220 million vs. 60 million).

Ablation Study To test the impact of each component in our WebVLN-Net, we conduct an ablation study by incorporating them alternately. The basic model only considers an input question Q and a screenshot I. From Table 3, the basic model obtains the lowest results on both navigation and QA evaluation metrics. While with an auxiliary description D, our model achieves higher performance (*e.g.*, SR: 6.32 → 12.23; WUPS0.9: 4.74 → 8.50). Moreover, the model's performance can be enhanced by using buttons (containing text d only) from the HTML (SR: 12.23 → 28.63). After further incorporating a multimodal button (containing text d & image e), it attains the best performance across both navigation and QA metrics (except for TL, which is comparable).

Zero-shot using LLMs To further investigate the difficulty of our proposed WebVLN task and the corresponding WebVLN-v1 dataset, we conduct a zero-shot evaluation setting for the popular large language models (LLMs). We randomly select 201 samples from the validation set and test on two LLMs-as-agent pipelines, AgentBench (Liu et al. 2023) and NavGPT (Zhou, Hong, and Wu 2023). For AgentBench, we adopt the same prompt from the WS (Web Shopping) task in the paper and delete the one-shot example, test on the gpt-3.5-turbo model. For NavGPT, we modify the VLN task description into a web shopping task description and test on both gpt-3.5-turbo and gpt-4 models. All the formatted observations are replaced with the WebVLN-v1 sample observation. From Table 4, the zero-shot performance of LLMs-as-agent methods falls short of reaching human-level performance. These results show the necessity for the ongoing advancement of intelligent agents, as the current SoTA LLMs are far from perfect performance in our task. Moreover, our WebVLN can also serve as a metric to gauge such progress.

Conclusion

In this paper, we propose a novel task, Vision-and-Language Navigation on Websites (WebVLN), extended from conventional VLN. It seeks to enable an agent to answer the user's questions via navigating/exploring the websites and integrating useful information. To support research in this new task, we collect a new WebVLN-v1 dataset and design a baseline method called Website-aware Vision-and-Language Navigation Network (WebVLN-Net). The experiments demonstrate the effectiveness of our WebVLN-Net. Moreover, we perform a zero-shot evaluation of LLM-based methods using the WebVLN-v1 dataset, where the performance is far from saturation, highlighting the utility of our WebVLN-v1 as a benchmark to assess progress in this field.

References

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and Van Den Hengel, A. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 3674–3683.

Burns, A.; Arsan, D.; Agrawal, S.; Kumar, R.; Saenko, K.; and Plummer, B. A. 2022. A dataset for interactive vision-language navigation with unknown command feasibility. In *Proc. Eur. Conf. Comput. Vis.*, 312–328.

Chang, Y.; Narang, M.; Suzuki, H.; Cao, G.; Gao, J.; and Bisk, Y. 2022. Webqa: Multihop and multimodal qa. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 16495–16504.

Chen, S.; Guhur, P.-L.; Schmid, C.; and Laptev, I. 2021. History aware multimodal transformer for vision-and-language navigation. *Proc. Adv. Neural Inf. Process. Syst.*, 5834–5847.

Chen, S.; Guhur, P.-L.; Tapaswi, M.; Schmid, C.; and Laptev, I. 2022. Think Global, Act Local: Dual-scale Graph Transformer for Vision-and-Language Navigation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 16537–16547.

Coppersmith, G.; and Kelly, E. 2014. Dynamic wordclouds and vennclouds for exploratory data analysis. In *Proc. Worksh. Interactive Language Learn. Visualization Interfaces*, 22–29.

Das, A.; Datta, S.; Gkioxari, G.; Lee, S.; Parikh, D.; and Batra, D. 2018. Embodied question answering. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1–10.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2023. Mind2Web: Towards a Generalist Agent for the Web. *arXiv preprint arXiv:2306.06070.*

Furuta, H.; Nachum, O.; Lee, K.-H.; Matsuo, Y.; Gu, S. S.; and Gur, I. 2023. Multimodal Web Navigation with Instruction-Finetuned Foundation Models. *arXiv preprint arXiv:2305.11854.*

Gao, C.; Chen, J.; Liu, S.; Wang, L.; Zhang, Q.; and Wu, Q. 2021. Room-and-object aware knowledge reasoning for remote embodied referring expression. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 3064–3073.

Hao, W.; Li, C.; Li, X.; Carin, L.; and Gao, J. 2020. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 13137–13146.

He, K.; Huang, Y.; Wu, Q.; Yang, J.; An, D.; Sima, S.; and Wang, L. 2021a. Landmark-RxR: Solving Vision-and-Language Navigation with Fine-Grained Alignment Supervision. *Proc. Adv. Neural Inf. Process. Syst.*, 652–663.

He, Z.; Sunkara, S.; Zang, X.; Xu, Y.; Liu, L.; Wichers, N.; Schubiner, G.; Lee, R.; and Chen, J. 2021b. Actionbert: Leveraging user actions for semantic understanding of user interfaces. In *Proc. AAAI Conf. Artif. Intell.*, 5931–5938.

