# Investigating Context-Aware Collaborative Text Entry on Smartphones using Large Language Models

Weihao Chen
Department of Computer Science and Technology
Tsinghua University
Beijing, China
chenwh20@mails.tsinghua.edu.cn

Yuanchun Shi
Department of Computer Science and Technology
Tsinghua University
Beijing, China
Qinghai University
Xining, Qinghai, China
shiyc@tsinghua.edu.cn

Yukun Wang
Department of Computer Science and Technology
Tsinghua University
Beijing, China
wang-yk21@mails.tsinghua.edu.cn

Weinan Shi[*]
Department of Computer Science and Technology
Tsinghua University
Beijing, China
swn@tsinghua.edu.cn

Meizhu Chen
School of Architecture
Tsinghua University
Beijing, China
cmz23@mails.tsinghua.edu.cn

Cheng Gao
Department of Computer Science and Technology
Tsinghua University
Beijing, China
gaoc24@mails.tsinghua.edu.cn

Yu Mei
Department of Computer Science and Technology
Tsinghua University
Beijing, China
meiy24@mails.tsinghua.edu.cn

Yeshuang Zhu
Pattern Recognition Center, WeChat AI
Tencent Inc.
Beijing, China
yshzhu@tencent.com

Jinchao Zhang
Pattern Recognition Center, WeChat AI
Tencent Inc.
Beijing, China
dayerzhang@tencent.com

Chun Yu
Department of Computer Science and Technology
Tsinghua University
Beijing, China
chunyu@tsinghua.edu.cn

## Abstract

Text entry is a fundamental and ubiquitous task, but users often face challenges such as situational impairments or difficulties in sentence formulation. Motivated by this, we explore the potential of large language models (LLMs) to assist with text entry in real-world contexts. We propose a collaborative smartphone-based text entry system, CATIA, that leverages LLMs to provide text suggestions based on contextual factors, including screen content, time, location, activity, and more. In a 7-day in-the-wild study with 36 participants, the system offered appropriate text suggestions in over 80% of cases. Users exhibited different collaborative behaviors depending on whether they were composing text for interpersonal communication or information services. Additionally, the relevance of contextual factors beyond screen content varied across scenarios. We identified two distinct mental models: AI as a supportive facilitator or as a more equal collaborator. These findings outline the design space for human-AI collaborative text entry on smartphones.

## CCS Concepts

• **Human-centered computing → Empirical studies in HCI**; **Ubiquitous and mobile computing**.

## Keywords

Human-AI Collaboration, Text Entry, Context-aware Computing, Smartphones, Large Language Models, In-the-wild Study

**ACM Reference Format:**
Weihao Chen, Yuanchun Shi, Yukun Wang, Weinan Shi, Meizhu Chen, Cheng Gao, Yu Mei, Yeshuang Zhu, Jinchao Zhang, and Chun Yu. 2025. Investigating Context-Aware Collaborative Text Entry on Smartphones using Large Language Models. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3706598.3713944

[*] Corresponding author.

