Small Models, Big Results: Achieving Superior Intent Extraction with through Decomposition

Anonymous ACL submission

Abstract

Understanding user intents from UI interaction trajectories remains a challenging, yet crucial, frontier in intelligent agent development. While massive, datacenter-based, multi-modal large language models (MLLMs) possess greater capacity to handle the complexities of such sequences, smaller models which can run on-device to provide a privacy-preserving, low-cost, and low-latency user experience, struggle with accurate intent inference. In this paper, we address these limitations by introducing a novel decomposed approach: first, we perform structured interaction summarization, capturing key information from each user action. Second, we perform fine-tuned intent extraction model operating on the aggregated summaries. Remarkably, this method empowers resource-constrained models to not only achieve improved intent understanding, but also surpass the base performance of large MLLMs.

1 Introduction

Accelerated advancements in the capabilities of multi-modal large language models (MLLMs) has led to recent interest in modeling sequences of user interactions with phone and web graphical interfaces, both for the purposes of automation (Wang et al., 2024b; Jiménez-Ramírez, 2024), and understanding (Berkovitch et al., 2024; Zhang et al., 2024).

In this work, we focus on the user intent extraction task, which consists of producing a free-form description of the inferred intent of a user from a sequence of interactions with a device e.g., screenshots and actions.

Large MLLMs are naturally fairly good at this task, however, it is more challenging for smaller models (E.g., Gemini 1.5 Flash 8B (Gemini Team et al., 2024) or Qwen2 VL 7B (Wang et al., 2024a)). The performance of smaller models is important for user interaction tasks due to their ability to operate within private, on-device environment like a phone or browser, with reduced cost, energy usage, and latency (Xu et al., 2024).

In this paper, we introduce a two-stage approach for extracting user intent with small models. In the first stage, each atomic interaction is summarized. In the second stage, the full sequence of summaries is fed to a second model which outputs an intent. The overall flow is illustrated in Figure 1. Using semantic equivalence metrics on public UI automation data, our two-stage approach demonstrates superior performance compared to both smaller models and a state-of-the-art large MLLM, independent of dataset and model type. Our approach also naturally handles scenarios with noisy data that traditional supervised fine-tuning methods struggle with. The modular nature of the architecture is extremely helpful from an engineering perspective, allowing us to evaluate the approach in detail and identify key areas to improve.

Our contributions can be summarized as follows: 1) We describe an effective decomposition the intent-extraction that unlocks the potential of small models; 2) We present non-trivial design components related to each stage of the decomposition; 3) We extensively evaluate our approach and demonstrate the effectiveness of our method across a range of data sets, base models and metrics.

2 Background

2.1 Intent Extraction from UI Interactions

We formalize the intent extraction task, sometimes called goal understanding, similarly to Berkovitch et al. (2024) and Zhang et al. (2024). Consider a user journey T within a mobile or web application, represented as a sequence of interactions: $T = (I_1, I_2, ..., I_n)$, where each interaction $I_i = (O_i, A_i)$ consists of an observation O_i, and the action A_i the user performed at that step. This

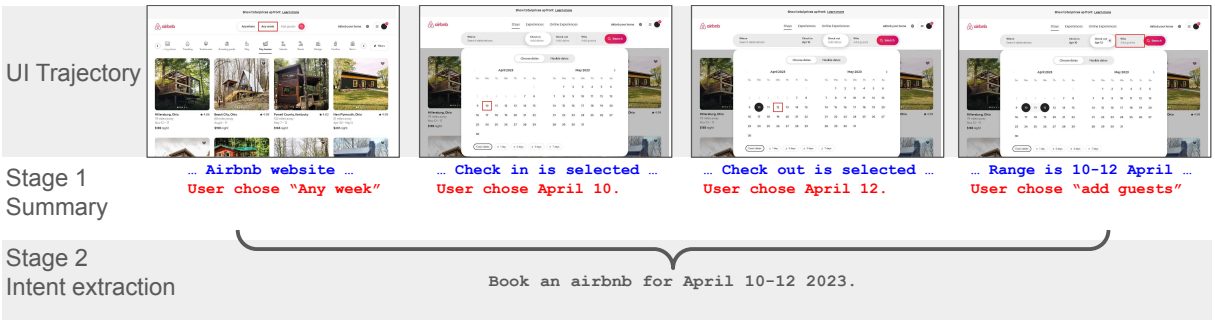

Figure 1: Proposed intent extraction flow (Described in detail in Section 4). Individual interactions are summarized independently and then sequence of summaries are combined to output a short inferred intent for the trajectory. The summaries are structured in two fields corresponding to screen summary (top, blue) and user action (bottom, red).

description is general and different modeling approaches have used different representations for observations and actions (e.g., textual descriptions, screenshot images, DOM hierarchies, etc.) (e.g., Rawles et al., 2023; Burns et al., 2022). The objective of the intent-extraction task is to generate a free-form sentence describing the user's intent. Effectively, this setting can be thought of as the inverse problem of the UI automation task, with inputs and outputs swapped. Rather than producing a sequence of actions from an instruction, we ask "what was the user trying to accomplish with this trajectory?". Intent extraction have been identified as an important building block for UI automation tasks, proactive assistance, and personalized memory (Berkovitch et al., 2024; Zhang et al., 2024).

