SummAct: Uncovering User Intentions Through Interactive Behaviour Summarisation

Guanhua Zhang
University of Stuttgart
Stuttgart, Germany
Guanhua.Zhang@vis.uni-stuttgart.de

Mohamed Adel Naguib Ahmed
University of Stuttgart
Stuttgart, Germany
st178773@stud.uni-stuttgart.de

Zhiming Hu*
University of Stuttgart
Stuttgart, Germany
zhiming.hu@vis.uni-stuttgart.de

Andreas Bulling
University of Stuttgart
Stuttgart, Germany
andreas.bulling@vis.uni-stuttgart.de

Abstract

Recent work has highlighted the potential of modelling interactive behaviour analogously to natural language. We propose *interactive behaviour summarisation* as a novel computational task and demonstrate its usefulness for automatically uncovering latent user goals while interacting with graphical user interfaces. We introduce *SummAct* – a novel hierarchical method to summarise low-level input actions into high-level goals to tackle this task. SummAct first identifies sub-goals from user actions using a large language model and in-context learning. In a second step, high-level goals are obtained by fine-tuning the model using a novel UI element weighting mechanism to preserve detailed context information embedded within UI elements during summarisation. Through a series of evaluations, we demonstrate that SummAct significantly outperforms baseline methods across desktop and mobile user interfaces and interactive tasks by up to 21.9%. We further introduce two exciting example use cases enabled by our method: interactive behaviour forecasting and automatic behaviour synonym identification.

CCS Concepts

• **Computing methodologies** → **Machine learning**; **Artificial intelligence**; • **Human-centered computing** → **Human computer interaction (HCI)**.

Keywords

Interactive behaviour, Goal recognition, Large language model, Next action prediction

ACM Reference Format:
Guanhua Zhang, Mohamed Adel Naguib Ahmed, Zhiming Hu, and Andreas Bulling. 2025. SummAct: Uncovering User Intentions Through Interactive Behaviour Summarisation. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3706598.3713190

*Corresponding author

CHI '25, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713190

1 Introduction

Recent work has demonstrated that users' interactive behaviour, e.g., when interacting with graphical user interfaces using the mouse or keyboard, shares similarities with the sequential and hierarchical nature of natural language [56]. In parallel, an increasing number of works have started to describe interactive behaviour using natural language and process it using large language models [13, 24, 32, 34, 53]. One key advantage of this language perspective is facilitating a more interpretable analysis and understanding of interactive behaviour, thus enabling novel paradigms for solving human-computer interaction (HCI) tasks.

Among these tasks, understanding users' goals is key to intelligent interactive systems and anticipatory user interfaces [23, 56]. Recognising goals based on the user's behaviour history has been widely studied and applied in HCI, including for unintentional error detection [1], next action prediction [4], or task automation [25, 53, 61]. Despite its potential for HCI and promising first results, predicting users' goals from their interactive behaviour remains challenging, partly due to human behaviour's high variability and complexity. Previous works typically assumed a pre-defined and fixed set of goals and treated goal recognition as a classification task. However, this approach neither captures the wide variety of user goals in everyday scenarios nor can it robustly adapt to unseen or context-dependent goals [55]. It can also result in misinterpretations when users' needs do not align with predefined goal categories, which often happens in real-world applications [57].

In this work, we take inspiration from text and image summarisation tasks studied in natural language processing and computer vision. These tasks involve summarising long text or complex videos into a concise sentence description. Similarly, we formulate goal recognition as an *interactive behaviour summarisation* task: Human interactive behaviour is to be summarised into a sentence, i.e., a natural language description of users' underlying interactive goals. In contrast to existing methods, interactive behaviour summarisation enables recognising an open-ended set of goals. Furthermore, it allows for capturing more flexible and varied interaction goals and handling goals not seen during training.

To address this task, we propose *SummAct* – a novel large language model (LLM)-based method that uses a hierarchical interactive behaviour summarisation process: the method first summarises *low-level* actions into *mid-level* sub-goals, then uses them to augment the input, and finally summarise behaviour into a *high-level* goal. This work focuses on interactive behaviour at the user interface (UI) element level. Each input action sample consists of the interacted UI element and the user's operation on this element (e.g., click or select). The UI element information includes its category (e.g., button or combo box), inherent (e.g., the name or the visible text on a button), and additional content (further values that users are interested in and pick, e.g., a value selected from a combo box). On these input actions, SummAct first generates sub-goals using in-context learning via a pre-trained, frozen LLM due to the lack of ground-truth annotations. It then fine-tunes the LLM to produce the final summary. During fine-tuning, we further propose a *UI element weighting mechanism* that assigns higher weights to the UI element contents, thereby preserving the detailed context information embedded within these elements. This is crucial for accurately interpreting goals that exhibit subtle differences and for certain use cases like behaviour forecasting.

We evaluate SummAct on two datasets that cover a desktop (Mind2Web [13]) and a mobile (MoTIF [6]) interaction setting. We show that SummAct can accurately uncover the goals underlying user actions, with a sentence embedding cosine similarity of up to 0.842 compared to the ground-truth goals. We also demonstrate the importance of our design choices with the full SummAct model significantly outperforming ablated versions by up to 21.9% in cosine similarity. We finally introduce two exciting use cases enabled by interactive behaviour summarisation: 1) providing contextual information of user goals to enhance behaviour forecasting for anticipatory user interfaces and 2) automatically identifying behaviour synonyms to understand user preferences, interaction strategies, system usability and common design patterns.
In summary, the specific contributions of our work are three-fold:

- We formulate goal recognition as the novel open-ended task of summarising interactive behaviour into natural language descriptions. This formulation overcomes existing limitations associated with pre-defined goal sets and improves generalisability to unseen goals. Towards this task, we propose an LLM-based method, SummAct[1] incorporating two distinct and novel designs – hierarchical summarisation and UI element weighting mechanism.
- We show the effectiveness of these designs, and SummAct in general, for interactive behaviour summarisation, through a series of evaluations on two datasets covering desktop and mobile interaction settings.
- We demonstrate the potential of interactive behaviour summarisation for two example use cases: interactive behaviour forecasting and identifying behaviour synonyms. These are widely relevant in HCI, particularly for developing intelligent interactive systems or UI optimisation.

[1]https://collaborative-ai.org/publications/zhang25_chi/

2 Related work

We discuss related work on (1) understanding user goals behind interactive behaviour, (2) large language models for interactive behaviour modelling, and (3) summarising non-language data.

2.1 Understanding User Goals Behind Interactive Behaviour

Recognising users' goals (specific tasks users intend to perform) from their interactive behaviour is key to intelligent user interfaces, which can proactively support users by automatically adjusting the UIs or providing action recommendations [19, 57]. Therefore, an increasing number of works in HCI field have studied automatic goal prediction from interactive behaviour. For example, in virtual reality (VR) interactive environments, David-John et al. [12] recognised user goals of selecting an item from their gaze behaviour, which is necessary for building adaptive interfaces to reduce users' physical and cognitive workload. Hu et al. [23] built a model based on convolutional neural networks and bidirectional gated recurrent units on user eye and head movements to recognise four goals (Free viewing, Visual search, Saliency, and Track), providing insights into different human visual attention under different VR goals. Researchers have also developed various goal recognition methods based on user actions in more pervasive, daily scenarios such as interacting with personal computers or mobile devices. These actions were captured through different modalities, including mouse movements and clicks, keyboard typing, eye tracking and touch interactions. For instance, Elbahi et al. [15] recognised which e-learning task the users performed from mouse movement to help learners quickly interact with the new platform. Koldijk et al. [28] developed different machine learning models to recognise office tasks from users' mouse and keyboard actions. The recognition results provided workers with overviews of performing tasks that could benefit their self-management. Zhang et al. [57] proposed a multimodal random forest-based approach to recognise text formatting goals from users' mouse, keyboard and gaze actions, showing the potential of creating proactive text editors. In the mobile interaction settings, Xu et al. [54] identified intentional versus unintentional touches from gaze, head and screen touch behaviour, enabling a more natural interaction.

However, all the above works studied a pre-defined, fixed, and closed set of user goals, which inherently limits the adaptability and scalability of systems to new goals. Inspired by the prior finding that interactive behaviour shares a similar sequential and hierarchical structure with natural language [56], in this paper, we formulate goal recognition as summarising interactive behaviour into a sentence. This attempt accommodates an open-ended set of user goals and thus allows for more flexible and comprehensive interactive behaviour modelling.

2.2 Large Language Models for Interactive Behaviour

LLMs have recently achieved ground-breaking success in HCI research, bringing novel insights and methodology to model user actions for different applications. For example, Liu et al. [38] proposed HintDroid, an LLM-based method using in-context learning to generate hint-text in Android applications based on the user's input

and corresponding UI context. Wang et al. [49] used a pre-trained LLM to investigate conversational interactions with mobile user interfaces via prompt engineering and zero-shot learning. Their results demonstrated the potential of using LLMs for language-based mobile interactions. Huang et al. [24] applied pre-trained LLMs and a chain-of-thought technique to extract macros from mobile interaction traces in existing datasets. Other research focuses on building LLM-based automatic agents that navigate through interactive systems and complete pre-defined tasks [10, 34, 53, 62]. For instance, Deng et al. proposed MindAct [13] to perform given tasks in complex web environments automatically. MindAct first fine-tuned a language model to rank all the UI elements available on the web page based on the task description and action history. Selecting the top-ranked candidates, MindAct then formulated task automation as a multi-choice question-answering task and used in-context learning for task automation.

Despite the acknowledged potential of LLMs in interactive behaviour modelling, their application in goal recognition remains largely under-explored. In this paper, we approach goal recognition through an interactive behaviour summarisation task and propose an LLM-based method, SummAct, to address this task.

2.3 Summarising Non-language Data

In language processing, summarisation has been widely studied and applied in condensing large amounts of information into concise sentences, enabling efficient content consumption across various domains, such as documents [14, 37], code [21], and speech [47]. Inspired by the success of these works, researchers have started to summarise non-language data into coherent textual descriptions for quick and accessible understanding of complex data [49]. For instance, image captioning enhances the comprehension of the images and meanwhile enables text-based image retrieval [48]. Kawamura et al. [27] proposed a multimodal method to summarise lecture videos using audio transcripts, on-screen images and texts, enabling users to obtain information from lengthy video content effectively. Lin et al. [36] and Chen et al. [8] summarised human motion videos, which not only enhanced the understanding of the motion sequence but also had the potential to allow controllable text-to-motion generation. Chen et al. [9] annotated natural-language explanations for fixations in scan paths, providing insights into implicit gaze behaviour change and benefiting explainable scan path prediction. In HCI, researchers have studied the summarisation of graphical user interfaces. Wang et al. [50] summarised core information of mobile UI screens into natural language via the proposed multimodal method, Scree2Words, integrating the text, image, structures and UI semantics. They showcased that the summarisation could potentially be used for language-based UI retrieval, enhancing screen readers and screen indexing for conversational mobile interactions.