*CHI '25, Yokohama, Japan*

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713944

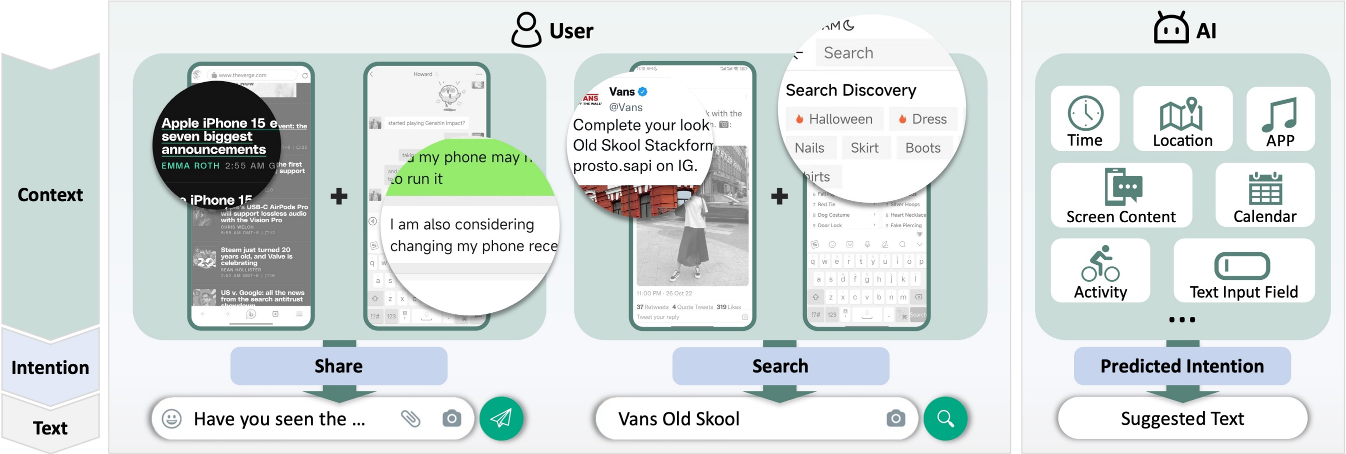


**Figure 1: Mobile text entry behaviors exist in the context of smartphone usage. For example, a user may share read news in an instant messaging app, or go to a shopping app to search for a favorite product after seeing it. We explore a text-suggestion AI that utilizes contextual information on device to infer the user's input intention and suggest texts.**

## 1 Introduction

Text entry is an essential part of everyday smartphone usage, supporting diverse activities such as communication, information retrieval, and note-taking [33]. However, text entry in real-world scenarios poses significant challenges [30]. For instance, situational impairments (e.g., walking or driving) can make typing inconvenient [23], while heavy text input (e.g., composing lengthy responses) may increase cognitive load, leading to difficulties in formulating coherent sentences [44, 45].

We envision a **human-AI collaborative text entry** model, where AI leverages contextual cues to provide relevant input suggestions, while users refine or adapt these suggestions based on situational needs. Specifically, we aim to investigate the potential of large language models (LLMs) in providing this predictive assistance.

The feasibility of this approach stems from the observation that, in many cases, a user's input or underlying intent for text entry can be inferred from contextual information captured by the smartphone [16, 28]. As illustrated in Figure 1, a scenario may involve a user summarizing a news article from one app and sharing it with friends via an instant messaging app. Alternatively, a user could be reading about a product in an app and then want to search for related items in a shopping app.

Moreover, with recent advances, large language models (LLMs) have demonstrated the ability to generate high-quality text based on the given context [9, 17, 57, 66]. Due to their flexibility in input and output, as well as their generality across a wide range of tasks, LLM-driven AI assistants have the potential to become end-users' text entry agents to tackle open and complex input tasks [8]. In this collaborative model, human-to-AI communication can be divided into two complementary channels: active expression, where users intentionally type or provide direct input to the AI, and implicit communication, where users convey through data derived from regular smartphone use [6, 48, 49]. Users can leverage context to reduce their input burden while also actively providing additional instructions to refine the AI's suggestions with minimal effort.

The challenge, however, lies in the fact that real-world collaboration between end-users and LLM-driven AI extends beyond traditional typing, as it involves a more complex dynamic than simple end-to-end predictions. When given inaccurate instructions or incomplete context, LLM suggestions may not reflect the user's true intent [55, 67]. These suggestions might be perceived either as errors or as additional sources of inspiration. Moreover, users may adapt their behavior based on the context and performance of the LLM, and their initial interaction intent could shift over time [50, 53]. This complexity is difficult to predict and understand in advance and could significantly impact design decisions.

Existing research has explored the integration of LLMs with end-user devices to provide context-aware text suggestions for various scenarios, such as automatic form filling on PCs [4], and mobile blogging on smart glasses [10]. Commercial practices (e.g., Apple Intelligence [3]) have similarly explored integrating on-device contextual information into user applications, particularly on smartphones. Despite these advances in specific, predefined scenarios, a comprehensive understanding of how diverse contextual factors and user needs interact with LLM capabilities in open-ended, real-world environments remains largely unexplored. Furthermore, the way users perceive and expect AI systems with human-like capabilities to collaborate in such real-world scenarios has yet to be investigated. Empirical insights from such investigations can inform the design of ubiquitous human-AI collaboration systems.