A good intent is: *faithful* - i.e., only describes things that actually occur in the trajectory, *comprehensive* - i.e., provides all of the information about the user intent required to re-enact the trajectory, and *relevant* - i.e., does not contain extraneous information beyond what is needed for comprehensiveness. However, even with a well-defined ground truth intent, accurately evaluating a model's extracted intent is challenging. User intents often contain many details, such as trip planning specifics or transaction data, necessitating metrics that can handle partial matches. Such metrics fall into two categories: semantic, which analyze underlying meaning, and lexical, which assess surface-level word overlap. As shown in previous work (Caduri, 2025), lexical metrics (e.g., BLEU and ROUGE) correlate poorly with human judgments of intent similarity, as they merely compare words. In contrast, semantic metrics, such as NLI (Natural Language Inference) and BI-Fact (a bi-directional variant of FActScore (Min et al., 2023)), strive to capture the intended meaning.

Further, there is inherent subjectivity in intent extraction, as a single trajectory could have been driven by multiple underlying motivations (e.g., a user may have selected a flight based on its price versus its departure time). This subjectivity is evident in prior work, such as Berkovitch et al. (2024) where human-composed intentions matched each other in only 80% and 76% of web and phone trajectories, respectively. This level of human agreement may be considered a practical upper bound for performance on this task.

2.2 UI Interaction Datasets

Recently, a number of datasets have been developed for evaluating UI interaction agents, (surveyed in Wang et al. (2024b)). We use two that are representative and suitable for measuring the intent extraction task. We confirmed that our usage of the data adhered to all ethical and legal standards.

Mind2Web (CC BY 4.0 license) (Deng et al., 2024): Has 2,350 human demonstrations on websites. Each user trajectory is on average 7.3 steps long and contains screenshots and actions for each step, as well as high level description of the task the human was asked to perform.

AndroidControl (Apache 2.0 license) (Li et al., 2024): Has 15,283 examples of humans performing tasks on Android apps. Each user trajectory is on average 5.5 steps long, and contains screenshots and actions for each step, as well as high level description of the goal.

Mind2Web's data collection included a validation step where annotators verified the alignment between the completed trajectory steps and the intent making this dataset highly suitable for the intent extraction task as well. This crucial step was absent from the AndroidControl collection protocol, resulting in noisier labels. For example con-

sider the following task "Delete all emails from sender X" while there were no emails from that sender. Based on the execution of task it's impossible to identify that the original goal was to delete the emails. We preprocess labels to remove clearly irrelevant statements (Section 5.2) and analyze the effect of remaining discrepancies between the labels and trajectories in Section 6.2.

2.3 Related Work

User interaction understanding for HCI Single screen summarization as a special case of image description has been extensively studied for the purposes of e.g., accessibility, automation, and question answering (e.g., Li et al., 2021b; Bai et al., 2021; Li and Li, 2022; Wang et al., 2021; Yang et al., 2024).

Our setting of identifying and summarizing intents from trajectories has been recently proposed in Berkovitch et al. (2024); Zhang et al. (2024); Jiménez-Ramírez (2024).

Multi-stage summarizations Decomposing a complex task into smaller simpler stages is a well-known approach for problem solving. Hierarchical models are common in summarization tasks of many modalities, e.g., text (Christensen et al., 2014), audio (Li et al., 2021a), video (Zhao et al., 2022; Cheng et al., 2024).

Chain of thought reasoning (Wei et al., 2022) is a very popular general purpose method for prompting models to decompose a problem into smaller parts. Khot et al. (2022) propose an automated decomposition step that delegates different parts of the problem to distinct model calls.

3 Baseline Modeling Approaches

In this section, we first present natural baseline approaches for addressing our task, whose lessons have led us to developing our decomposed two-stage approach which will be described in Section 4. Our task is a text generation task, where intent descriptions are generated from the multi-modal input of UI trajectories. As such, it is most natural to address it through multi-modal LMs, applying either prompt-based or fine-tuned methods, as described below. The focus of our work is to explore the use of small LMs, aiming at their eventual utilization on-device. The particular models we experimented with are specified in Section 5.1, including a top-tier large model as a reference point.

Prompt-based methods Such methods are advantageous in that they do not require training data, instructing a *generic* LM via its prompt. We found that a *Chain-of-Thought* (CoT) prompt worked best. Specifically, our CoT prompt (see C1) instructs the model to first generate a sequence of individual descriptions of the user intents within *each UI interaction*, and then to consolidate these interaction-level description into the final description of the accumulated user intent along the trajectory.

Fine-tuned models Since performance of prompting a generic model may not be fully aligned with the intended task output, while prompt-based performance of small LMs might generally be limited, we explore also baseline fine-tuning methods. To that end, we fine-tuned small models using available training datasets, specifically those developed for the inverse problem of UI automation, while swapping their input/output roles (see Section 2.2).

All of the baselines require composing a single prompt that contains the entire user trajectory including images, requiring a large context window. As described in Section 5.1, practically this required some filtering over the input to fit the available context size.

