LLM-powered Multimodal Insight Summarization for UX Testing

Kelsey Turbeville
Jennarong Muengtaweepongsa
Samuel Stevens
Jason Moss
Amy Pon
Kyra Lee
Charu Mehra
Jenny Gutierrez Villalobos
UserTesting, San Francisco, CA, USA

Ranjitha Kumar
University of Illinois at
Urbana-Champaign, Urbana, IL, USA
UserTesting, San Francisco, CA, USA

Abstract

User experience (UX) testing platforms capture many data types related to user feedback and behavior, including clickstream, survey responses, screen recordings of participants performing tasks, and participants' think-aloud audio. Analyzing these multimodal data channels to extract insights remains a time-consuming, manual process for UX researchers. This paper presents a large language model (LLM) approach for generating insights from multimodal UX testing data. By unifying verbal, behavioral, and design data streams into a novel natural language representation, we construct LLM prompts that generate insights combining information across all data types. Each insight can be traced back to behavioral and verbal evidence, allowing users to quickly verify accuracy. We evaluate LLM-generated insight summaries by deploying them in a popular remote UX testing platform, and present evidence that they help UX researchers more efficiently identify key findings from UX tests.

CCS Concepts

• **Human-centered computing** → **HCI design and evaluation methods**.

Keywords

UX research; usability testing; large language models; multimodal insights

ACM Reference Format:
Kelsey Turbeville, Jennarong Muengtaweepongsa, Samuel Stevens, Jason Moss, Amy Pon, Kyra Lee, Charu Mehra, Jenny Gutierrez Villalobos, and Ranjitha Kumar. 2024. LLM-powered Multimodal Insight Summarization for UX Testing. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION (ICMI '24), November 04–08, 2024, San Jose, Costa Rica.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3678957.3685701

ICMI '24, November 04–08, 2024, San Jose, Costa Rica

ACM ISBN 979-8-4007-0462-8/24/11
https://doi.org/10.1145/3678957.3685701

1 Introduction

User experience (UX) testing involves the collection of many types of *implicit* and *explicit* feedback, including clickstream data, survey responses, screen recordings of participants performing tasks, and think-aloud audio [25]. While UX testing platforms automate much of this collection, analyzing the feedback and correlating across different data channels to extract insights remains a manual, time-consuming process for UX practitioners [19, 26]. Although platforms have begun to leverage recent advances in natural language processing — specifically, large, pre-trained transformer-based language models (LLMs) — to summarize written and verbal feedback, these efforts typically do not address data streams without straightforward natural language representations.

This paper presents an LLM-based approach for generating insights from multimodal UX testing data. More specifically, it describes a method for unifying verbal, behavioral, and design data streams into a novel natural language representation that can be used to prompt an LLM to generate insights combining information across all data types. The prompts require the LLM to map each insight back to verbal and behavioral evidence to reduce hallucination and allow users to verify insight accuracy.

We implemented and deployed multimodal insight summarization as an analysis feature on UserTesting, a remote UX testing platform [12]. As participants interact with digital assets during a UX test, the platform captures their interactions, think-aloud audio, and the asset's underlying design data. The summarization feature first prompts an LLM to compute a sequence of natural language descriptions detailing what the participant *said* and *did* ordered by timestamp — a *multimodal transcript* — for each individual test session. Then, the feature prompts the LLM to extract, aggregate, and summarize insights across all the multimodal transcripts taken together. These summary insights synthesize information across verbal, behavioral, and design data from the UX tests, and link directly to timestamped actions in the transcript so that users can quickly inspect the evidence used to generate them.

We report usage statistics and feedback collected from in-product surveys based on the real-world deployment starting August 30, 2023. UX researchers generally find the insight summarization feature makes them more efficient: they are able to extract and identify themes and reduce their overall "time-to-insight."