These applications highlight the transformative potential of applying summarisation techniques to diverse data modalities. Building upon this, our work introduces a novel method to summarise the complex interactive behaviour into human-interpretable natural language sentences, which reflect users' latent goals. Additionally, we present two example use cases enabled by interactive behaviour summarisation, interactive behaviour forecasting and identifying behaviour synonyms, that are widely relevant for intelligent interactive systems and user interface optimisation.

3 Interactive Behaviour Summarisation using SummAct

Building on recent advances demonstrating the potential of analysing interactive behaviour similarly to natural language [24, 34, 56], our method SummAct addresses the novel task of interactive behaviour summarisation. Input to SummAct is a sequence of UI element-level action events the user triggered while interacting with a graphical user interface. Every action consists of the UI element the user has interacted with and the operation performed on this element (e.g., click or select). The UI element contains the information of its category (e.g., button or combo box), the inherent (e.g., its name or the text on it) and additional content (the specific value the user is interested in, e.g., the value selected from a combo box). The output of our method is a natural language sentence that concisely summarises their interactive behaviour and, as we show here, their latent interaction goals. Interactive behaviour summarisation allows us to recognise an open-ended set of goals, including those not seen during training. This is in stark contrast to existing methods for the classification of goals that are limited to a closed and predefined set of possible goals [15, 16, 58]. As such, SummAct can provide a more comprehensive and generalisable understanding of interactive behaviour.

Figure 1 provides an overview of SummAct's hierarchical approach to interactive behaviour summarisation: Given a sequence of user input actions encoded in natural language descriptions, SummAct first summarises these low-level actions into a set of sub-goals. Due to the absence of ground-truth data, we used expert annotations in combination with in-context learning to adapt a pre-trained, frozen LLM to generate sub-goals. In the second step, these sub-goals are combined with the original actions and summarised into high-level goals via fine-tuning the LLM. To preserve UI element content (indicated in bold in Figure 1) in the summary, our method uses a novel UI element weighting mechanism during fine-tuning. Two previous findings inspired this hierarchical approach: Hierarchical modelling of language data can robustly handle extensive and complex input, such as long documents [37]; and interactive behaviour has an inherent hierarchical nature similar to that observed in natural language [56]. In the following, we describe each of these steps in more detail.

3.1 Sub-goal Generation

The first step involves generating sub-goals from low-level input actions. As shown in Figure 1, actions marked in the same colour are summarised into the same sub-goal, which later becomes a phrase integrated into the overall goal. Given the lack of HCI datasets offering annotations of interaction sub-goals, we used in-context learning. In-context learning involves giving an LLM a small set of examples presented within the context (the prompt) at inference time to guide its response [52]. This approach leverages LLM's ability to understand and adapt to patterns presented in the immediate context of the query without the need to fine-tune the model. To obtain these examples, we asked three HCI, GUI, and behaviour modelling experts to annotate the sub-goals on five samples from

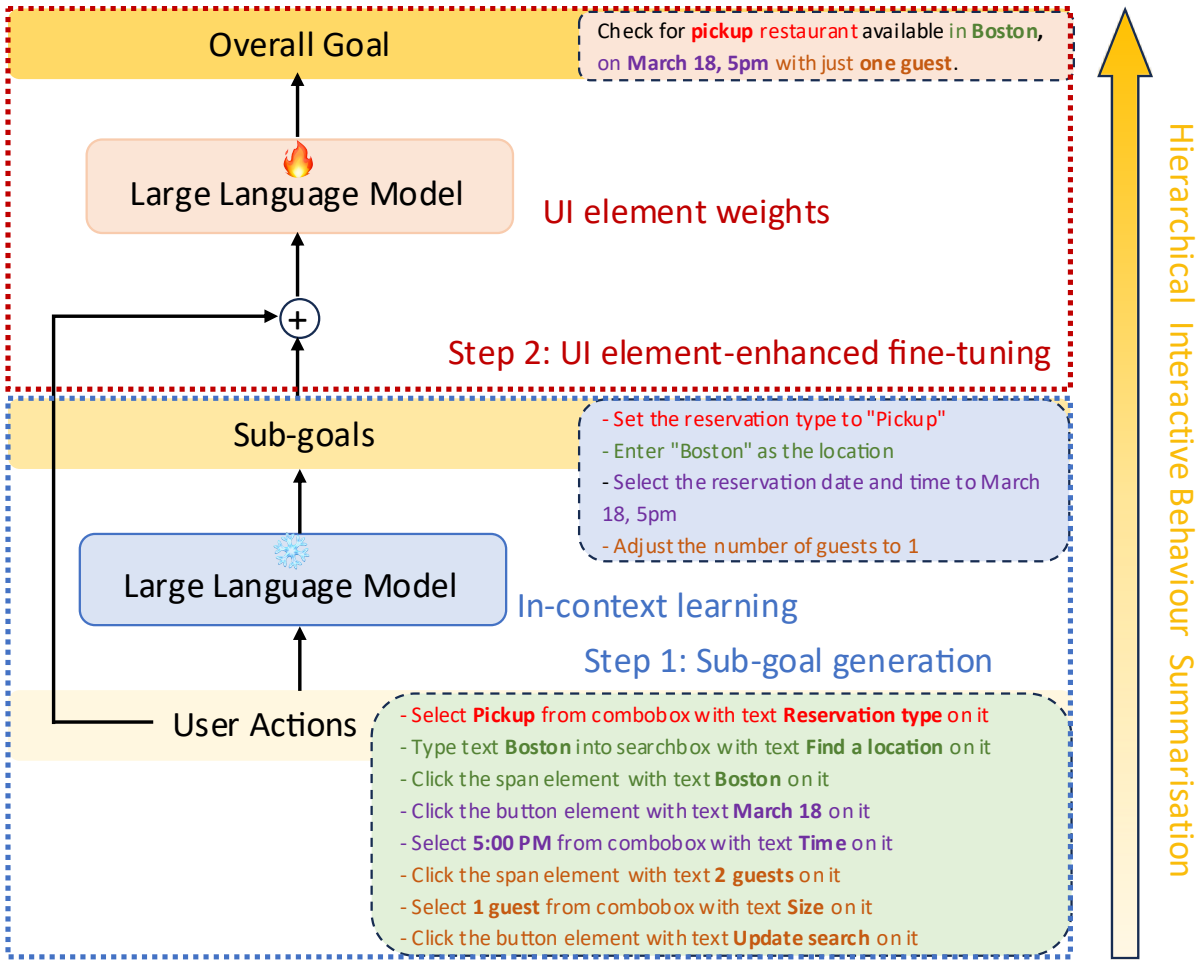


Figure 1: Overview of SummAct for uncovering user goals during user interface interactions through interactive behaviour summarisation. SummAct employs a hierarchical process that initially generates sub-goals and produces the overall goal in natural language. The input is a sequence of user actions, including the interacted UI element and the user's operation on this element. SummAct uses in-context learning to infer an arbitrary number of sub-goals using a pre-trained, frozen LLM (Step 1) and then fine-tunes the LLM while introducing a UI element weighting mechanism (Step 2) to keep detailed context embedded in UI element contents, as highlighted in bold. Actions in the same colour are summarised into the same sub-goal and then to a phrase in the overall goal. The output summary reflects the latent goals that underlie these actions.

the training set collaboratively and reach agreements on the annotation results [33]. These samples are five different action sequences completing five different tasks. In Appendix B.1, we provide the used prompt, including the example of sub-goal annotation.

3.2 UI Element-enhanced Fine-tuning

In the second step, we fine-tuned the LLM to summarise the overall goal from the generated mid-level sub-goals and the original low-level actions. In Appendix B.2, we provide sample prompts for this fine-tuning step. LLMs are typically structured as sequence-to-sequence models, i.e. they are trained to generate output sequences based on input sequences, such as summarising an input. Therefore, LLMs are commonly trained using a next token prediction task in a teacher-forcing setup, where the model is guided by a ground-truth token rather than the previously predicted token to predict the next token [46]. This training strategy helps stabilise the training process and accelerates convergence by reducing the propagation of errors through the sequence [49]. Thus, based on the input prompt, LLMs iteratively predict the next token and continually update their predictions as each new token is added to the output sequence. Next token prediction is formulated as a classification task and thus uses a cross-entropy loss $L_{NextToken}$ [30]. For the j-th token in the input sequence, this loss is calculated as

$$L_{NextToken_j} = log(\mathbb{P}_\theta(Token_j|Token_1, ..., Token_{j-1})) \quad (1)$$

where θ are the model parameters.

In preliminary experiments, we found that fine-tuning the LLM only using next token prediction led to UI element content getting

excluded from the final interactive behaviour summary. This may be because the model tends to rely on frequent patterns of general natural language rather than focus on task-specific information embedded in the UI elements [2]. Figure 1 shows examples of such information related to the interactive behaviour summarisation task (highlighted in bold). Let us consider the first input action ("Select **Pickup** from combo box with text **Reservation type** on it") as an example: "Reservation type" is the name of the combo box, i.e., the inherent content of the combo box, representing what this combo box is about; while "Pickup" is an additional content of the combo box, namely one value the combo box provides that the user is interested in and ultimately selects. Retaining such UI element contents in the final summaries is particularly important for interactive behaviour summarisation: First, such content provides interactive context information necessary to distinguish between subtle goals. For instance, actions include selecting "1" from a combo box named "guest number" on a booking site by some users versus selecting more guests by other users. However, these detailed contents are ignored, and the summarised goals are both *finding a hotel room*; the system may further inaccurately suggest unsuitable accommodations, e.g., family rooms for solo travellers and vice versa, causing decreased usability and potential frustration. Second, they are important for downstream applications, such as behaviour forecasting, to ensure the prediction is relevant and consistent with the current context and underlying goals [53]. For example, if a user clicks on a button named "gluten-free" but this content is overlooked, the interactive system may mistakenly predict the upcoming actions to involve browsing or purchasing products containing gluten, leading to a worse user experience.