4 A Decomposed Two-stage Model

4.1 Overview

While CoT prompting works well with large language models (LLMs), we observe limitations in both CoT and fine-tuned small LMs when presented with the full trajectory at once. When applying CoT reasoning, small models struggle to generate a high-quality thoughts that cover the full trajectory. Fine-tuned small models also have trouble generating comprehensive intents from the full trajectory.

These observations led us to develop a decomposed, two-stage approach that emulates the CoT process, illustrated in Figure 1. First, we use prompting to generate a summary for each interaction in a trajectory. This stage is prompt-based as there is currently no training data available with summary labels for individual interactions. Second, we feed all of the interaction-level summaries into a second stage model to generate an overall intent description. We can apply fine-tuning in the second stage and we describe that process in more detail below (Section 4.3). The following subsections

provide a detailed description of each stage in our proposed method.

4.2 Interaction Summarization

In the first stage, we apply a screen summarizer to each individual user interaction $I_i = \{O_i, A_i\}$ of the length-n trajectory $T = (I_1, \ldots, I_n)$. The summarizer extracts relevant information regarding the user's goals and actions within that each interaction. The output of this stage is a summary of the screen context and user action (see Figure 1). This key information, which describes this particular user interaction, will be used in the subsequent fusion stage. This summarization process is entirely prompt-based (our prompt is given in Appendix C).

We add two improvements to the design of this stage that improve overall performance: Adding a context window and structured summaries. The effects of both of these improvements are measured in ablation studies in Section 6.3.

Context window While the primary task is to understand I_i in isolation, we recognize that often context can be crucial for eliminating ambiguity and/or uncertainty. Therefore, in addition to I_i the model also receives as input the preceding and successive interactions, I_{i-1} and I_{i+1}, respectively.

Structured summaries We request that the summary be structured in two distinct components: (a) the relevant screen context – a short list of salient details on the current screen O_i, and (b) the user action: a list of mid-level actions that the user took in the current interaction (example in Figure 1).

As a practical measure for dealing with cases where the model outputs its (unwarranted) interpretations of the user's underlying intent, we also instructed it to output those in a third field (labelled "*speculative intent*") that we discard before proceeding to the next stage.

This structured format was selected to address challenges encountered with alternative prompting strategies. Simply asking the model to be concise resulted in summaries that lacked crucial details. Conversely, prompting for comprehensive information, including user intent, led to excessive speculation that could hinder the subsequent summary fusion stage. Our structured format aims to capture a broader range of information while enabling the removal of speculative elements prior to the second stage. This balanced approach mitigates the risk of contradictions and improves the overall summarization process.

4.3 Generating Session-Level Intent

In the second stage, we aggregate the information extracted during the first stage. A second-stage model takes as input the summaries of all interactions in the trajectory to infer the user's overall intent.

This aggregation stage can be implemented by using either pure prompting of a base model, or by additionally fine-tuning a model that specializes in the aforementioned aggregation. For fine-tuning, the training data consists of: a) input summaries representing all interactions in the trajectory, and b) a corresponding ground truth target that describes the user's overall intent in the given trajectory.

We noted in early explorations that naively applying fine-tuning yields a model that embellishes or hallucinates by introducing details that were not present in the screen summary inputs. On further examination, we found that the training procedure encourages the model to act this way since the inputs are potentially incomplete summaries and the targets are the complete intent statements. Thus, when looking at (input, target) pairs, the model learns that it needs to sometimes add additional information in order to produce the target intent.

Following this insight we refine our target intents at training time to remove details not reflected in the corresponding input (using a large language model, see Appendix C for details on the prompt used in this stage). This ensures that the model will learn to infer intents based solely on the provided interaction summaries. We discuss the effects of this cleanup stage in Subsection 6.3.

5 Experimental Setting

5.1 Models

We focus on smaller, multi-modal (image and text input) models, that can be fine-tuned. In particular, we use *Gemini*[1] *1.5 Flash 8B* (Gemini Team et al., 2024) and *Qwen2 VL 7B* (Apache 2.0 license) (Wang et al., 2024a). For comparisons with a MLLM, we use *Gemini 1.5 Pro* (Gemini Team et al., 2024).

When using the Qwen2 VL 7B for baseline models, we dropped frames randomly from the trajectory if they exceeded the context window length. We found that limiting trajectories to 15 steps was sufficient to run our experiments. We also downsized AndroidControl images by a factor of 4 in

[1]Terms of service: https://ai.google.dev/gemini-api/terms

each dimension when inputting them to Qwen models. Details of fine-tuning can be found in Appendix A.

5.2 Datasets and Preprocessing

We use the Mind2Web (Deng et al., 2024) and AndroidControl (Li et al., 2024) datasets as representative user interaction datasets. We follow the standard train/test split of each dataset, fine-tuning with train, and reporting results on test data.

In Mind2Web, we represent the action from the dataset textually: (e.g., "[element name] click" or "[element name] hover".). In AndroidControl, we use the accessibility tree to convert the screen coordinates of the interacted item to an element name and format the action in the same way. In both datasets, we use screenshots as observations. We highlight the interacted element in the screenshot with with a red box (Zheng et al.; Yang et al., 2023).