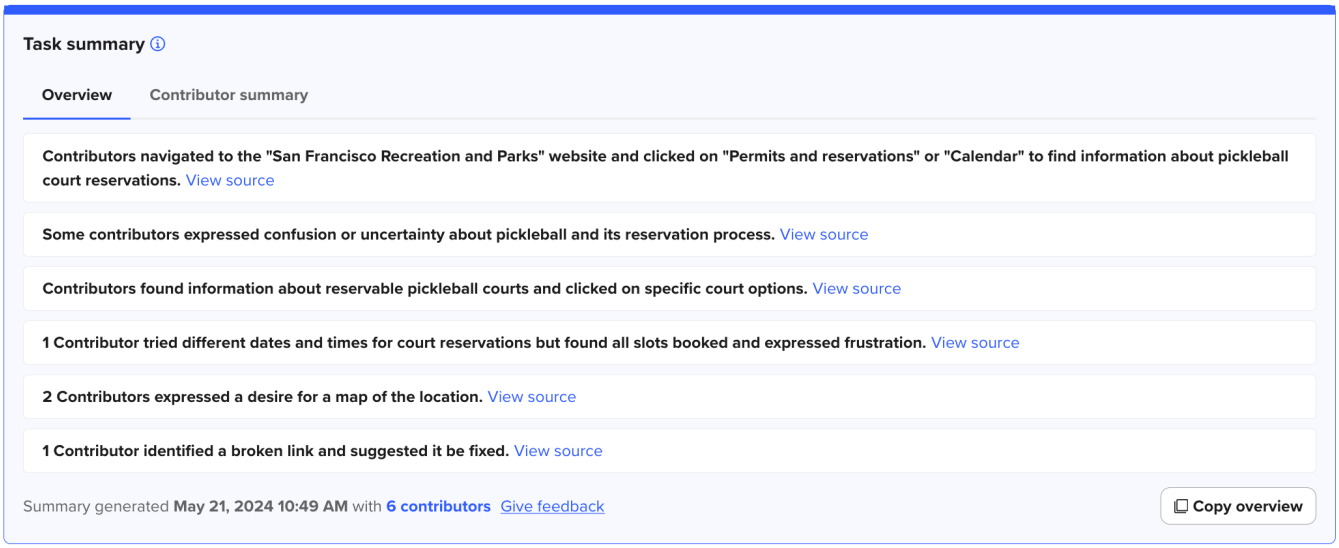

Figure 1: Multimodal insights summarize the results of an unmoderated UX test run on the UserTesting platform with six participants.

2 Related Work

LLMs can potentially support UX research workflows in many ways: assisting in test creation [6], scaling qualitative analysis [30], and even generating synthetic research data [20]. This paper examines how LLMs can scale qualitative analysis by generating insights that can summarize different types of data streams.

2.1 Supporting UX Research and Testing

There are a number of remote UX testing platforms for evaluating digital experiences (e.g. UserTesting [12], Sprig [9], Dscout [3], and Maze [5]), as well as UX research repositories for organizing customer feedback data (e.g. Dovetail [2] and Notably [11]). Several of these platforms support LLM-powered features for summarizing natural language data streams such as verbal and written feedback [1, 7, 10]. This paper introduces an LLM-based approach for generating insights across both non-language data streams — interaction events and UI design (i.e., webpage DOMs) — and natural language ones.

Many research systems have explored strategies for automating analysis, since UX researchers frequently work under time constraints [15, 19]. Some systems help researchers identify anomalous user behavior [21] and extract insights across multimodal data streams [14]; others automatically surface usability issues [18, 27].

The ZIPT system demonstrates that capturing interaction and design data during a usability test makes it easier to quantitatively analyze the results [16]. ZIPT leverages *interaction mining* [17] to generate Sankey diagrams that summarize paths taken by users through an Android application and compute performance metrics such as task completion rate. This paper introduces an approach that combines interaction mining data with think-aloud feedback to derive richer insights that correlate what people *did* with what they *said*.

2.2 Natural Language for UI/UX Understanding

Researchers have have developed ML-powered techniques for generating natural language representations of UI elements [24, 32] and screens [22, 29], powering retrieval and accessibility applications. To support applications such as task automation, researchers have developed methods for generating UI interaction sequences based on natural language task descriptions leveraging transformers [23] and LLMs [28]. This paper leverages LLM prompting to generate a natural language representation for each session in a UX test that interleaves timestamped verbal, interaction, and design data. The LLM is then prompted to extract summative insights across all the multimodal session transcripts.

3 Automating Multimodal Analysis

To evaluate digital design, UX researchers often observe and analyze how users perform various tasks on interfaces. Remote UX testing platforms facilitate these studies, capturing multimodal data streams as participants perform tasks and provide feedback. This paper introduces an approach for automating qualitative analysis and extracting insights across the multimodal data streams captured during UX testing.

3.1 Multimodal Insight Summaries

We implemented and deployed multimodal insight summarization on UserTesting, a remote UX testing platform [12]. Researchers can review the results of an unmoderated UX test as a list of insights summarizing what people *did* and *said* across all sessions (Figure 1).

Suppose a researcher wants to assess and improve the navigability of the San Francisco Department of Parks and Recreation website. The researcher launches a test on UserTesting asking participants to "reserve a pickleball court" to understand how easily users can complete the task. UserTesting then sources participants to take the test, capturing their screens, interactions, and verbal feedback as they perform the task. Once the test is filled and the participant sessions are completed, the platform generates multimodal insights summarizing the results that the researcher can review.

Since the researcher knows that users must navigate to the "Reservations" page and select a specific time to reserve a court, an insight stating that "Contributors navigated to the 'San Francisco Recreation and Parks' website and clicked on 'Permits and reservations' or 'Calendar' to find information about pickleball court reservations" could indicate a high success rate. By clicking "View source," the researcher verifies that all participants did succeed and