To address this challenge we propose a *UI element weighting mechanism* to enhance the fine-tuning process by guiding the model to focus on these contents. This is similar to ensuring that a text summary covers essential keywords in natural language processing [14]. Specifically, for the i-th training sample, we create a weight vector $\mathbf{K}_i$, where each component K_{ij} denotes the weight assigned to the j-th token in the ground-truth summary:

$$K_{ij} = \begin{cases} \lambda & \text{if } \ Token_j \in Token_{Detail} \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

As such, tokens that contain action details ($Token_{Detail}$) receive λ times the weight compared to other tokens. We empirically set $\lambda = 2$ in our experiments. The overall fine-tuning loss integrating this weighted mechanism $\mathcal{L}_{Enhanced}$ is then computed as a weighted version of the original cross-entropy loss:

$$\mathcal{L}_{Enhanced} = \mathbf{K} \circ \mathcal{L}_{NextToken} \tag{3}$$

3.3 Implementation

We opted for the lightweight open-source Mistral-7B model [26] as the LLM backbone, known for its efficiency and effectiveness in handling various NLP tasks. Mistral-7B incorporates advanced techniques such as grouped-query attention for fast inference and sliding window attention for managing long sequences and complex contexts. These features contribute to Mistral-7B's superior performance on various benchmarks compared to other state-of-the-art models while using fewer parameters, thus conserving computational resources [44]. We used a batch size of 16 and a maximum input sequence length of 1024, with an initial learning rate of 1e-6. We used the Adam optimiser with $\beta_1 = 0.9$ and $\beta_2 = 0.95$ [46], and a cosine annealing scheduler for a progressive reduction of the learning rate following a cosine curve, a strategy proven to stabilise the training phase [39]. We fine-tuned the model for 15 epochs using eight Tesla V100-SXM2-32GB GPUs, completing the training within ten hours.

4 Experiments

We conducted experiments to evaluate the quality of interactive behaviour summaries generated by SummAct. Given the novelty of this task and the lack of existing baseline methods, we compare the full model with several ablated versions instead. More specifically, starting with using an off-the-shelf, pre-trained LLM – the common practice in HCI research currently [5, 24] – we incrementally add fine-tuning, sub-goal generation, and the UI element weighting mechanism. We report quantitative metrics that measure how similar the generated summaries are compared to the ground truth and qualitative similarities and differences of the generated summaries.

4.1 Datasets

We conducted all evaluations using two prominent datasets that encompass both the desktop (Mind2Web [13]) and mobile (MoTIF [6]) interaction contexts. These datasets are extensively used to understand and model user interfaces and interactive behaviours [5, 62]. We used the text instructions assigned to participants in the two datasets as ground-truth overall goals. They encode interactive behaviour as user actions to achieve specified interaction objectives. Each user action is annotated with information related to the UI element (category and content) and the user operation (e.g., click or swipe) associated with this element.

4.1.1 Mind2Web. This dataset provides crowdsourced actions across 2,350 tasks performed on 137 real-world websites (e.g., Booking, Uniqlo, IMDB) spanning 31 domains (e.g., travel, shopping, entertainment). As such, this dataset offers a wide variety of user actions and goals and allows us to evaluate the performance of SummAct in real-world scenarios. Mind2Web was divided by its authors into a training set as well as a testing set, which comprised three testing subsets: 1) *cross-domain* includes data instances from different domains, e.g., shopping vs travel; 2) *cross-website* includes instances from unseen websites, e.g., Booking vs Airbnb; and 3) *cross-task* includes unseen tasks, e.g., booking a flight vs buying a shirt. We followed this train-test split in our experiments. Mind2Web provided symbolic representations of actions (see examples in Table 3). To leverage the capacity of LLMs and create a uniform input format for SummAct, we additionally converted these raw action strings into natural language descriptions using a transformation template [41]. Our evaluations verified the effectiveness of this preprocessing (see Appendix A for more details).

4.1.2 MoTIF. This dataset targets a mobile interaction setting comprising screen touch data collected on 756 different tasks across 125 Android applications. The dataset directly provides synthetic natural language sentences describing each low-level action including the interacted UI element and the user's operation (click, type or swipe) on it. We followed the same way of splitting MoTIF into

training and testing sets (refer to [6] for more details). The testing set includes feasible and infeasible tasks, such as tasks that are too unclear or cannot be completed in the given App. We only used the feasible tasks for our evaluations to ensure that user actions reliably reflect the corresponding goals.

4.2 Ablations

We compared the full SummAct model **LLM+FT+SubGoal+Weight** (fine-tuned LLM using both the input actions and sub-goals with $\mathcal{L}_{Enhanced}$ for summarisation) with several ablated versions to evaluate the impact of the different modifications. All methods used the same LLM and prompt templates to ensure a fair comparison.

- **LLM**: pre-trained LLM using only the input actions.
- **LLM+SubGoal**: pre-trained LLM using both the input actions and the sub-goals.
- **LLM+FT**: fine-tuned LLM using only the input actions with $\mathcal{L}_{NextToken}$ for summarisation.
- **LLM+FT+Weight**: fine-tuned LLM using only the input actions with $\mathcal{L}_{Enhanced}$ for summarisation.
- **LLM+FT+SubGoal**: fine-tuned LLM using both input actions and sub-goals with $\mathcal{L}_{NextToken}$ for summarisation.

Since the UI element weighting mechanism is specifically incorporated into the loss function of fine-tuning, we do not have a standalone version of the pre-trained LLM enhanced solely by the weighting mechanism, i.e., LLM+Weight.

4.3 Quantitative Evaluations

We first quantify the similarity between ground truth and summarised goals for all methods with four widely used NLP metrics [49, 60]. Specifically, we report Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [35], Metric for Evaluation of Translation with Explicit ORdering (METEOR) [3], and Bilingual Evaluation Understudy (BLEU) [43] that count the overlapping n-grams between texts to assess their similarity, thus providing a measure that reflects lexical precision and recall. We further report an embedding-based metric using a state-of-the-art sentence encoder, Sentence Transformer[2], to obtain the sentence embeddings. Embedding-based metrics evaluate the cosine similarity between sentences and capture deeper and more robust semantic meanings that go beyond mere lexical matches [45]. All of these metrics indicate better results as their values increase.

Table 1 provides an overview of the results of this comparison. As can be seen from the table, our proposed full model consistently outperforms the ablated versions on all test sets and all metrics, obtaining a cosine similarity of up to 0.842 with the ground-truth user goals. Among the three generalisation test sets from Mind2Web, SummAct performed better in *cross-website* and worse in the *cross-domain* setting. The former is likely because different websites within the same domain share similar UI designs and thus require similar navigation patterns [13], which can be efficiently captured and integrated by our SummAct. These results also show that generalisation across domains remains challenging due to the variations in context and user interactions.

[2]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

We can also see from Table 1 that directly using a pre-trained LLM performs the worst while adding fine-tuning, sub-goals, or the UI element weighting mechanism improved performance notably. Although adding sub-goals (LLM+SubGoal) increased cosine similarity by up to 13.3% (0.203 vs 0.230 on MoTIF), the largest performance increase was achieved when adding fine-tuning (LLM vs LLM+FT), where the cosine similarity improved by up to 89.9% (0.348 vs 0.661, cross-task setting) on Mind2Web, and 240.4% (0.203 vs 0.691) on MoTIF.

Also, adding our two novel designs of sub-goals and UI element weighting mechanism contributes to the effectiveness of SummAct (LLM+FT vs LLM+FT+SubGoal+Weight), together leading to an up to 21.9% improvement on the cosine similarity (0.691 vs 0.842 on MoTIF). Comparing our method with LLM+FT+Weight, the UI element weighting mechanism increased the cosine similarity on Mind2Web by 7.1% (0.705 vs 0.755) in the cross-domain setting, 7.6% (0.740 vs 0.796) in the cross-task setting, 6.1% (0.756 vs 0.802) in the cross-website setting, and by 7.3% (0.785 vs 0.842) on MoTIF. On n-gram-based metrics, SummAct obtained improvements of up to 53.0% (0.296 vs 0.453 on Mind2Web cross-task setting) on BLEU, 16.7% (0.383 vs 0.447 on Mind2Web cross-website setting) on ROUGE, and 28.3% (0.293 vs 0.376 on Mind2Web cross-domain setting) on METEOR. In Section 5.1, we further show that a lack of the UI contents harms performance for behaviour forecasting. Similarly, SummAct outperformed its ablated version that removed the sub-goals (LLM+FT+Weight), where the cosine similarity increased by 5.2% (0.718 vs 0.755) in Mind2Web cross-domain setting, 5.7% (0.753 vs 0.796) in the cross-task setting, 5.1% (0.763 vs 0.802) in the cross-website setting and 11.4% (0.756 vs 0.842) on MoTIF. Moreover, BLEU improved by up to 50.5% (0.301 vs. 0.453), achieved on Mind2Web cross-task setting; the maximum enhancement of ROUGE 15.97% (0.689 vs. 0.799) on MoTIF; while the largest improvement of METEOR reached 29.5% (0.332 vs. 0.430), obtained in Mind2Web cross-task setting. Taken together, these evaluations show the effectiveness of the proposed components for interactive behaviour summarisation.

4.4 Qualitative Analysis

We further examined the summaries generated by SummAct and its ablations qualitatively to understand the impact of the different designs.

4.4.1 Impact of UI element weighting Mechanism. Compared to the summaries generated by the full SummAct implementation, the ablated version without UI element weighting mechanism lacks detailed context information embedded in UI element contents. For example, the ground-truth goal to *Find a **campground** in Orlando for two adults to check in on **Mar 29** and check out on **Mar 30*** was correctly summarised by SummAct as *Find a **campground** in Orlando for two adults from **March 29** to **March 30***. On the contrary, the ablated version (LLM+FT+SubGoal) produced a less accurate summary, *Find **hotels** in Orlando for two adults in **March***, missing the information of the precise dates and the specific type of accommodation. This issue arose because during the summarisation, the ablation ignored the detailed content in a clicking action on the button of "Find a KOA", which specified the accommodation type as a campground instead of any hotel.