For the gold standard extracted goal, we use the high-level goal for each dataset. As mentioned in Section 2.2, the annotation process of AndroidControl was less rigorous than that of Mind2Web, resulting in noisier labels. Furthermore, AndroidControl labels, designed to simulate real user instructions, often contain irrelevant information that cannot be inferred from the trajectory (e.g., "I'm hungry, order an olive pizza from DoorDash"). To mitigate the impact of this noise, we cleaned the labels using Gemini 1.5 Pro (prompt in Appendix C). This cleaning still doesn't completely provide clean goals like Mind2Web's validation process. We find that even after applying a prompt-based cleaning, manual validation on 100 examples (following the annotation protocol in Berkovitch et al. (2024)) makes changes to $\sim 30\%$ of the label intents. We use this manually cleaned data in Section 6.2 to quantify the impact of AndroidControl's noisy labels.

5.3 Evaluation Metrics

We measure quality of extracted goals using two different semantic equivalence metrics.

T5 NLI (Honovich et al., 2022): A T5-XXL model[2] trained for NLI (Natural Language Inference). We compute the entailment probability of the produced summary from the gold standard and vice versa, and then average the two values to get a single bidirectional score.

[2]Available at: https://huggingface.co/google/t5_xxl_true_nli_mixture

BiFact (Caduri, 2025): A bi-directional variation of FActScore (Min et al., 2023) developed for assessing the equivalence of intents in UI interactions, demonstrating the highest correlation with human judgments compared to existing methods. This metric deconstructs both the ground-truth and predicted intents into their fundamental factual components using an LLM (we use Gemini 1.5 Pro for this). These components are then compared to measure the extent of coverage. We use the BiFact measures of precision (the proportion of facts in the predicted intent that are present in the true intent - i.e., relevance), recall (the proportion of facts in the true intent that are captured by the predicted intent, i.e., comprehensiveness) and F1.

We believe that the BiFact, which uses a fine-grained, fact-level comparison is ideally suited for our task since intents can be composed of many parts (e.g., book a flight, flight is to LAX, flight is on Friday). NLI, which holistically evaluates logical entailment of the full sentences is less ideal, but provides an extra signal.

6 Experiments

6.1 Evaluating Extracted Intents

To show that our decomposed approach is generally helpful compared to baselines across models and data modalities, we evaluate the metrics in Section 5.3 on two different datasets using two different models. The results are displayed in Table 1.

In this table, CoT (Chain of Thought) and E2E-FT (End-to-End Fine-tuned) represent the natural baselines described in Section 3.

Of these two baselines, neither is uniformly more effective across all settings. On the Mind2Web dataset, which has cleaner labels (described in 2.2), Gemini, as a stronger base model, has higher BiFact F1 and Bi-NLI scores with CoT, whereas Qwen2 VL 7B benefits from fine-tuning.

Without fine-tuning, the decomposed model has strong recall, but lower precision, giving it mixed results compared to the baselines. However, by adding a fine-tuned intent extraction step, the decomposed FT model outperforms all baselines on the BiFact metric and nearly all baselines on the Bi-NLI metric.

Gemini 1.5 Pro COT is presented as a comparison to a top-tier large MLLM. We find that on Mind2Web, the fine-tuned decomposed approach allows the Gemini Flash 8B to even exceed the performance of the Gemini 1.5 Pro model using COT.

	Mind2Web				AndroidControl			
	BiFact			Bi-NLI	BiFact			Bi-NLI
Method	Recall	Precision	F1		Recall	Precision	F1	
Gemini Flash 8B								
CoT	0.656	0.751	0.660	0.326	0.660	0.628	0.594	0.302
E2E-FT	0.676	0.676	0.652	0.311	0.656	0.655	0.611	0.343
Decomposed	**0.792**	0.717	0.718	0.221	**0.719**	0.488	0.528	0.185
Decomposed-FT	0.756	**0.807**	**0.753**	**0.391**	0.688	**0.664**	**0.630**	**0.35**
Qwen2 VL 7B								
CoT	0.551	0.694	0.563	0.272	0.603	0.589	0.538	0.28
E2E-FT	0.621	0.670	0.610	0.233	0.546	0.594	0.506	**0.343**
Decomposed	**0.672**	0.621	0.591	0.154	0.630	0.385	0.420	0.181
Decomposed-FT	0.609	**0.736**	**0.623**	**0.3**	**0.646**	**0.661**	**0.608**	0.333
Gemini-1.5-Pro								
CoT	0.740	0.761	0.721	0.331	0.767	0.612	0.634	0.347

Table 1: BiFact and Bi-NLI results on the Mind2Web (N=1,005) and AndroidControl (N=1,543) datasets using Gemini 1.5 Flash 8B, Qwen2 VL 7B, and Gemini 1.5 Pro. Best scoring method for each model is bolded. F1, precision, recall are micro-averaged over the dataset.

On AndroidControl, the scores are comparable.

The BiFact score is non-deterministic as it uses an LLM to compute the score. We observe a 0.016 standard deviation when running it multiple times.