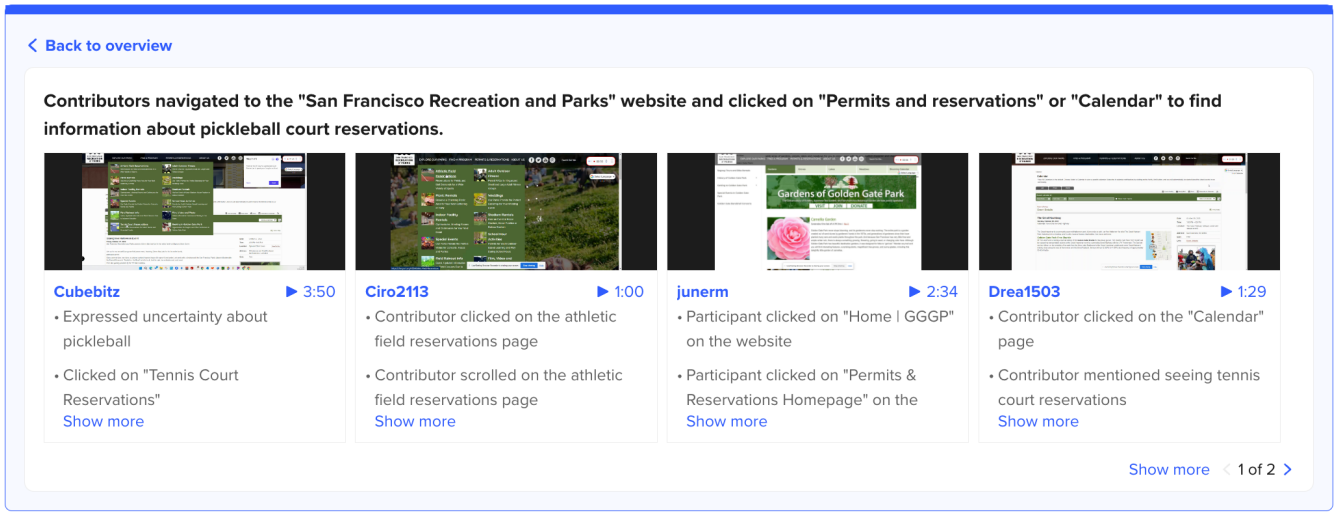

Figure 2: By clicking 'view source,' researchers can quickly validate an insight based on the multimodal session evidence it references.

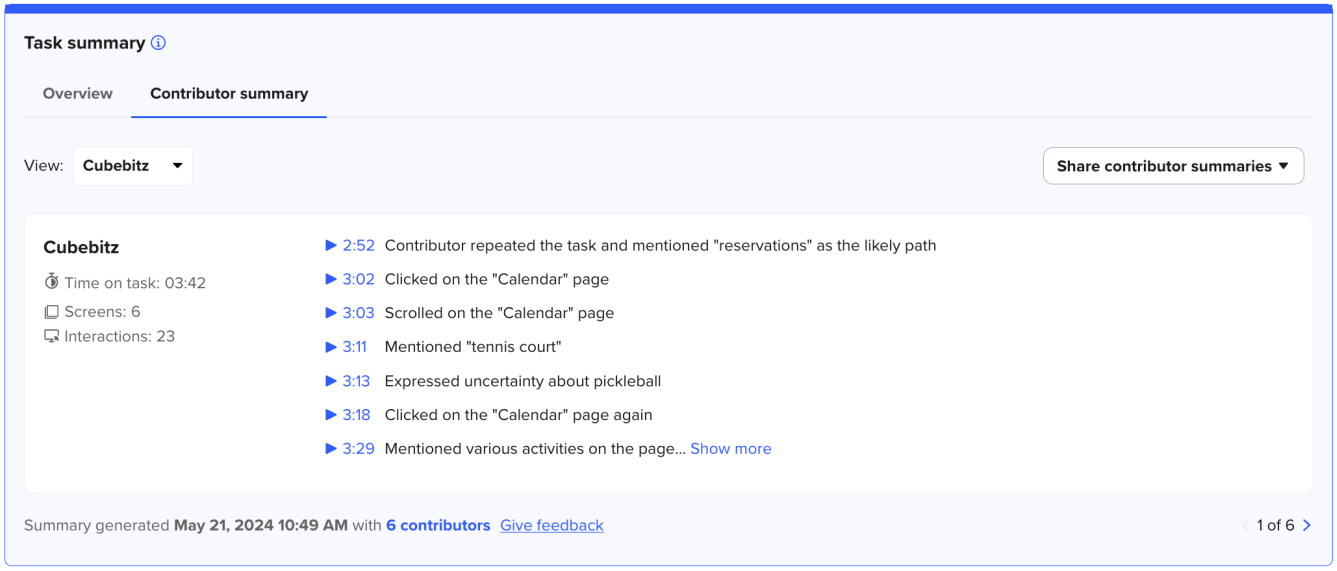

Figure 3: Multimodal session summaries allow researchers to quickly review what a participant *did* and *said* without having to watch an entire session video.

that the insight is correct (Figure 2). Evidence is presented as an array of thumbnails, each representing a video clip from a session annotated with the behavioral and/or verbal data that informed the insight. The researcher can watch the clips for additional context and to review any data streams that were not directly included in the insight—for instance, any comments participants made reviewing the calendar.