Method	Metric	Mind2Web			MoTIF
		Cross Domain	Cross Task	Cross Website	
LLM	CosSim	.357	.348	.380	.203
	BLEU	.004	.004	.004	.012
	ROUGE	.050	.056	.060	.077
	METEOR	.126	.132	.146	.093
LLM+SubGoal	CosSim	.381	.374	.404	.230
	BLEU	.006	.004	.006	.009
	ROUGE	.051	.055	.059	.063
	METEOR	.150	.157	.167	.098
LLM+FT	CosSim	.631	.661	.673	.691
	BLEU	.201	.217	.204	.293
	ROUGE	.313	.301	.312	.511
	METEOR	.303	.306	.322	.510
LLM+FT+Weight	CosSim	.705	.740	.756	.785
	BLEU	.349	.296	.305	.407
	ROUGE	.345	.372	.383	.730
	METEOR	.293	.341	.374	.709
LLM+FT+SubGoal	CosSim	.718	.753	.763	.756
	BLEU	.291	.301	.301	.359
	ROUGE	.381	.392	.391	.689
	METEOR	.323	.332	.363	.662
LLM+FT+SubGoal+Weight (SummAct)	CosSim	**.755**	**.796**	**.802**	**.842**
	BLEU	**.406**	**.453**	**.445**	**.453**
	ROUGE	**.390**	**.432**	**.447**	**.799**
	METEOR	**.376**	**.430**	**.454**	**.758**

FT = Fine-tuning; SubGoal = Sub-goal generation; Weight = UI element weighting mechanism

Table 1: Interactive behaviour summarisation results achieved by our proposed SummAct and its ablated versions. The evaluation is conducted on a desktop dataset Mind2Web (including three test subsets for generalisability assessment across domains, tasks and websites) and a mobile dataset MoTIF. We measure the summarisation quality with four metrics: cosine similarity between sentence embeddings, n-gram-based BLEU, ROUGE, and METEOR. The best results are shown in bold.

Another example is the goal *Find a highest rated dealer for Cadillac with a rating above **4 stars** within **20 miles** of **zip 60606***. SummAct effectively summarised this into *Find a highest rated Cadillac dealer above **4 star** within **20 miles** of **60606***. However, the ablation's prediction was *Find the highest rated dealer for Cadillacs*, missing the specific criteria of rating and proximity.

This analysis shows that without the UI element weighting mechanism, although the summaries retain the overall logic, they lack crucial specific information the UI elements provide.

4.4.2 Impact of sub-goals. We then examined the summaries generated by the other ablation (LLM+FT+Weight) in which the sub-goals were removed from SummAct. We found that the summaries retained specific information but often failed to capture the overarching logic or coherence, especially when handling complex, multi-step interactive behaviour. Figure 2 shows two examples of this phenomenon, each with the user's input actions and their underlying goals, the ground-truth goal, the summary generated by the full version of SummAct, and the summary generated by the ablation. For example, at the top, the user browsed through top-trending content within a community about work, then picked one post with the heading "...woman-dominated work..." and saved it. Based on this interactive behaviour, SummAct's summarisation was the same as the ground truth, i.e. *save a rising post on a community about work*. However, the ablation produced *find a post about women-dominated workplace*, focusing disproportionately on specific content cues, such as the particular post's heading, rather than the overall context of these actions. This demonstrates that without sub-goals, summaries may lack the essence and broader context of user interactions and instead focus on specific keywords or aspects, resulting in a summary that does not reflect the overall goal.

In the second example, the user searched for *auto repair* with filtering conditions, selected one particular item, and then read its reviews. This was summarised correctly by SummAct: the summary includes that the user first looks for a business and then verifies its quality according to its reviews. However, the ablation summarised the goal as *find an accredited auto repair shop in zip code 10002 that*

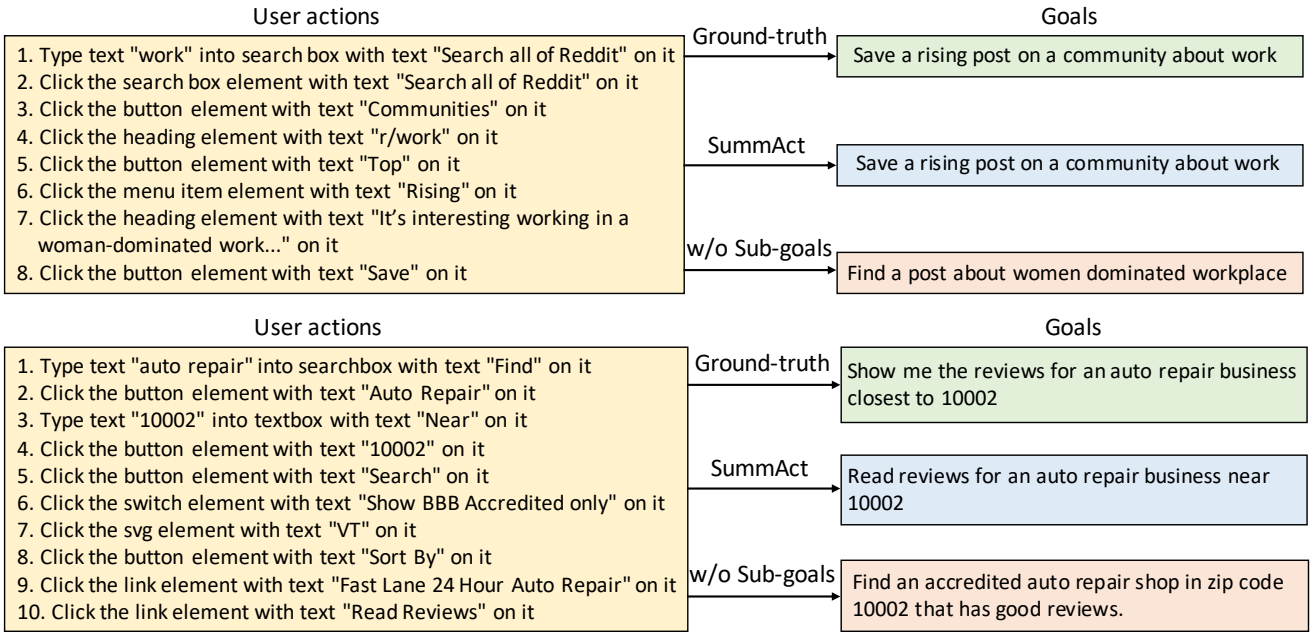


Figure 2: Two examples showing the input user actions, their underlying ground-truth goals and those summarised by the full version of SummAct and its ablation removing sub-goals.

has good reviews, mistakenly understanding the goal of filtering the auto repair business based on their reviews. This occurs because, without sub-goals, the model processes all behaviour information indiscriminately and struggles to dissect the intricate dependencies and hierarchy among input actions. As a result, the ablated model erroneously swapped the sequence priorities between *finding auto repair* and *reading reviews*.

This analysis underscores the importance of our hierarchical approach in handling complex interactive behaviour. By generating intermediate sub-goals, SummAct not only distils key information from different stages of the user actions but also maintains a coherent understanding throughout each interaction stage, ensuring that the final summary encapsulates the overall context.

5 Use Cases of Interactive Behaviour Summarisation

5.1 Interactive Behaviour Forecasting

Interactive behaviour forecasting is core to anticipatory and proactive interactive systems [57]. Specifically, we conducted the next action prediction following [58]. Most current next-action prediction methods are based solely on historical information without explicitly understanding overall goals [19, 29, 59]. The summarisation of action history can potentially enhance the next action prediction by providing contextual information on users' goal trajectories.

We approached next action prediction as a multi-choice question-answering task following [11, 20], where each target action includes two components – the *UI element* users interact with, like a button or link, and the *operation* users apply on this element, like clicking or typing. The model needs to select from a list of candidate UI elements that users may interact with and, meanwhile, predict the corresponding operation. For example, if the predicted action is *B.CLICK*, this means the user may click on the UI element shown as the option *B* in the candidate list.

The pipeline includes three steps [13, 61]: First, we summarise the actions a user has performed using our SummAct. Then, we employ a candidate extraction method proposed by [13] to filter and rank UI elements on the current web page and retain the top-k elements as the candidate targets for the next action. Retaining only top-k elements is because the raw HTML contains a large amount of noisy UI data that can distract LLMs and cause hallucinations [40] and exceeds the maximum length of allowed input tokens. In our experiments, we set k to 50 [13]. Finally, we fine-tune an LLM to predict the next action, i.e., selecting the next target UI element out of the 50 candidates and predicting its corresponding operation out of three classes (click, select or type). We used the same fine-tuning set-up as the summarisation, i.e., Mistral-7B as the backbone LLM and the same learning rate, optimiser and scheduler. We fine-tuned the model for only three epochs, given its fast convergence. We required the behaviour history to include at least five past actions to offer adequate context, consistent with prior next action prediction works [29, 58]. The prompt for fine-tuning the LLM included these past actions, the goal summarised by SummAct, and the list of candidate UI elements (see Appendix B.3).

We compared our results with two baselines, as shown in Table 2. The first is when only using the action history to compare with and examine the effectiveness of summaries in next action prediction. The other is when using the history plus the summary generated by the ablated version of SummAct excluding the UI element weighting

Input	Cross Domain		Cross Task		Cross Website	
	Element	Operation	Element	Operation	Element	Operation
History	31.2	42.1	34.2	46.2	30.6	40.4
History+Summary (*w/o weight*)	37.8	45.9	43.8	47.7	34.9	43.9
History+Summary (*Full*)	**46.8**	**50.5**	**47.8**	**54.5**	**40.1**	**45.0**

Table 2: Next action prediction performance achieved using 1) only behaviour history, 2) behaviour history and summary generated by SummAct without the UI element weighting mechanism, and 3) behaviour history and summary generated by SummAct full version. Each action includes the target UI element and the corresponding operation applied on it (click, select or type). The two metrics are the accuracy of choosing the correct target (Element) and the F1 score of predicting the operation category (Operation), both in percentage. The best results are shown in bold.

mechanism, to check the specific contribution of keeping UI content as discussed in Section 3.2. Following [13, 62], we calculated the accuracy of UI element option, and F1 score of operation category to measure the imbalanced operation classes. We showcased the performance of next action prediction on Mind2Web given that this dataset has more variety of goals and interfaces than MoTIF, as shown in Section 4.1.

As presented in Table 2, integrating summarised goals consistently enhanced the performance across domains, websites and tasks, with an average 12.9% higher element accuracy and 7.1% higher operation F1 score. Moreover, we observed adding the UI element weighting mechanism improved the element accuracy and the operation F1 score by 6.1% and 4.2% on average, respectively, verifying that the UI content preserved by our method was helpful for next action prediction.