Hong, Y.; Rodriguez-Opazo, C.; Wu, Q.; and Gould, S. 2020. Sub-Instruction Aware Vision-and-Language Navigation. *arXiv preprint arXiv:2004.02707.*

Hong, Y.; Wu, Q.; Qi, Y.; Rodriguez-Opazo, C.; and Gould, S. 2021. VLN↻BERT: A recurrent vision-and-language bert for navigation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1643–1653.

Hsiao, Y.-C.; Zubach, F.; Wang, M.; et al. 2022. ScreenQA: Large-Scale Question-Answer Pairs over Mobile App Screenshots. *arXiv preprint arXiv:2209.08199.*

Jia, S.; Kiros, J.; and Ba, J. 2019. Dom-q-net: Grounded rl on structured language. *Proc. Int. Conf. Learn. Represent.*

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597.*

Li, X.; Zhang, Y.; Yuan, W.; and Luo, J. 2022. Incorporating External Knowledge Reasoning for Vision-and-Language Navigation with Assistant's Help. *Applied Sciences*, 7053.

Li, Y.; He, J.; Zhou, X.; Zhang, Y.; and Baldridge, J. 2020. Mapping natural language instructions to mobile UI action sequences. *Proc. Annu. Meeting Assoc. Comput. Linguist.*

Liu, E. Z.; Guu, K.; Pasupat, P.; Shi, T.; and Liang, P. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. *Proc. Int. Conf. Learn. Represent.*

Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; et al. 2023. AgentBench: Evaluating LLMs as Agents. *arXiv preprint arXiv:2308.03688.*

Malinowski, M.; and Fritz, M. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. *Proc. Adv. Neural Inf. Process. Syst.*

Mazumder, S.; and Riva, O. 2020. Flin: A flexible natural language interface for web navigation. *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*

Qi, Y.; Pan, Z.; Hong, Y.; Yang, M.-H.; van den Hengel, A.; and Wu, Q. 2021. The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In *Proc. IEEE Int. Conf. Comput. Vis.*, 1655–1664.

Qi, Y.; Pan, Z.; Zhang, S.; Hengel, A. v. d.; and Wu, Q. 2020a. Object-and-action aware model for visual language navigation. In *Proc. Eur. Conf. Comput. Vis.*, 303–317.

Qi, Y.; Wu, Q.; Anderson, P.; Wang, X.; Wang, W. Y.; Shen, C.; and Hengel, A. v. d. 2020b. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 9982–9991.

Qiao, Y.; Qi, Y.; Hong, Y.; Yu, Z.; Wang, P.; and Wu, Q. 2022. HOP: History-and-Order Aware Pre-training for Vision-and-Language Navigation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 15418–15427.

Qiao, Y.; Qi, Y.; Hong, Y.; Yu, Z.; Wang, P.; and Wu, Q. 2023. HOP+: History-enhanced and Order-aware Pre-training for Vision-and-Language Navigation. *IEEE Trans. Pattern Anal. Mach. Intell.*

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 5485–5551.

Sun, L.; Chen, X.; Chen, L.; Dai, T.; Zhu, Z.; and Yu, K. 2022. META-GUI: Towards Multi-modal Conversational Agents on Mobile GUI. *Proc. Conf. Empirical Methods Natural Lang. Process.*

Tan, H.; and Bansal, M. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *Proc. Conf. Empirical Methods Natural Lang. Process.*

Toyama, D.; Hamel, P.; Gergely, A.; Comanici, G.; Glaese, A.; Ahmed, Z.; Jackson, T.; Mourad, S.; and Precup, D. 2021. Androidenv: A reinforcement learning platform for android. *arXiv preprint arXiv:2105.13231*.

Williams, R. J.; and Zipser, D. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2): 270–280.

Wu, Z.; and Palmer, M. 1994. Verb semantics and lexical selection. *Proc. Annu. Meeting Assoc. Comput. Linguist.*

Xu, N.; Masling, S.; Du, M.; Campagna, G.; Heck, L.; Landay, J.; and Lam, M. S. 2021. Grounding open-domain instructions to automate web support tasks. *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*

Yao, S.; Chen, H.; Yang, J.; and Narasimhan, K. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Proc. Adv. Neural Inf. Process. Syst.*, 20744–20757.

Zhao, C.; Qi, Y.; and Wu, Q. 2023. Mind the Gap: Improving Success Rate of Vision-and-Language Navigation by Revisiting Oracle Success Routes. In *Proc. ACM Int. Conf. Multimedia*, 4349–4358.

Zhou, G.; Hong, Y.; and Wu, Q. 2023. NavGPT: Explicit Reasoning in Vision-and-Language Navigation with Large Language Models. *arXiv preprint arXiv:2305.16986*.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Bisk, Y.; Fried, D.; Alon, U.; et al. 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents. *arXiv preprint arXiv:2307.13854*.

Zhu, W.; Hu, H.; Chen, J.; Deng, Z.; Jain, V.; Ie, E.; and Sha, F. 2020. BabyWalk: Going Farther in Vision-and-Language Navigation by Taking Baby Steps. In *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2539–2556.