Even when all participants succeed at a task, insights based on verbal feedback can capture challenges with the experience and yield suggestions for improvement: "Some contributors expressed confusion or uncertainty about pickleball and its reservation process," "2 Contributors expressed a desire for a map of the location," and "1 Contributor identified a broken link and suggested it be fixed." This example illustrates how UX researchers may rely on multimodal data streams to more deeply understand how users interact with digital experiences.

3.2 Multimodal Session Summaries

Researchers can also review individual sessions as a list of events summarizing what participants *did* and *said* with corresponding video timestamps (Figure 3). These multimodal summaries allow researchers to quickly understand what happened in a session without having to watch the entire video, and each timestamped event in the summary links to the relevant part of the session in case a researcher wants to jump in for additional context. Researchers may also use session summaries as a starting point for manual analysis, mirroring how researchers takes notes while reviewing recorded sessions.

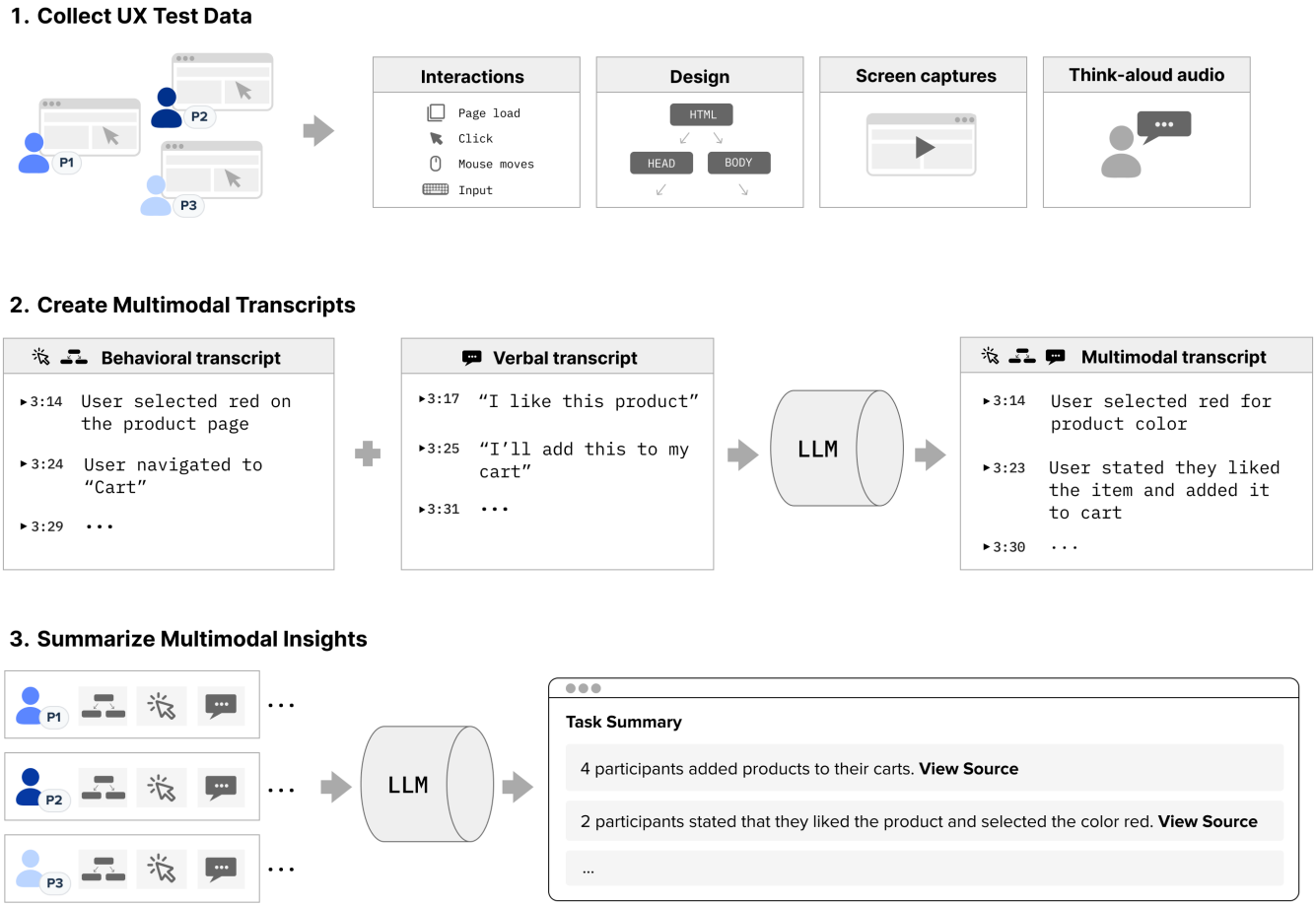

Figure 4: Multimodal insight summaries are generated by chaining together two different LLM prompts. The first prompt generates "multimodal transcripts" for a single session, summarizing across the behavioral, verbal, and design data streams captured during a test. The second prompt summarizes across a set of multimodal transcripts to generate summative insights for a UX test.