Manual verification - Human preference: As a small additional verification, we presented a human rater with 20 trajectories from the Mind2Web with intent predictions from Gemini Flash 8B and asked them to choose whether they preferred the response using CoT or Decomposed-FT (details in Appendix B). Overall, Decomposed-FT was preferred 12 times, CoT 4 times, and 4 judged to be equally good.

6.2 Label Quality and Comparison with Expert-Written Intents

As mentioned in 5.2, we elicited expert-written intent statements for 100 examples in the AndroidControl dataset. In Table 2 we compare the BiFact F1 metric for proposed intents against dataset labels and against expert written intents.

Overall, the performance of each model improves when compared to expert annotations, except for the E2E-FT model, which was trained on the noisy labels. The fine-tuned decomposed approach also uses fine-tuning, and could have been expected to similarly suffer from training on noisy labels, but instead it significantly improves when evaluated using expert intents. We believe this is due to our approach to constructing fine-tuning labels (Section 4.3) which removes information present in the gold labels but absent from summaries. Interesting to notice that after cleaning the AndroidControl the performance of Gemini Pro CoT on it is similar to the performance on Mind2Web suggesting the gap in performance between the dataset is only the result of the noisier data.

Method (Gemini-1.5 Flash 8B)	Expert Labels	Dataset Labels
CoT	0.652	0.580
E2E-FT	0.590	0.565
Decomposed	0.535	0.512
Decomposed-FT	**0.701**	**0.596**
Gemini-1.5-Pro CoT	0.724	0.635

Table 2: A comparison of BiFact F1 scores for intent prediction on the AndroidControl dataset, using expert annotations and dataset labels as ground truth.

6.3 Ablation Study

We consider three variants of our method to estimate the impact of each design choice.

No Context In this variant, Stage 1 is provided with only a single interaction, without previous or next interactions. Our analysis reveals that incorporating information from the previous and next interactions significantly helps the model to infer the user action in the current screen, thereby leading to a noticeable increase in Stage 1 recall. Removing this contextual information significantly reduces overall recall, as shown in Table 3.

Unstructured interaction-level summaries Our method instructs the model to output interaction summaries that are structurally broken down into context, user actions, and a speculative intent list (which is removed prior to proceeding to the next stage). Instead, we permit free-form summaries, and the concatenation of those are provided to the goal extraction. As the results in Table 3 show, instructing the model to output these particular structured responses allows the Stage 2 model to focus on user actions on the one hand, while mitigating Stage 1 hallucinations as much possible. We notice a slight decrease in both precision and recall, as a result of eliminating this part in our method.

No Label Refinement Recall that label refinement was added to address Stage 2 hallucinations. In this variant, we exclude the label refinement step, during the data preparation for the fine-tuning of the Stage 2 model, as described in Subsection 4.3. As expected, after removing this step, we notice a significant decrease in precision. However, we also see a slight increase in recall, suggesting potential areas for improvement in the refinement process.

Method	F1	Precision	Recall
Decomposed-FT	**0.753**	**0.807**	0.756
- No context	0.710	0.787	0.709
- Unstructured	0.733	0.787	0.741
- No label refine	0.728	0.740	**0.776**

Table 3: Ablation study on Mind2Web using BiFact scores. The Decomposed-FT model is the full model and then each subsequent line shows the effect of removing a single design component.

6.4 Manual Error Analysis

To gain a deeper understanding of the errors produced by the decomposed-FT model, we manually analyzed 20 examples. We categorized the errors based on the two main components of our approach: Interaction Summarization and Generating Session-Level Intent. Additionally, we identified instances where the evaluation method itself contributed to discrepancies.

Counts are indicated in parentheses after each error type. Some examples exhibited multiple error types, so the counts do not necessarily add up to the total number of examples.

Errors in Interaction Summarization:

- **Incorrect screen understanding (6)**: Includes instances where the model misinterpreted the UI elements or incorrectly understood the user action.
- **Omissions (6)**: Includes instances where the model failed to capture important on-screen details, like not mentioning the destination on a flight booking site.
- **Hallucinations (4)**: Includes instances involving generating information not present on the screen, such as claiming the user selected a specific item when they did not.
- **Irrelevant details (0)**: Includes instances where the model included correct but excessive information which often confused the second stage. While this error was not present in our full model, it was significant in the "no formatting" models used in the ablation study (Section 6.3).

Errors in Generating Session-Level Intent:

- **Omissions (8)**: Includes instances where the second stage failed to include important details present in the individual summaries.

Evaluation Errors (1): These errors were infrequent and typically involved situations where complex screen understanding was required to determine the equivalence of intents.

Our analysis highlighted the Interaction Summarization stage as the primary source of errors, suggesting potential benefits from distillation training (fine-tuning the smaller model based on the outputs of a larger one). However, initial experiments with distillation did not yield significant improvements, a finding that warrants further investigation in future work.