Enhancing next action prediction offers practical benefits for various interactive scenarios. For instance, the adaptive user interfaces can dynamically adjust the layout and functionality or directly recommend future actions to users to reduce required cognitive and physical demands and enhance usability [32]. Additionally, this approach allows automation agents to operate more efficiently by intuitively responding to user preferences without requiring explicit task instructions. This potentially leads to smoother, more personalised interactions adapting to evolving user behaviour.

5.2 Automatic Identification of Behaviour Synonyms

Besides directly capturing user goals, our summarisation method can also identify "interactive behaviour synonyms", which, in our examples, are alternative action sequences reflecting the same underlying user goal. We used multiple windows of various lengths from two to the maximum sequence length to segment each input action sequence into sub-sequences. We identified synonyms from sub-sequences instead of the full sequence because, in real-world scenarios, it is common that two long, full sequences are not synonyms (under different overall goals). Still, they may have shared sub-goals that synonym sub-sequences can capture. On each sub-sequence, we used SummAct to summarise its underlying goal into a sentence and used the sentence encoder (the same as Section 4.3) to compute its sentence embedding. Then, we calculated the cosine similarity between these sentence embeddings. If the cosine similarity was higher than a threshold, we considered the synonyms of the two sub-sequences. We set the threshold to 0.7 in our experiments.

Using this approach, we identified three types of synonyms that can provide interesting insights into interaction strategies and system usability: 1) when two synonym behaviours are different but generated from the same UI, the synonyms reflect different strategies or preferences users can take towards a goal; 2) when two synonym behaviours are different and generated from different UIs, the shorter action path indicates better usability; 3) when two synonym behaviours are generated from different UIs but have the same actions, this presents that there are common behaviour patterns or UI designs.

5.2.1 Different behaviours from the same UI. Based on the same goal on the same UI, users can still generate various behaviours, showing different user preferences, interaction habits and strategies. For instance, to *add an item to a new shopping list*, users could choose a shorter action path, i.e., creating a new list when adding an item, or a longer trajectory first navigating to the page showing all the existing lists, adding a list there, then returning to the item page, and finally adding the item. In another example, when the goal was to *find a top-rated restaurant in Miami*, one behaviour directly navigated to a page listing all restaurants and then selected the desired city. In contrast, another user path first identified the city, browsed a broad range of "things to do", and then narrowed down to restaurants. These variations may reflect different user preferences, browsing habits, or the clearness of user goal: the former path shows that the user may have a straightforward motivation to look for a *restaurant*. At the same time, the latter shows that the user may just look for a *place to go* in the city, not necessarily for a restaurant. These synonyms can help designers understand user preferences and habits, find the optimal interaction strategies, and design user-tailored interfaces.

5.2.2 Different behaviours from different UIs. UI designs impact the efficiency of achieving interactive goals, shown by user behaviours. Therefore, the length of the synonyms found through our summarisation can be used as a metric of UI efficiency. Unlike classical metrics like the keystroke-level model (KLM) that measure interactive system's usability via task completion time [18, 22], this metric will compare UIs via the number of actions required for the same goal. As shown in Figure 3, to *add N items into the shopping*

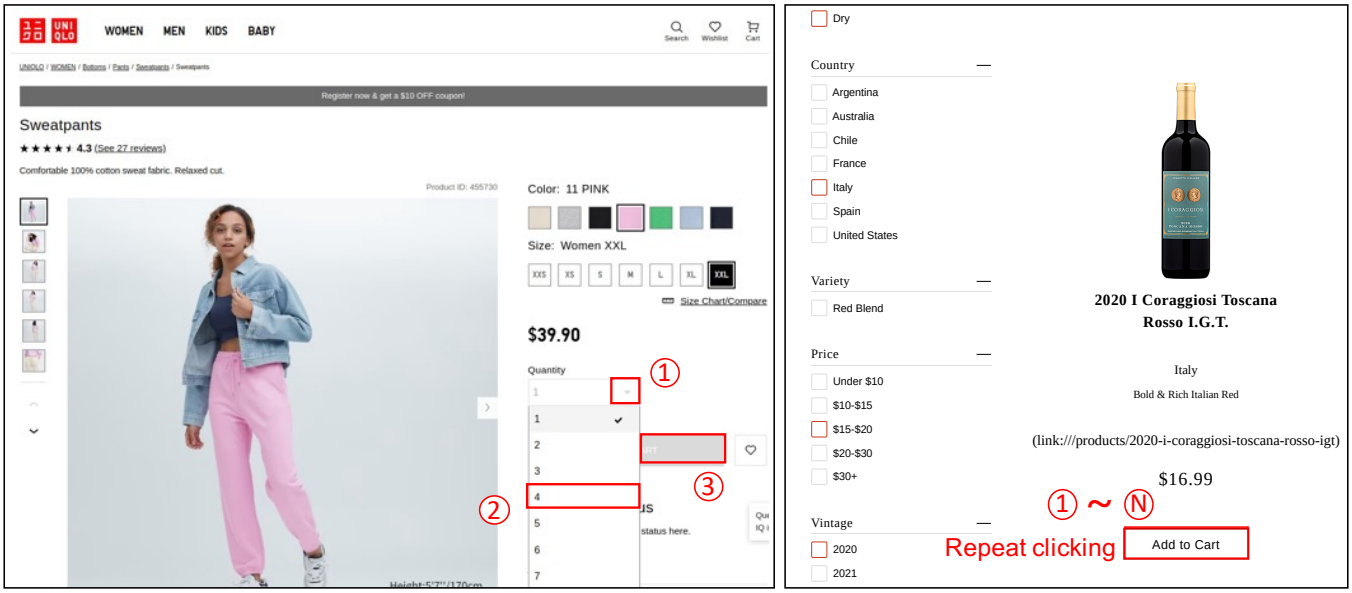


Figure 3: An example of using synonyms to compare UI usability for the task of *adding N items into the shopping cart*. The Uniqlo website (left) allows users to add multiple items with just three clicks, while the Macy's website (right) requires one click per item, leading to more effort and less usability as N increases.

cart: the Uniqlo website (left) enables users to directly choose the total quantity and add all of them to the cart at once, i.e., finishing with three clicks; while Macy's (right) only allows adding one item each time, i.e., with N clicks. When N has a large value, users on the latter interface will have to perform many more actions, harming efficiency and usability. Another example is shown in Figure 4, where the goal was to *search for jobs in a city* (Essex in the upper and London in the lower example). In the upper example, users only needed three clicks to navigate from the main page to view all jobs from the city, i.e., *Countries →England →Jobs in Essex*. On the contrary, in the lower example, users had to perform more actions to see the list of jobs, i.e., *click on Find a job →click on Start now →click on the text box under Where →type London →click on London →click on Search*. The reason is that the former website, Indeed, is specifically designed for job searching, optimising its UI to streamline this function. In contrast, the latter website, Gov.UK, serves multiple functions, not focusing primarily on job hunting, consequently leading to relatively lower efficiency. As such, these types of synonyms can help UX designers optimise interface design to facilitate quicker and more intuitive access, such as creating shortcuts for the interactive goals that are dominant among users.

5.2.3 Same behaviours from different UIs. We found cases where the synonym behaviours followed the same patterns, although in different interactive systems. For example, when the goal was *to book a flight ticket from A to B*, user behaviours on different interfaces, including Kayak, Trip.com, American Airlines, and Expedia, typically followed a uniform process: users clicked flights, typed and selected the departure city or airport, and then typed and selected the destination. When *making an appointment with a doctor* on various medical platforms such as Zocdoc, Mayoclinic.org and Healthgrades, users also employed the same procedure: typed and clicked on the type of specialist, browsed and selected an available doctor, and finally picked the appointment time.

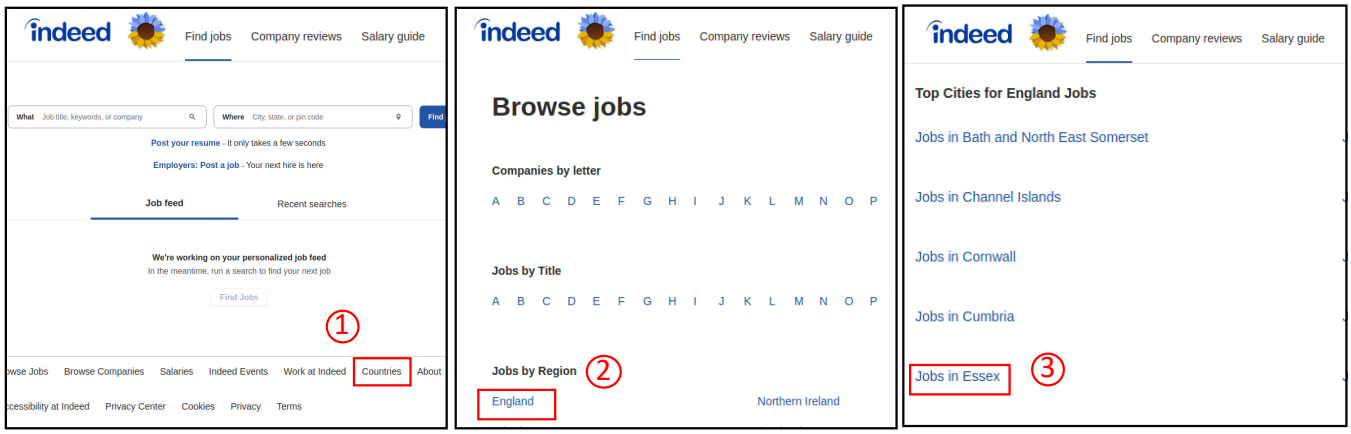
Such common patterns in interactive behaviour are why our model can generalise across different websites and tasks (as shown in Table 1). Understanding these patterns also gives UX designers intuitive starting points to create a new interface that aligns with established user behaviour and expectations. This can reduce the learning curve while ensuring a consistent, friendly user experience.