4 Generating Multimodal Insights

This paper introduces an approach for generating multimodal insight summaries for UX tests by chaining together two different LLM prompts. The first prompt generates a data structure that chronologically details in natural language what happened across the various data streams captured during a test, producing a "multimodal transcript" for each session. Given a concatenated list of multimodal session transcripts, the second LLM prompt generates the final set of summative insights for a UX test (Figure 4).

4.1 Multimodal Data Stream Capture

During each UX test session, the UserTesting platform concurrently records four different data streams: events/interactions, UI design, screen capture, and think-aloud audio. For web-based sessions, the platform employs a browser extension to capture these data streams, including web page Document Object Model (DOM) trees as the UI design stream.

The event stream records both user-level interactions like clicks and scrolls, and system-level activities such as page loads and form submissions. Every event is tagged with its specific type and timestamp. Given the high frequency of certain events like `scroll` and `mousemove`, a debouncing strategy is applied to filter out redundant events, thereby reducing noise in the data stream.

For each web page at render time, browsers compute a DOM tree: a hierarchical data structure specifying the UI components, their render-time properties, and how they are composed together to define the page's overall layout. Upon page load, an initial snapshot of the entire DOM is taken. Subsequent captures focus solely on changes—added, removed, or modified DOM nodes.

The video stream captures the participant's screen during the test. Each event/interaction has a corresponding screenshot, a video frame from the screen capture that is extracted based on the event's timestamp. These screenshots are used to further analyze the event stream and compute aggregate visualizations (e.g., interaction heatmaps).

The audio stream captures the participant's think-aloud audio during the test. The platform generates transcriptions using a state-of-the-art third party service [8] and segments them into timestamped sentences.

4.2 Multimodal Session Transcript

Based on these captured data streams, the platform employs LLM prompting to generate a "multimodal transcript" for each session, a data structure that chronologically details what a participant *did* and *said*. To input these multimodal streams into an LLM, we developed a novel data representation that combines the non-language streams — events/interactions and DOM trees — with audio transcription.

The platform combines the event/interaction and DOM data streams into a "behavioral transcript" that describes the participant's actions throughout the session in natural language. In the behavioral transcript, each interaction (e.g., click, scroll) is recorded as a text string, detailing the event type, timestamp, page title, URL, and DOM elements relevant to the interaction (e.g., clicked element). More specifically, the displayed text within the relevant DOM element is used with the other event and page metadata to capture an interaction's intent: "Clicked on 'Add to cart' button on 'Product 314' page (https://www.site.com/product/314?color=red)." Through experimentation, we found that the inclusion of full URLs and query parameters proves beneficial; for instance, an LLM can identify that a user applied filters on a shopping website merely by analyzing the URL's query parameters.

The platform merges the verbal and behavioral transcript sentences into an ordered list based on timestamps. When timestamp collisions occur, we found that ordering does not matter, so we simply list the sentences one after the other. A unique identifier tags each integrated event to be referenced later in the output.

The merged verbal-behavioral transcript serves as context input to the first LLM prompt, which is tasked with generating a list-formatted summary of a participant's feedback and behavior for a single session. We found that this input representation allows the LLM to synthesize cohesive insights across user actions and verbal feedback. For instance, an LLM can infer that a participant is interested in a specific product by connecting the verbal feedback about that item with the click interaction on the DOM element linking to that item's product page. Moreover, by incorporating unique identifiers from the original data into each list item, the LLM ensures direct traceability to the source transcripts necessary for subsequent data verification and analysis.

We use few-shot prompting to provide the LLM reference examples it should use while generating output. We found that explicitly marking sections with delimiters such as <PATH> — the verbal-behavioral transcript provided as context input — and <SUMMARY> — the desired list-formatted output — helps improve the quality of results. The basic structure of the first LLM prompt is shown here:

```
Below is a path a participant took during a UX test.
Summarize the path the participant took and include the
    notable data points found.
Rephrase the summary to be easy to digest and quickly
    understand in list format.
---
## Example ##
"Go check out the product on the below page. Explain why
    you chose the product."
<PATH>
24: Clicked on 'Red color' on 'Product 314' page
    (https://www.site.com/product/314)
25: Scrolled down on 'Product 314' page
    (https://www.site.com/product/314?color=red)
26: Said "I liked the red color"
27: Clicked on 'Add to cart' button on 'Product 314' page
    (https://www.site.com/product/314?color=red)
<SUMMARY>
- Participant clicked on the red color option on the
    product page [24]
- Participant liked the red color option and added the
    product to cart [26, 27]
---
"Instruction"
<PATH>
...
<SUMMARY>
```

To generate the final multimodal transcript, the LLM list output is parsed into a JSON representation that preserves references to the original verbal-behavioral transcripts while removing them from summary sentences. The resulting multimodal transcript not only feeds into the next phase of generating insights but also serves as a more digestible session summary that can be exposed to the user (Figure 3). An example multimodal transcript is shown here:

```
...
{ "id": 1,
  "text": "Participant clicked on the red color option on
      the product page",
  "sources": [{"type": "event", "id": 24}]},
{ "id": 2,
  "text": "Participant liked the red color option and added
      the product to cart",
  "sources": [{"type": "event", "id": 26}, {"type":
      "sentence", "id": 27}]}
...
```

4.3 UX Test Insight Summary

To generate summative insights for a UX test, the platform provides a concatenated list of all the multimodal session transcripts as context input to a second LLM prompt. The prompt instructs the LLM to consider commonalities, differences, and anomalies across sessions to give a full picture of user behavior and experiences. For example, in our experiments, we found that LLMs can spot if a product is frequently searched for, or if different search terms lead to the same product being clicked on. The generated insights preserve the multimodality of the data but are expressed in more colloquial language.