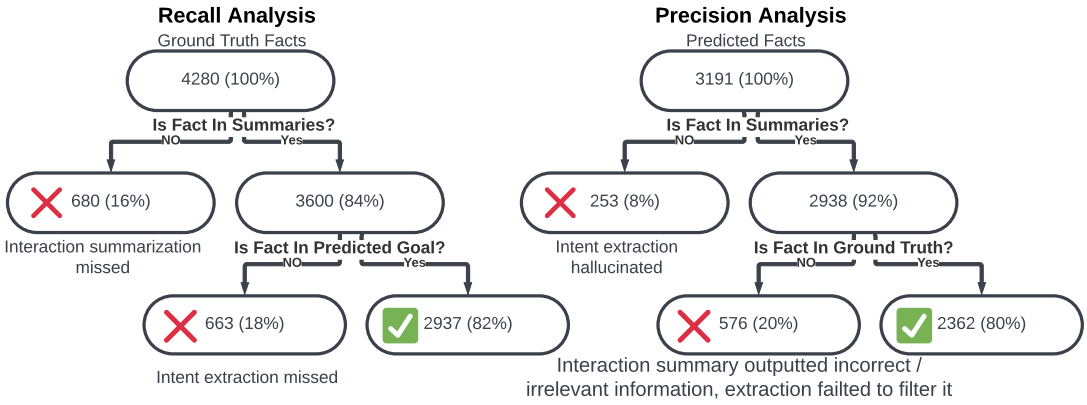

Figure 2: Performance analysis of our method on the Mind2Web dataset, tested with Gemini Flash-8B, tracking ground-truth and predicted facts to obtain stage-level recall and precision.

6.5 Granular Performance Analysis

The modular structure of our decomposed approach, combined with the granularity of the BiFact metric allows us to gain a deeper understanding of our method's performance at each stage and identify areas for future optimization. For that purpose, we adopt the BiFact approach, and break down both the ground-truth intents and the predicted intents into atomic facts, enabling us to track them across both stages with a finer granularity than analyzing full intents. Recall is assessed by tracking ground-truth facts and attributing misses to either the interaction summarization stage, or the intent extraction stage. As for precision, we attribute incorrectly predicted facts to two main issues: the first is hallucinations; i.e., facts that were not present in the summaries. The second cause for incorrect facts prediction poor quality summary content that gets propagated to the output of the second stage without being filtered-out. We lump incorrect and irrelevant information summarization facts together as it is challenging to automatically distinguish between the two. Our analysis of the Mind2Web test set is given in Figure 2 using the Decomposed-FT model. The left-hand side, which focuses on recall, shows that the summarization process results in a 16% loss of ground truth facts. Subsequently, intent extraction further reduces the remaining facts by 18%. Effectively, each stage introduces a similar magnitude of error. The right-hand side describes the precision analysis, showing that 8% of the facts predicted by Decomposed-FT were, in fact, hallucinations. This low hallucination rate is attributed to the label processing techniques employed during training. Following that, 20% of the remaining predicted facts were present in the summary but absent from the ground truth, indicating incorrect or irrelevant information in the interaction summarization output and a filtering issue of the intent extractor. We propose this analysis framework as a means to evaluate future two-stage intent extraction methods by helping to determine the optimal focus of future efforts and by assessing the impact of each stage to overall performance.

7 Discussion

Our study utilized datasets designed for automation to tackle the challenge of user intent identification, despite their inherent limitations such as noise and information gaps. We observe that fine-tuning alone does not surpass Chain-of-Thought, especially in noisy data scenarios. However, our two stage decomposition exhibited superior performance delivering significant improvements regardless of data quality. This improvement can be attributed to the cleaning process and the combination of prompts and fine-tuning, which effectively mitigated the impact of data noise.

Furthermore, our approach significantly reduced the storage footprint of individual screenshots by summarizing each screen independently, thereby minimizing the required tokens for representation. This reduction in token usage is particularly beneficial for on-device models with limited context windows, enabling them to handle longer trajectories more effectively.

8 Ethical Considerations & Risks

Autonomous agents offer significant innovation, but their development necessitates careful ethical consideration, particularly regarding user privacy. Our research, which aims to interpret user intent from UI interactions, inherently involves sensitive data. We particularly study small models that can run on-device, thereby reducing some of the privacy risks associated with transmitting data to external servers. Furthermore, accurately understanding user intents can greatly benefit users through enhanced personalization, improved work efficiency, and facilitating future recall of past activities on their devices. While this work focuses on intent understanding, the development of agents capable of autonomously completing actions requires extreme care. The potential for for misalignment with user intentions and the need for robust safeguards must be thoroughly addressed to ensure responsible deployment.

9 Limitations

We acknowledge several discrepancies between our datasets and real-world user behavior. The datasets predominantly feature English-language, U.S.-centric web interactions, restricting our analysis to this specific demographic. In contrast, real-world users frequently navigate multiple applications, adapt their goals on the fly, and exhibit varying levels of digital literacy, resulting in more complex and unpredictable interaction patterns. The Mind2Web dataset's single-website limitation further deviates from the multi-site nature of typical user tasks. Additionally, our study's reliance on Android and web environments limits the generalizability of our findings to other platforms.

References

Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, et al. 2021. Uibert: Learning generic multimodal representations for ui understanding. *arXiv preprint arXiv:2107.13731*.

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*.

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*, pages 312–328. Springer.