6 Discussion

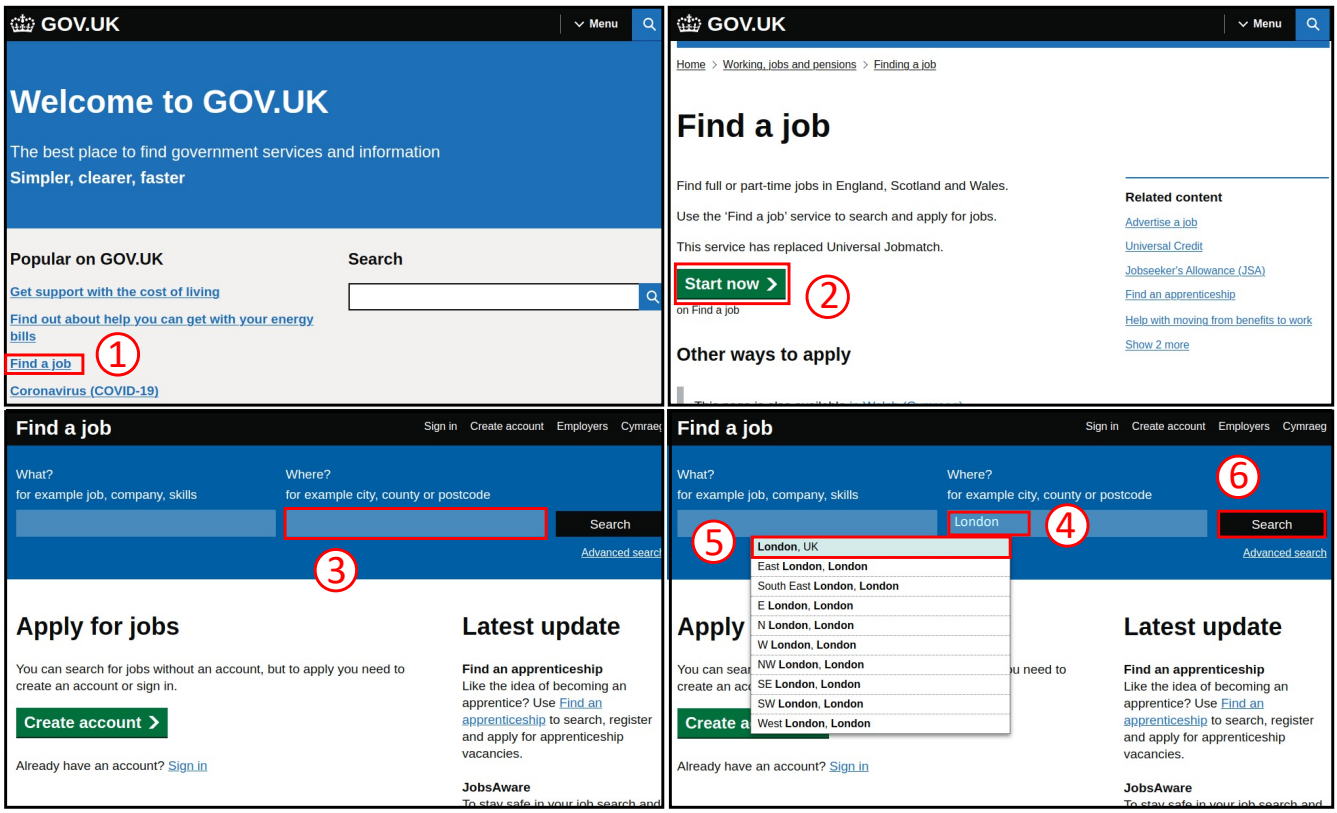
6.1 Interactive Behaviour Summarisation

As a first step of summarising input actions into natural language reflecting interactive goals, our work represents a paradigm shift in goal modelling. The summarisation, within a familiar linguistic framework, lowers the barrier of interpreting and analysing interactive behaviour. The open-ended nature of language offers a more flexible and scalable framework that can 1) eliminate the exhaustive definition of every possible user goal, 2) allow for recognising unseen, out-of-distribution user goals, which was verified by the SummAct's robust performance in the cross-task setting on Mind2Web (shown in Table 1), and 3) adapt to dynamically evolving user needs and hence enable continuous learning and system refinement [51].

The natural language representation also enables language-based behaviour retrieval that will be helpful in question-answering agents and conversational user interfaces. For instance, a system streamlines the diagnostic process by providing quick access to relevant troubleshooting steps based on user-provided problem descriptions and their intended functions. Additionally, when novice



(a) Searching for jobs in Essex on Indeed



(b) Searching for jobs in London on Gov.UK

Figure 4: An example of using synonyms to compare UI usability for the goal of *Searching for jobs in city A*. On the upper interface, the users can finish the task with only three actions; on the lower interface, the users must perform six actions, indicating worse usability.

users struggle to complete an interactive goal, a system can quickly pull up the most relevant teaching actions or tutorials from experts for them to watch and learn.

In addition to its inherent utility, we also demonstrate two example use cases enabled by the summary, namely interactive behaviour forecasting and automatic behaviour synonym identification. Understanding users' goals from past actions is important to forecast future actions along the goal. Therefore, we enhanced current computational next-action prediction methods based on action history

by adding the summary as an additional input. SummAct's summarisation led to better prediction performance, as shown in Table 1. These improvements underscore the effectiveness of adding semantic information through interactive behaviour summarisation. Moreover, if two behaviours have the similar summary, they can be considered synonyms, i.e., alternatives achieving the same goal. Synonyms can provide insights into understanding diverse user preferences and interaction strategies, optimising interactive systems, and finding common patterns in behaviour and UI designs across systems.

6.2 Design of SummAct

We propose a novel LLM-based approach, SummAct, that includes two methodological contributions and design choices to generate natural language summarisation from user actions and their UI context. The first design choice is to employ a hierarchical process adept at handling the complexity and variability of user actions. The method first enriches the semantic context available for the final summary generation by producing intermediate sub-goals from low-level actions. The second design choice is an UI element weighting mechanism applied on the fine-tuning loss to prevent over-abstraction by preserving essential context information embedded in the UI elements, such as the meaning of the clicked button. Extensive quantitative comparisons with ablations showed that both design choices contribute to the effectiveness of SummAct (see Table 1). Additionally, in Section 4.4, the qualitative evaluations examined the generated text and verified that both designs were necessary and complementary to each other: without the sub-goals, the summaries can have logical errors, while without the weighted mechanism, the summaries lacked action details (see Figure 2).

Moreover, as shown in Table 1, our method significantly improves on the results obtained from using pre-trained LLMs (up to three times better, on MoTIF), which is a common practice in HCI research that use LLMs [24, 38, 49]. The substantial enhancement brought by SummAct reveals the considerable potential of further enhancements in these HCI works and positions our approach as a pioneering example in the field. Our findings suggest that the HCI community can gain immensely by further adapting advanced natural language processing techniques to specific HCI applications.

In implementing SummAct, we utilised Mistral-7B as the backbone LLM due to its lightweight and robust performance across various NLP tasks. However, our SummAct framework allows replacing Mistral-7B with other LLM models if computing resources are available or more powerful models appear, such as GPT-4 [42].

6.3 Limitations and Future Work

In our work, we investigated summarising the interactive behaviour at the UI-element level, i.e., the users operate on UI elements such as buttons or combo boxes. In the future, we will dive deeper into the raw, pixel-level interactive behaviours, e.g., integrating the specific on-screen locations of each move, click or tap into our model, which will carry more information about users and their goals [58, 59]. Currently, we integrated UI information via HTML and DOM elements. Future enhancements can adopt vision language models to process GUI screenshots [31, 62]. This will allow for capturing visual cues that HTML alone cannot provide, such as iconography, layout spatial arrangements, and thematic designs that influence user interactions. The proposed UI element weighting mechanism matched tokens by considering whether they were identical. Moving forward, we could consider matching their semantics to increase the model's flexibility. This work considered three levels (action, sub-goal and overall goal) and showed SummAct's potential of handling noisy behaviour via examples (see Appendix C). Future work will explore more complex real-world scenarios where the goals may have more intermediate levels or when users start with vague goals that are even higher-level than the overall goal, which needs to be translated into specific goals. Potential solutions may be to introduce theory of mind [17] and cognitive models [7]. Moreover, our exploration of interactive behaviour summarisation is based on the assumption that the observed UI trajectories accurately reflect user goals without error. While essential for developing our method, in real-world interactions, it is possible that actions may not always convey true goals due to errors in user operation, which will be interesting for developing more robust models in the future.

7 Conclusion

In this work, we modelled interactive behaviour from a natural language viewpoint and investigated a novel interactive behaviour summarisation task, namely summarising input actions into natural language descriptions. These descriptions reflect user goals underlying their interactive behaviour. Towards this task, we proposed SummAct – an LLM-based method with two specific designs, hierarchical summarisation and UI element weighting mechanism. We evaluated our method on two datasets, covering both desktop and mobile interactive settings, from both a quantitative and qualitative perspective. Results demonstrated the effectiveness of our method in summarising goals and the complementary contributions of our two designs. We then showcased two example use cases of interactive behaviour summarisation, including behaviour forecasting and automatic identification of behaviour synonyms. The natural language representation of interactive behaviour can boost the explainability of computational behaviour modelling and contribute to developing more intuitive and responsive interactive systems. Furthermore, the significant improvement over the common practice of directly using LLMs in HCI suggests large potential benefits from further adapting advanced NLP techniques to HCI tasks.

Acknowledgments

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting G. Zhang. Z. Hu was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech).

References

[1] Abdulaziz Almehmadi. 2021. Micro-Behavioral Accidental Click Detection System for Preventing Slip-Based Human Error. *Sensors* 21, 24 (2021), 8209.
[2] Gregor Bachmann and Vaishnavh Nagarajan. 2024. The pitfalls of next-token prediction. *arXiv preprint arXiv:2403.06963* 1, 1 (2024), 1–1.
[3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings*

of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. ACL, ., 65–72.

[4] Roman Bednarik, Hana Vrzakova, and Michal Hradis. 2012. What do you want to do next: a novel approach for intent prediction in gaze-based interaction. In *Proceedings of the symposium on eye tracking research and applications*. ACM, ., 83–90.

[5] Omri Berkovitch, Sapir Caduri, Noam Kahlon, Anatoly Efros, Avi Caciularu, and Ido Dagan. 2024. Identifying User Goals from UI Trajectories. *arXiv preprint arXiv:2406.14314* 1, 1 (2024), 1–1.

[6] Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A. Plummer. 2022. A Dataset for Interactive Vision Language Navigation with Unknown Command Feasibility. In *European Conference on Computer Vision (ECCV)*. Springer, ., 1–1.

[7] Stuart K Card. 2018. *The psychology of human-computer interaction*. Crc Press, .

[8] Ling-Hao Chen, Shunlin Lu, Ailing Zeng, Hao Zhang, Benyou Wang, Ruimao Zhang, and Lei Zhang. 2024. MotionLLM: Understanding Human Behaviors from Human Motions and Videos. *arxiv:2405.20340* 1, 1 (2024), 1–1.