In the prompt, clear markers are added between each session's transcript to help the LLM identify the start and end points of each session. The prompt can be adjusted by changing or adding questions, allowing for more targeted insights. Using placeholders for participant IDs improves the ability to trace back to the original data and makes it easier to present and store key findings later on. As with the first LLM prompt, special delimiters like

<CONTRIBUTOR_PATHS> and <SUMMARY> are included to enhance both the reliability and reproducibility of results. The basic structure of the second LLM prompt is shown here:

```
Given the paths that contributors of a UX study took to
     achieve a task,
write a short holistic summary of bullet points to answer
     the below questions:

- Are there any differences and similarities in how
     contributors navigated to achieve the task?
- Were there any points where contributors expressed any
     comments such as sentiment or emotion?
- Did the contributors encounter any issues, blockers, or
     struggled during the task?
---
## Example ##
"Go check out the product on the below page. Explain why
     you chose the product."
<CONTRIBUTOR_PATHS>
participant_123:
1: Participant clicked on the red color option on the
     product page
2: Participant liked the red color option and added the
     product to cart
--
participant_456:
3: Participant dislikes the red option and opted instead
     for the blue option
4: Participant added the product to cart
<SUMMARY>
- [participant_123] and [participant_456] both added the
     product to cart [2, 4]
- While [participant_123] liked the red color option,
     [participant_456] opted for the blue option [1, 3]
---
"Instruction"
<CONTRIBUTOR_PATHS>
...
<SUMMARY>
```

The LLM output is once again parsed into a JSON representation to separate individual line items. The unique identifiers in each line item are extracted and mapped back to their corresponding original events in a two step process: linking to an individual line item in the respective sessions' multimodal transcript and then, to the original source event. This process, known as "walking the references," ensures each insight is anchored in the raw data, thereby enhancing its credibility and interpretability for researchers. This final step outputs a list of insights that spans multiple UX test sessions, along with references to source data rooted in video recordings of the test session. An example of a UX test insight summary is shown here:

```
...
{ "id": 1,
  "text": "[participant_123] and [participant_456] both
       added the product to cart",
  "sources": [{"type": "event", "id": 24}, {"type":
       "event", "id": 25}]},
{ "id": 2,
  "text": "While [participant_123] liked the red color
       option, [participant_456] opted for the blue option",
  "sources": [{"type": "sentence", "id": 26}, {"type":
       "sentence", "id": 27}, {"type": "event", "id": 26}]}
...
```

4.4 Implementation

To capture video, event, and DOM data for user sessions, we leverage UserTesting's Chrome browser extension. The extension provides a basic UI for participants to start and stop recording during a UX test session. Aside from this UI, the extension has two main components: a *content script* that is injected into web pages to record events/interactions and DOM data, and a *background script* that runs at a global browser level. The background script keeps track of page loads, injects our content script into web pages, handles the video recording using Chrome's `desktopCapture` API, and sends the collected data back to our server. User and system events are captured by registering appropriate *event listeners* in the `window` and `document` objects. To capture the DOM and changes in the DOM over time, the extension's content script initially traverses the DOM tree — starting at the `document` node — and records a DOM event with this data. Then, it registers a `MutationObserver` [13] to monitor subsequent modifications..

To implement multimodal insight summarization, we leveraged both the GPT-3.5 16k and GPT-4 models via the OpenAI API. We faced constraints related to context size, which was 16,384 tokens for the GPT-3.5 16k model and 8,192 tokens for the GPT-4 model. These limits affected the number of sessions and the amount of content we could include in a single task summary prompt. Additionally, the API's rate limits necessitated a queuing system to manage requests. To optimize costs and rate-limit availability, we primarily used GPT-3.5 models for individual session summaries. We reserved the more capable GPT-4 model for generating the final, aggregated multi-session insights. The web UI for multimodal insight summaries is built using React.

5 Results

We evaluate LLM-generated multimodal insight summarization by deploying the capability in the remote UX testing platform UserTesting. Between August 30, 2023 and April 30, 2024, UX researchers created 75,464 UX studies eligible for insights summarization. Researchers could choose to watch videos from their UX studies sequentially or review a 'Results' page which included the option to generate an insights summary. During the deployment, 11,030 individual researchers reviewed the Results page, and 3,821 of them (34.6%) created 56,830 insights summaries. The mean number of summaries created was 13.4, and the median number was 5; 66 researchers created 100 summaries or more, while 867 created only one summary.