Sapir Caduri. 2025. Bi-fact: A bidirectional factorization-based evaluation of intent extraction from ui trajectories. *arXiv preprint arXiv:2406.14314*.

Dingxin Cheng, Mingda Li, Jingyu Liu, Yongxin Guo, Bin Jiang, Qingbin Liu, Xi Chen, and Bo Zhao. 2024. Enhancing long video understanding via hierarchical event-based memory. *CoRR*.

Janara Christensen, Stephen Soderland, Gagan Bansal, et al. 2014. Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 902–912.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Gemini Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

A Jiménez-Ramírez. 2024. A screenshot-based task mining framework for disclosing the drivers behind variable human actions. *Information Systems*, 121:102340.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Daniel Li, Thomas Chen, Albert Tung, and Lydia B Chilton. 2021a. Hierarchical summarization for longform spoken dialog. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 582–597.

Gang Li and Yang Li. 2022. Spotlight: Mobile ui understanding using vision-language models with a focus. *arXiv preprint arXiv:2209.14927*.

Toby Jia-Jun Li, Lindsay Popowski, Tom Mitchell, and Brad A Myers. 2021b. Screen2vec: Semantic embedding of gui screens and gui components. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024. On the effects of data scale on computer control agents. *arXiv preprint arXiv:2406.03679*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Android in the wild: A large-scale dataset for android device control. *arXiv preprint arXiv:2307.10088*.

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*, pages 498–510.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Shuai Wang, Weiwen Liu, Jingxuan Chen, Weinan Gan, Xingshan Zeng, Shuai Yu, Xinlong Hao, Kun Shao, Yasheng Wang, and Ruiming Tang. 2024b. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. 2024. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2024. Aria-ui: Visual grounding for gui instructions. *arXiv preprint arXiv:2412.16256*.

Guanhua Zhang, Mohamed Ahmed, Zhiming Hu, and Andreas Bulling. 2024. Summact: Uncovering user intentions through interactive behaviour summarisation. *arXiv preprint arXiv:2410.08356*.

Bin Zhao, Maoguo Gong, and Xuelong Li. 2022. Hierarchical multimodal transformer to summarize videos. *Neurocomputing*, 468:360–369.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. In *Forty-first International Conference on Machine Learning*.

A Fine-Tuning Details

We fine-tuned Gemini models similarly to the process described at https://ai.google.dev/gemini-api/docs/model-tuning. We use a batch-size of 32 and fine-tuned for 1,000 steps saving checkpoints every 100 steps and then choose the model that minimizes negative-log-likelihood of the validation data. We fine-tuned the Qwen2-VL-7B model using the Hugging Face TRL library. We followed the steps described in the official Hugging Face VL fine-tuning cookbook[3], and adhered to the hyper-parameters used by the author. As for done for Gemini, we used the checkpoint that minimized the validation NLL loss function. We saved a checkpoint every 20 steps, as recommended in the tutorial, and performed up to three training epochs in total.
For the AndroidControl dataset, we used 5,000 training examples and 137 validation examples randomly sampled from the train set. For Mind2Web we used 900 training examples and 90 validation examples.

B Human Preference Annotation

We presented the rater with a full trajectory of screenshots and actions, and then asked the following question: "After you have seen the trajectory, which intent better describes the trajectory? A or B." The choices A and B contained either CoT or Decomposed-FT. The order of the two options were randomized in each question and the names of the methods were not shown to the respondent. The decoding of choices to model name was only done after the rater had finished the task.

[3] https://huggingface.co/learn/cookbook/en/fine_tuning_vlm_trl

C Prompts

You are analyzing a user's session on a mobile app. Each session consists of a sequence of screenshots and actions that will appear at the end of this prompt. Your goal is to understand the user's overall intent based on the series of interactions provided.

Instructions:

1. Analyze Each Step in the Sequence**:
  - What is displayed in the screenshot?
  - What action did the user take?
  - Why might the user have taken this action?
  - What specific details are relevant? (e.g., dates, items, locations, quantities)

2. Summarize the User's Goal:
  - After analyzing each screenshot-action pair, combine insights to determine the user's overall objective.
  - Include all observed details to make the goal clear and specific.

Output:

1. **Reasoning**:
  - Provide a step-by-step analysis of each screenshot and action pair.
  - Focus on the user's likely intentions and relevant details observed in the input sequence.

2. **Final Answer**:
  - Summarize the user's overall goal in one concise sentence.
  - Start with an action verb. Phrase the action in the imperative form.
  - Include all specific details observed in the input sequence.
  - Avoid using any phrases or structures from the instructions or examples above.
  - Your final answer should start with the appname, followed by a semicolon, and then the inferred goal (example: "eBay; order a basket ball").

Important Notes:

- Do not reuse or paraphrase any examples provided in the instructions.
- Base your response solely on the screenshots and actions in the input sequence.
- Each session is unique ensure your final answer reflects the specific details of this session alone.