[9] Xianyu Chen, Ming Jiang, and Qi Zhao. 2024. GazeXplain: Learning to Predict Natural Language Explanations of Visual Scanpaths. *arXiv preprint arXiv:2408.02788* 1, 1 (2024), 1–1.

[10] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935* 1, 1 (2024), 1–1.

[11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53.

[12] Brendan David-John, Candace Peacock, Ting Zhang, T Scott Murdison, Hrvoje Benko, and Tanya R Jonker. 2021. Towards gaze-based prediction of the intent to interact in virtual reality. In *ACM symposium on eye tracking research and applications*. ACM, ., 1–7.

[13] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2Web: Towards a Generalist Agent for the Web. arXiv:2306.06070 [cs.CL]

[14] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert systems with applications* 165 (2021), 113679.

[15] Anis Elbahi, Mohamed Ali Mahjoub, and Mohamed Nazih Omri. 2013. Hidden markov model for inferring user task using mouse movement. In *Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)*. IEEE, ., 1–7.

[16] Anis Elbahi and Mohamed Nazih Omri. 2015. Web user interact task recognition based on conditional random fields. In *Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015 Proceedings, Part I 16*. Springer, Valletta, 740–751.

[17] Chris Frith and Uta Frith. 2005. Theory of mind. *Current biology* 15, 17 (2005), R644–R645.

[18] Erik Frøkjær, Morten Hertzum, and Kasper Hornbæk. 2000. Measuring usability: are effectiveness, efficiency, and satisfaction really correlated?. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, ., 345–352.

[19] Eugene Yujun Fu, Tiffany CK Kwok, Erin You Wu, Hong Va Leong, Grace Ngai, and Stephen CF Chan. 2017. Your mouse reveals your next activity: towards predicting user intention from mouse interaction. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, ., 869–874.

[20] Yu Gu, Xiang Deng, and Yu Su. 2022. Don't Generate, Discriminate: A Proposal for Grounding Language Models to Real-World Environments. *arXiv preprint arXiv:2212.09736* 1, 1 (2022), 1–1.

[21] Sakib Haque, Zachary Eberhart, Aakash Bansal, and Collin McMillan. 2022. Semantic similarity metrics for evaluating source code summarization. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*. IEEE, ., 36–47.

[22] Kasper Hornbæk. 2006. Current practice in measuring usability: Challenges to usability studies and research. *International journal of human-computer studies* 64, 2 (2006), 79–102.

[23] Zhiming Hu, Andreas Bulling, Sheng Li, and Guoping Wang. 2022. EHTask: recognizing user tasks from eye and head movements in immersive virtual reality. *IEEE Transactions on Visualization and Computer Graphics* 29, 4 (2022), 1992–2004.

[24] Forrest Huang, Gang Li, Tao Li, and Yang Li. 2024. Automatic Macro Mining from Interaction Traces at Scale. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, ., 1–16.

[25] Peter C Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Adam Santoro, and Timothy Lillicrap. 2022. A data-driven approach for learning to control computers. In *International Conference on Machine Learning*. PMLR, ACM, ., 9466–9482.

[26] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* 1, 1 (2023), 1–1.

[27] Kazuki Kawamura and Jun Rekimoto. 2024. FastPerson: Enhancing Video-Based Learning through Video Summarization that Preserves Linguistic and Visual Contexts. In *Proceedings of the Augmented Humans International Conference 2024*. ACM, ., 205–216.

[28] Saskia Koldijk, Mark Van Staalduinen, Mark Neerincx, and Wessel Kraaij. 2012. Real-time task recognition based on knowledge workers' computer activities. In *Proceedings of the 30th European Conference on Cognitive Ergonomics*. ACM, Edinburgh, 152–159.

[29] Tiffany CK Kwok, Eugene Yujun Fu, Erin You Wu, Michael Xuelin Huang, Grace Ngai, and Hong-Va Leong. 2018. Every little movement has a meaning of its own: Using past mouse movements to predict the next interaction. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*. ACM, Berlin, 397–401.

[30] Haoran Li and Wei Lu. 2021. Mixed cross entropy loss for neural machine translation. In *International Conference on Machine Learning*. PMLR, ., 6425–6436.

[31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, ., 19730–19742.

[32] Jiahao Nick Li, Yan Xu, Tovi Grossman, Stephanie Santosa, and Michelle Li. 2024. OmniActions: Predicting Digital Actions in Response to Real-World Multimodal Sensory Inputs with LLMs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, ., 1–22.

[33] Rui Li, Guoyin Wang, and Jiwei Li. 2024. Are Human-generated Demonstrations Necessary for In-context Learning?. In *The Twelfth International Conference on Learning Representations*. ICLR, ., 1–1.

[34] Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020. Mapping natural language instructions to mobile UI action sequences. *arXiv preprint arXiv:2005.03776* 1, 1 (2020), 8198–8210.

[35] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. ., ., 74–81.

[36] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. 2024. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems* 36 (2024), 1–1.

[37] Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *The 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, ., 1–1.

[38] Zhe Liu, Chunyang Chen, Junjie Wang, Mengzhuo Chen, Boyu Wu, Yuekai Huang, Jun Hu, and Qing Wang. 2024. Unblind Text Inputs: Predicting Hint-text of Text Input in Mobile Apps via LLM. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, ., 1–20.

[39] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* 1, 1 (2016), 1–1.

[40] Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (LLM) hallucination. In *European Semantic Web Conference*. Springer, ., 182–185.

[41] Runliang Niu, Jindong Li, Shiqi Wang, Yali Fu, Xiyu Hu, Xueyuan Leng, He Kong, Yi Chang, and Qi Wang. 2024. Screenagent: A vision language model-driven computer control agent. In *The 33rd International Joint Conference on Artificial Intelligence*. ACM, ., 1–1.

[42] OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[43] Kishore Papineni. 2001. BLEU: a method for automatic evaluation of MT. *Research Report, Computer Science RC22176 (W0109-022)* 1, 1 (2001), 1–1.

[44] Shiva Kumar Pentyala, Zhichao Wang, Bin Bi, Kiran Ramnath, Xiang-Bo Mao, Regunathan Radhakrishnan, Sitaram Asur, et al. 2024. PAFT: A Parallel Training Paradigm for Effective LLM Fine-Tuning. *arXiv preprint arXiv:2406.17923* 1, 1 (2024), 1–1.

[45] Derek J. Phillips, Tim A. Wheeler, and Mykel J. Kochenderfer. 2017. Generalizable intention prediction of human drivers at intersections. In *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, ., 1665–1670. https://doi.org/10.1109/IVS.2017.7995948

[46] Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Yankai Lin, Zhong Zhang, Zhiyuan Liu, and Maosong Sun. 2024. Tell Me More! Towards Implicit User Intention Understanding of Language Model Driven Agents. *arXiv preprint arXiv:2402.09205* 1, 1 (2024), 1–1.

[47] Dana Rezazadegan, Shlomo Berkovsky, Juan C Quiroz, A Baki Kocaballi, Ying Wang, Liliana Laranjo, and Enrico Coiera. 2020. Automatic speech summarisation: A scoping review. *arXiv preprint arXiv:2008.11897* 1, 1 (2020), 1–1.

[48] Noam Rotstein, David Bensaïd, Shaked Brody, Roy Ganz, and Ron Kimmel. 2024. Fusecap: Leveraging large language models for enriched fused image captions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE, ., 5689–5700.

[49] Bryan Wang, Gang Li, and Yang Li. 2023. Enabling conversational interaction with mobile ui using large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, ., 1–17.

[50] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. 2021. Screen2words: Automatic mobile UI summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, ., 498–510.
[51] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. 2024. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems* 36 (2024), 1–1.
[52] Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* 1, 1 (2019), 1–1.
[53] Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2023. Empowering llm to use smartphone for intelligent task automation. *arXiv preprint arXiv:2308.15272* 1, 1 (2023), 1–1.
[54] Xuhai Xu, Chun Yu, Yuntao Wang, and Yuanchun Shi. 2020. Recognizing unintentional touch on interactive tabletop. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–24.
[55] Lin-Ping Yuan, Boyu Li, Jindong Wang, Huamin Qu, and Wei Zeng. 2024. Generating Virtual Reality Stroke Gesture Data from Out-of-Distribution Desktop Stroke Gesture Data. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, ., 732–742.
[56] Guanhua Zhang, Matteo Bortoletto, Zhiming Hu, Lei Shi, Mihai Bâce, and Andreas Bulling. 2023. Exploring Natural Language Processing Methods for Interactive Behaviour Modelling. In *The Proceeding of 2023 IFIP TC13 Conference on Human-Computer Interaction (INTERACT)*. IFIP, ., 1–18.
[57] Guanhua Zhang, Susanne Hindennach, Jan Leusmann, Felix Bühler, Benedict Steuerlein, Sven Mayer, Mihai Bâce, and Andreas Bulling. 2022. Predicting Next Actions and Latent Intents during Text Formatting. In *Proceedings of the CHI Workshop Computational Approaches for Understanding, Generating, and Adapting User Interfaces (2022-01-01)*. ACM, ., 1–6.
[58] Guanhua Zhang, Zhiming Hu, Mihai Bâce, and Andreas Bulling. 2024. Mouse2Vec: Learning Reusable Semantic Representations of Mouse Behaviour. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, ., 1–17.
[59] Guanhua Zhang, Zhiming Hu, and Andreas Bulling. 2024. DisMouse: Disentangling Information from Mouse Movement Data. In *Proc. ACM Symposium on User Interface Software and Technology (UIST)*. ACM, ., 1–13.
[60] Haopeng Zhang, Philip S Yu, and Jiawei Zhang. 2024. A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models. *arXiv preprint arXiv:2406.11289* 1, 1 (2024), 1–1.
[61] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614* 1, 1 (2024), 1–1.
[62] Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. 2024. Synapse: Trajectory-as-Exemplar Prompting with Memory for Computer Control. In *The Twelfth International Conference on Learning Representations*. ICLR, ., 1–1.

A Describing Mind2Web Actions in Natural Language

As mentioned in Section 4.1.1, we preprocessed the raw action strings provided by Mind2Web into natural language descriptions to better leverage the understanding and reasoning capabilities inherent in LLMs and to standardise the input format for SummAct. Table 3 presents three examples of the original actions, each from an user operation category (click, select or type).