Researchers also had the option to provide written feedback of the summary, and 61 did so. Of those that provided feedback, 12 worked at companies in the technology space, seven in retail, and

seven in healthcare. The rest were split among communications, manufacturing, travel, defense, and entertainment. Researchers' company size varied from under 50 employees (3 researchers) to over 10,000 (17 researchers).

Researchers expressed that the insights sped their workflow: "This really helps with the fast data analysis and synthesis per task" (P48), "This is a fantastic feature: the source information makes this extremely useful in its current state," (P35), and "This feature is great and a huge time saver!!" (P28).

Nine of the 61 researchers who provided feedback specifically remarked on counting. As LLMs have limited performance with arithmetic tasks, we prompted the LLM to use terms like *a few*, *some*, *most*, and *several* for numbers greater than one. researchers could access counts by checking 'view source' and seeing how many participants were cited. However, researchers remarked that specific counts would better enable their workflows: "I would like to see the statement in this format: 'Most contributors (17 of 20)', 'Several contributors (12 of 20)'… That way we can immediately see how many participants worth of data went into the summary without having to drill in and view." (P45). "It would be helpful to clarify how many the *many*, *several*, *some*, and *a few* are exactly." (P27). Given the importance of accurate counts and specific numbers to UX researchers, future implementations could take a rule-based approach for summing user feedback.

Transcription issues contributed to inaccurate insights and researcher dissatisfaction. P47 wrote, "the response 'one [tester] expressed annoyance at a small cost' was actually the tester saying that they had a small cough and it was annoying." In addition, not all comments participants make are equally useful: "One of the final summary comments was only mentioned because the participants were reading a question on screen, not because it was an answer to the question" (P39).

The current implementation does not take into account individual researchers' needs, and treats all participant transcript as similarly valuable. Two researchers' comments indicate a need for insight customization. P54 suggested that the insights be less tailored to UX specifically, "I'd love to use this for marketing research, ad tests, etc. . . the output makes me think this is super tailored to UX research only," while P49 wrote "I was expecting a summary in a narrative format... [with] recommendations as a conclusion."

Others requested more convenient means to export the insights into their reports or documentation, "It'd be nice to just be able to quick copy one [insight] at a time [in addition to the "Copy all" button]!" (P53), indicating that, for this researcher, some insights were useful or relevant and others were not. Similarly, P51 wrote, "I would like to be able to edit this, fine tune, remove what is not correct and then... download."

6 Discussion and Future Work

Analysis time is the key limiting factor for most qualitative research. Every additional user session adds 20 minutes or more of video review, plus additional time for qualitative coding and synthesis. In addition, the more unstructured data a study generates—and the more kinds of data it considers—the more challenging theme extraction and synthesis become for human analysts. With sufficiently comprehensive, accurate, and verifiable multimodal insights, analysis time could no longer limit scale for qualitative research, broadening the possibilities for future UX researchers to conduct studies with more participants and more types of data.

Feedback gathered from our deployment suggests several avenues for future development. Users rarely reported that the insights' content was inaccurate, but those that did observe inaccuracies reported frustration. In addition, practitioners did not find all insights generated by the system equally useful. A future implementation could allow users to 'hide' or remove less-relevant insights, or 'pin' those that provided more value. Future work could explore the impact of this affordance on users' preferences for a higher-recall system, and the data collected could be used to fine-tune the prompts given to LLMs.

While the system proposed in this paper fully automates the insight generation process, increased control could improve user satisfaction. Practitioners' intentions vary when creating UX tests, and those intentions are not always obvious from the task assigned to participants. A practitioner might, for instance, ask users to navigate a website while thinking aloud, but primarily be interested in organic user reactions to the homepage. Thus, a single prompt cannot meet the needs of all users in all disciplines. Allowing user input for prompting would allow practitioners to direct the LLM to produce insights more pertinent to their specific discipline or use case. Because non-experts struggle with prompt creation [31], a range of pre-generated prompts could be provided, or free-form user input could be included as an auxiliary instruction. Future work could explore how extensive user instructions should be to optimize insight relevance, or to customize insight presentation for dissemination to stakeholders.

In this work, we transform all data streams into natural language representations to generate multimodal insights. Future work should examine whether leveraging inherently multimodal LLMs such as Gemini [4] could improve performance. Can multimodal LLMs automatically generate a multimodal transcript from screen capture video and think-aloud audio without having to construct an intermediate verbal-behavioral transcript? Future research could also explore the extent to which basing summarizations off of multimodal data streams mitigates the risk of hallucination: with more streams of data offering more context, the LLM may not need to supply as much 'missing' material.

Acknowledgments

The authors would like to thank the reviewers for their helpful comments and suggestions. We would like to thank Jerry O. Talton for his multimodal insights; and Kaj van de Loo and Michelle Engle for their technical and product guidance. Finally, we would like to thank the members of the Data Platform and Machine Learning team at UserTesting and Scott Hutchins for their efforts in implementing and deploying this work in a real system.