The format of the the output should json:

```
{{
   "reasoning": "your reasoning here",
   "final_answer": "your final answer here"
}}
```

The sequence of screenshots and actions for this session will now follow:

Figure C1: CoT model prompt

Your task is to rephrase instructions given by users to automated agents that execute tasks on the user's phone.
Rephrase the instruction in the imperative mood, starting with a verb, and remove any text irrelevant to the user's objective.
Add the app name before the instruction, in the following format: "App name; Instruction".
If the app is now known, use "Unknown app; Instruction".
Correct any spelling or punctuation errors as needed.

Here are a few examples of rephrased instructions:
Input: I am tired of the hustle and bustle of the world. I want to just have a peaceful mind. Play the classic song "Casta diva by Maria Callas" in the Dailymotion app
Output: Dailymotion; Play the song "Casta diva by Maria Callas"

Input: I want to write the review comment, Perfect! My favorite dessert for this recipe
Output: Unknown app; Write the review comment: "Perfect! My favorite dessert for this recipe."

Input: Open TickTick app and share the wedding plan task on dwbscratchid3@google.com through Gmail
Output: TickTick; Share the wedding plan task on dwbscratchid3@google.com through Gmail

Input: I'd like to forward Google Community team emails to Cerebra Research at dbwscratch.test.id4@gmail.com.
Output: Gmail; Forward Google Community team emails to Cerebra Research at dbwscratch.test.id4@gmail.com.

Input: I am looking for a rental place in St. John, USA, under $4,000, so search for rental properties for me in St. John on the Redfin app.
Output: Redfin; Search for rental properties in St. John, USA under $4,000.

Your test instruction:

Figure C2: AndroidControl cleaning prompt

You are evaluating user behavior within a mobile app. Given a screenshot of the app interface and a description of the user's action, your task is to provide a comprehensive summary of the user's intent and the specifics of their interaction.

Instructions:

1. **Analyze the Input:**
   - Carefully examine the provided screenshot.
   - Interpret the user's action, including any additional information provided.

2. **Extract Key Information:**
   - Identify all relevant elements on the screen (e.g., buttons, text fields, images).
   - Pinpoint the user's specific action (e.g., tap, scroll, input text).
   - Note important details like dates, times, locations, quantities, or text content.

3. **Format the Output:**
   - **Output a newline-delimited list where each item represents a distinct piece of information.
   - **Do not include any explanatory text or labels.** Just the newline-delimited list.
   - Example:
     User viewed product details for iPhone 14 Pro Max.
     User added the product to their shopping cart.
     User selected the '256GB' storage option.

Input:

- **Screenshot:** <img>
- **Action:** {{action}}

Note:

- The action description may include contextual information like text content, direction (e.g., 'swiped left'), app name, or UI element name.
- The screenshot may contain a red bounding box highlighting the interacted element.

Figure C3: Interaction summarization prompt

You are given a summarized user journey, consisting of screen summaries that describe what the user saw on each screen and what they did. Your task is to analyze this journey and infer the user's intent.

Your output should be a concise description of the user's intent that includes:

1. **The user's primary goal:** What were they ultimately trying to achieve?
2. **All apps involved:** List every app used in the journey.
3. **Key actions:** Highlight specific actions within the summaries that reveal the intent (e.g., search queries, filter selections, options chosen). Avoid reporting purely navigational actions.

Important Considerations:

* **Complex Intents:** Longer journeys may involve evolving or multiple intents. Strive to identify the most plausible explanation for the user's actions, even if their initial goal shifted.
* **Conciseness:** Aim for 2-3 sentences that capture the essence of the intent.
* **Output Format:** "AppName; Intent description" (e.g., "Amazon; User viewed the product page for 'noise-canceling headphones', added them to their cart, and proceeded to checkout.")

Your response should contain nothing but the output in the specified format. Do not add any additional text or explanations.

Important: ALL information should be extracted from the summaries. Do NOT introduce any new information.

Output examples:

Expedia; User launched the Expedia app, searched for flights from Paris (CDG) to London (LHR) departing on January 7th, filtered results by "non-stop flights" and "lowest price", and finally selected a British Airways flight departing at 10 PM.

Clock; User opened the Clock app, tapped on the "Alarm" tab, set a new alarm for 7:00 PM tomorrow, toggled the "Snooze" option off, and saved the alarm.

Spotify; User opened Spotify, searched for "holiday music", tapped on the "Create Playlist" button, named the playlist "Christmas 2024", and added songs like "Jingle Bells" and "Silent Night" to the playlist.

Gmail; User opened the Gmail app, opened an email from "Bank of America" with the subject "Your November Statement", tapped on the link to view the PDF statement, then navigated back to their inbox and replied to an email from their boss with the subject "Project Update."

Figure C4: Session-level intent prompt

You are given a summary of a user trajectory and an inferred goals of the user.

Your task is to rewrite the inferred goal in a way that it only contains information that is present in the summaries.
Any information that is not present in the summaries should be removed.

Summaries: {{*[combined summaries]*}}
Inferred goal: {{*[clean goal]*}}

The output format should be a json object with the following format:

```
{{{{
    "facts_in_summaries": ["fact1", "fact2", ...],
    "facts_not_in_summaries": ["fact3", "fact4", ...],
    "rewritten_goal": "the rewritten goal in plain text"
}}}}
```

Figure C5: Label refinement prompt