Following [41], we first split the original action strings to the UI element category, content (the inherent meaning of this element, e.g., its name or the text on it), additional content for type and select operations (specific content the user is interested in, e.g., selected value from a combo box) and the user operation category; and then inserted these components into the natural language template, structured as:

- If $Operation = CLICK$: [$Operation$] the [$Category$] element with text " [$Content$]" on it
- If $Operation = SELECT$: [$Operation$] "[$Content(Additional)$]" from [$Category$] with text " [$Content$]" on it
- If $Operation = TYPE$: [$Operation$] text "[$Content(Additional)$]" into [$Category$] with text "[$Content$]" on it

Through this templating approach, the example action strings are represented as:

- Click the button element with text "Add to Cart" on it
- Select "Price Low to High" from combobox with text "Sort By" on it
- Type text "Johannesburg" into searchbox with text "Search" on it

Moreover, Table 4 compared the performance of interactive behaviour summarisation by replacing the input actions from the natural language descriptions (*with Preprocessing*) with their original symbolic representations in the prompt (*w/o Preprocessing*). The results of *with Preprocessing* were the same as the last row from Table 1. The drop of performance after removing the conversion demonstrates the effectiveness of this preprocessing.

B Prompts

B.1 In-Context Learning Prompts for Sub-goal Generation

As described in Section 3.1, we apply in-context learning to generate sub-goals from low-level interactive actions. Figure 5 shows an example of the prompt we used for the pretrained LLM. The prompt has three main components: 1) an overall context describing the task the LLM needs to solve, the input format and expected output; 2) examples used in the in-context learning annotated by experts, each including environment metadata, task (only in training samples), interactive behaviour and sub-goals (we only show one example due to its excessive length); 3) and a new sample for which the model needs to generate sub-goals, where the text in green should be replaced and updated for each sample. The texts in blue are related to the ground-truth overall goals, and thus should be removed from testing samples to avoid data leakage. The example in Figure 5 is from Mind2Web dataset, which provides the metadata including website, domain and sub-domain. When using the prompt on MoTIF dataset, simply replace them with the provided mobile application as the new meta data.

B.2 Prompts for Detail Enhanced Fine-tuning

As described in Section 3.2, we fine-tune LLMs to summarise the final, overall goal using both low-level interactive behaviour and mid-level sub-goals. Figure 6 illustrates an example of the prompt we used for fine-tuning. The prompt comprises five parts: 1) an overall context describing the task the LLM needs to solve, the input format and expected output; 2) environment metadata; 3) user input actions; 4) sub-goals; and 5) expected output format. The text in green should be replaced and updated for each sample. The example in Figure 6 is from Mind2Web dataset, which provides the metadata including website, domain and sub-domain. When using the prompt on MoTIF dataset, simply replace them with the provided mobile application as the metadata.

Element Category	Element Content		User Operation	Element Content (Additional)
[*button*]	Add to Cart	→	CLICK	–
[*combobox*]	Sort By	→	SELECT	Price Low to High
[*searchbox*]	Search	→	TYPE	Johannesburg

Table 3: Three example input action strings from Mind2Web dataset. Every string, representing an input action, contains the information of the interacted UI element (category and content) and user's operation on it.

Method	Metric	Cross Domain	Cross Task	Cross Website
w/o Preprocessing	CosSim	.733	.762	.769
	BLEU	.393	.410	.419
	ROUGE	.379	.416	.393
	METEOR	.342	.385	.399
with Preprocessing	CosSim	**.755**	**.796**	**.802**
	BLEU	**.406**	**.453**	**.445**
	ROUGE	**.390**	**.432**	**.447**
	METEOR	**.376**	**.430**	**.454**

Table 4: Interactive behaviour summarisation results achieved by SummAct on Mind2Web without (*w/o*) or *with* preprocessing. Without preprocessing, the actions in the input follow their original symbolic representations, while after preprocessing, these actions are described in natural language. Better results are shown in bold.

B.3 Prompts for Next Action Prediction

As described in Section 5.1, we fine-tune an LLM to predict the next action a user may perform based on the history actions and the goals summarised from them via our SummAct. Figure 7 shows an example of this prompt, consisting of five parts: 1) an overall context describing the task the LLM needs to solve and the allowed action space; 2) input actions the user has performed; 3) DOM of the current webpage; 4) interactive goal summarised from the performed actions; and 5) candidate UI elements as the next target. The text in green should be replaced and updated for each sample. We formulate this task as a multi-choice question-answering problem, where the model needs to output both the target UI element to interact with and the user's corresponding operation. An example output is:

```
[ANSWER] B. Action: CLICK [/ANSWER]
```

which means the predicted next action is clicking on the UI element shown as the option B in the input prompt.

C Summarisation Examples on Noisy Actions

The interaction traces provided by the datasets were clean, while in real-world interactions, the traces may be less structured and include redundant actions. We manually added actions to the original action sequences and then let SummAct summarise the noisy sequences. We observed that SummAct could still generate the correct summaries, i.e., ignoring the redundant actions. For example, when the overall goal was to *Find pet food for cats and sort the results by price, least to most expensive and from Chewy.com*, the original actions included choosing *cat food* in the combobox named "What are you looking for?", clicking on "Price - low to high" to rank the items, and selecting "Chewy.com". Before the first action of choosing cat food, we added a redundant action of choosing dog food to simulate that the user browsed another category but then decided to operate on the results of cat food. Our SummAct generated the same goal as the clean interaction trace as expected. Another example is, towards the goal to *Check permit availability for a group of 4 in Brooks Camp on May 22*, one action was typing *4* in the text box of *Number of People*. We added noisy actions of typing other numbers into this text box before this action, but our SummAct was able to recognise that the goal was related to four people. These example observations indicate the potential of extending SummAct to complex, noisy user behaviours in real-world applications.

You are a helpful computer task assistant. You can break any high-level tasks that can be performed in graphical user interfaces into sub-goals. The user will provide the interface to be used, the domain and subdomain of the interface. Along with these general information, the user will also provide the task they intend to achieve and the actions they perform towards this task.

You have to analyse the given information and create a list of sub-goals. Each sub-goal should be a summary for several actions, which means the number of sub-goals should be more than 1 and less than the number of actions. The sub-goals should together lead to the final task. The following are examples of generating such lists of sub-goals.

Overall context

Example 1
INPUT
Website: exploretock
Domain: Travel
Sub-domain: Restaurant

Environment metadata

Task: Check for pickup restaurant available in Boston, NY on March 18, 5pm with just one guest.

Actions (Each line is one action):
- Select "Pickup" from combobox with text "Reservation type" on it
- Type text "Boston" into searchbox with text "Find a location" on it
- Click the span element with text "Boston" on it
- Click the button element with text "18" on it
- Select "5 00 PM" from combobox with text "Time" on it
- Click the span element with text "2 guests" on it
- Select "1 guest" from combobox with text "Size" on it
- Click the button element with text "Update search" on it

Interactive Behaviour

OUTPUT
Sub-goals (Each line is one sub-goal):
- Set the reservation type to "Pickup".
- Enter "Boston" as the location.
- Select the reservation date and time to March 18, 5pm.
- Adjust the number of guests to 1.
.....

Sub-goals

Example demonstrations

INPUT
Website: exploretock
Domain: Travel
Sub-domain: Car rental
Task: Sign Allan Smith for email offers with the email allan.smith@gmail.com and zip code 10001
Actions (Each line is one action):
- Click the a element with text "Return to a different location" on it',
- Type text "Allan" into textbox with text "First Name (required)" on it',
- Type text "Smith" into textbox with text "Last Name (required)" on it',
- Type text "allan.smith@gmail.com" into textbox with text "Email Address (required)" on it',
- Type text "allan.smith@gmail.com" into textbox with text "Confirm Email Address (required)" on it',
- Type text "10001" into textbox with text "ZIP Code (required)" on it',
- Click the button element with text "Submit" on it'

OUTPUT

New sample for which the model needs to generate sub-goals

Figure 5: Prompt used to generate sub-goals using in-context learning.

Your task is to understand and summarize a user's intention behind their actions on a user interface. You have a list of information, including the website, domain and sub-domain of the user interface, history actions the user performed, and sub-goals (several low-level summarizations of subsets of history actions). Combine all the information and summarize the intention.

Overall context

Website:
dmv.virginia.gov

Domain:
Service

Sub-domain:
Government

Environment metadata

Actions (Each line is one action):
- Click the link element with text "Locations" on it
- Click the link element with text "DMV'S MOBILE OFFICES" on it
- Click the link element with text "View Calendar by Location" on it
- Click the button element with text "Location" on it
- Click the link element with text "HIGHLAND" on it

Interactive Behaviour

Sub-goals summarized from these actions:
- Go to DMV Locations page.
- Open Mobile Offices calendar view.
- Filter the locations based on Highland area.

Sub-goals

Expected output format

Instructions:
Summarize:
In clear and concise language, summarize the comprehensive goal the user wants to achieve via the history actions. Present the summary after the heading [SUMMARY].

Figure 6: Prompt used to summarise the overall goal from low-level interactive behaviour and mid-level sub-goals.

You are an agent in a web-based environment. Your task is to predict the user's next action within the given action space, based on the user's previous actions, the DOM of the current webpage, and intention so far.
Action space:
1. `CLICK': Click on an element.
2. `TYPE [value]': Type a value into an element.
3. `SELECT [value]': Select a value for an element.

Overall context

Previous actions (Each line is one action):
- Click the button element with text Hotel on it
- Click the button element with text Restaurant on it
- Select Pickup from combobox with text Reservation type on it
- Type text New into searchbox with text Find a location on it
- Click the span element with text New York on it

Previous actions

DOM:
(html (div (div (div (p DELICIOUS) (p id=0 STARTS) (p HERE.)) (div (label id=1 Reservation type) (div (label id=2 Date) (div (input text date thu, mar 16) (div id=3 (button button date, selected value is thu, (svg)))))))) (a id=4 (div img image of hilton alumni association) (section (h3 Hilton Alumni Association) (p Hilton, NY Non-culinary))))) ...

DOM of current webpage

User intention so far: Check for pickup restaurant available in New York.

Summary of previous actions

What should be the next action? Please select from the following choices by selecting the letter that identifies the correct choice. (If the correct action is not in the page above, please select A. 'None of the above'):
A. None of the above
B. (p id=0 STARTS)
C. (label id=1 Reservation type)
D. (label id=2 Date)
E. (div id=3 (button button date, selected value is thu, (svg)))
F. (a id=4 (div img image of hilton alumni association)) ...

Write your answer as: [ANSWER] your final answer [/ANSWER]

Multi-choice question

Figure 7: Prompt used to predict the next action based on previous input actions and the current web page.