References

[1] [n. d.]. AI Powered Qualitative Research in Notably | Notably — notably.ai. https://www.notably.ai/features/notably-ai-research. [Accessed 15-09-2023].
[2] [n. d.]. Customer Insights Hub — Dovetail — dovetail.com. https://dovetail.com/. [Accessed 15-09-2023].
[3] [n. d.]. dscout | Flexible, remote, in-context user research — dscout.com. https://dscout.com/. [Accessed 15-09-2023].
[4] [n. d.]. Gemini - chat to supercharge your ideas — gemini.google.com. https://gemini.google.com/. [Accessed 10-08-2024].

[5] [n. d.]. Maze | The continuous product discovery platform — maze.co. https://maze.co/. [Accessed 15-09-2023].

[6] [n. d.]. Maze AI | Elevate Your Product Research — maze.co. https://maze.co/ai/. [Accessed 15-09-2023].

[7] [n. d.]. Note and insight summaries — dovetail.com. https://dovetail.com/help/summarizing-your-data/. [Accessed 15-09-2023].

[8] [n. d.]. Rev AI: Speech to Text API | Speech Recognition Service — rev.ai. https://www.rev.ai/. [Accessed 15-09-2023].

[9] [n. d.]. Sprig | User Insights Platform for Teams Building Better Product Experiences — sprig.com. https://sprig.com/. [Accessed 15-09-2023].

[10] [n. d.]. Sprig AI Analysis - GPT-Powered Real-Time Product Insights | Sprig — sprig.com. https://sprig.com/ai-analysis. [Accessed 15-09-2023].

[11] [n. d.]. Synthesis Platform for User Research | Notably — notably.ai. https://www.notably.ai/. [Accessed 15-09-2023].

[12] [n. d.]. UserTesting Human Insight Platform | Improve Customer Experience (CX) — usertesting.com. https://www.usertesting.com/. [Accessed 15-09-2023].

[13] 2020. MutationObserver. https://developer.mozilla.org/en-US/docs/Web/API/MutationObserver.

[14] Tanja Blascheck, Markus John, Kuno Kurzhals, Steffen Koch, and Thomas Ertl. 2015. VA 2: a visual analytics approach for evaluating visual analytics applications. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 61–70.

[15] Parmit K Chilana, Jacob O Wobbrock, and Andrew J Ko. 2010. Understanding usability practices in complex domains. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2337–2346.

[16] Biplab Deka, Zifeng Huang, Chad Franzen, Jeffrey Nichols, Yang Li, and Ranjitha Kumar. 2017. ZIPT: Zero-Integration Performance Testing of Mobile App Designs. In *Proc. UIST*. 727–736.

[17] Biplab Deka, Zifeng Huang, and Ranjitha Kumar. 2016. ERICA: Interaction mining mobile apps. In *Proc. UIST*. ACM, 767–776.

[18] Mingming Fan, Ke Wu, Jian Zhao, Yue Li, Winter Wei, and Khai N Truong. 2019. VisTA: Integrating machine intelligence with visualization to support the investigation of think-aloud sessions. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 343–352.

[19] Asbjørn Følstad, Effie Law, and Kasper Hornbæk. 2012. Analysis in practical usability evaluation: a survey study. In *proceedings of the SIGCHI conference on human factors in computing systems*. 2127–2136.

[20] Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. 2023. Evaluating large language models in generating synthetic hci research data: a case study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[21] JongWook Jeong, NeungHoe Kim, and Hoh Peter In. 2020. Detecting usability problems in mobile applications on the basis of dissimilarity in user behavior. *International Journal of Human-Computer Studies* 139 (2020), 102364.

[22] Luis A Leiva, Asutosh Hota, and Antti Oulasvirta. 2022. Describing ui screenshots in natural language. *ACM Transactions on Intelligent Systems and Technology* 14, 1 (2022), 1–28.

[23] 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* (2020).

[24] Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. 2020. Widget Captioning: Generating Natural Language Description for Mobile User Interface Elements. arXiv:2010.04295 [cs.LG]

[25] S. McDonald, H. M. Edwards, and T. Zhao. 2012. Exploring Think-Alouds in Usability Testing: An International Survey. *IEEE Transactions on Professional Communication* 55, 1 (2012), 2–19. https://doi.org/10.1109/TPC.2011.2182569

[26] Mie Nørgaard and Kasper Hornbæk. 2006. What do usability evaluators do in practice? An explorative study of think-aloud testing. In *Proceedings of the 6th conference on Designing Interactive systems*. 209–218.

[27] Fabio Paternò, Antonio Giovanni Schiavone, and Antonio Conti. 2017. Customizable automatic detection of bad usability smells in mobile accessed web applications. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–11.

[28] 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*. 1–17.

[29] 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*. 498–510.

[30] Ziang Xiao, Xingdi Yuan, Q Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. 2023. Supporting Qualitative Analysis with Large Language Models: Combining Codebook with GPT-3 for Deductive Coding. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*. 75–78.

[31] JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–21.

[32] Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, Aaron Everitt, and Jeffrey P. Bigham. 2021. Screen Recognition: Creating Accessibility Metadata for Mobile Applications from Pixels. arXiv:2101.04893 [cs.HC]