Motivated by these gaps, we propose a research prototype system: a smartphone-based, context-aware text input assistant (CATIA) using LLMs. CATIA leverages a wide range of contextual factors, such as screen content, time, location, and activity, to provide personalized text suggestions tailored to specific text entry fields. In addition, the system facilitates collaborative refinement of suggestions by users. This system is designed to support the investigation of human-AI collaboration in real-world text input scenarios.

To understand how users interact with CATIA in natural settings, we conducted a 7-day in-the-wild study involving 36 participants. The study is guided by the following research questions to provide both quantitative and qualitative insights:

- RQ1: In what scenarios do users engage with the system, and how does it perform in these contexts?
- RQ2: Why do users choose to collaborate with the system rather than simply accepting its suggestions?
- RQ3: How do off-screen contextual factors (e.g., time, location, and activity) contribute to the prediction of user input?
- RQ4: How do users perceive the effectiveness of different LLMs?

The results show that users accepted the system's suggested text in 82.36% of cases, mainly for interpersonal communication and service-oriented tasks. For the remaining cases, users typically adjusted the suggestions rather than retyping. The system consistently required screen text or text field information (mainly from the final screen), but in some instances, contextual factors such as location, date, time, calendar, and activity provided valuable assistance for quick text suggestions. Our evaluation of various LLMs revealed that smaller, faster, and more cost-effective models have the potential to achieve results comparable to larger models.

Based on these findings, we propose a design space for human-AI collaborative text entry on smartphones, identifying two key mental models: in straightforward tasks like retrieval and quick replies, AI functions as a **supportive, non-intrusive facilitator**; in complex, evolving scenarios such as social interactions, AI acts as an **equal collaborator** offering diverse inspirations. We discuss design choices aligned with these models.

In conclusion, our study provides empirical evidence of the effectiveness of context-aware text input and emphasizes the importance of tailoring AI assistance to specific tasks. It highlights the synergistic interplay between contextual information, LLM knowledge, and user knowledge in the collaborative process. Our work contributes to the future design and development of more adaptive and contextually aware human-AI collaboration systems.

## 2 Related Work

Our research focuses on enhancing mobile text entry by integrating smartphone context and large language models (LLMs) for accurate text suggestions. This section reviews existing works in mobile text entry suggestions, introduces a novel perspective on text input behavior through smartphone usage context, and discusses the synergy between LLMs and context-aware applications.

### 2.1 Mobile Text Entry Suggestions

Text entry on smartphones, a recognized challenge [31, 38, 58], has been the focus of extensive research. Efforts to improve typing performance have addressed issues like the "fat finger" problem, limited screen real estate, and tactile feedback absence [31, 38, 58]. Proposed solutions include optimized keyboard designs (e.g., [7, 43, 65]), error correction or text prediction mechanisms (e.g., [19, 59, 62]), etc., aiming to enhance typing speed, accuracy, and overall user experience [30, 38].

Text entry suggestions in particular, aim to reduce interaction costs through predictive completions [21, 47]. Commercial smartphone keyboards utilize such techniques, primarily leveraging linguistic redundancy via statistical language models [24, 30]. However, these methods typically overlook the role of non-linguistic context in optimizing text suggestions.

Conversely, in search tasks, leveraging context information to ease query input has been a focus [11], such as using location and time for query suggestions [28] or enhancing app search with temporal behavior and app usage data [1]. Numerous market apps employ location data for search box suggestions. These works target accurate item retrieval intents rather than general text input tasks.

Our research aims to merge these approaches, leveraging a rich array of device context information to enhance text suggestions across various smartphone input fields.

### 2.2 Text Entry in the Context of Smartphone Usage

Rather than focusing on well-defined isolated text input tasks as in previous research, we delve into human text input behavior from the perspective of smartphone usage [16, 28]. For example, contextual information from the phone can hint at probable search goals. Local search behaviors and queries users input depend on location, time, and social context [54]. Search topics users input into different apps are related to the apps' functions [12]. Furthermore, Toby et al. [33] study various text entry behaviors within the realm of smartphone app usage. They discover that the type of text entered in different apps correlates with the app's primary function. Compared to non-text entry sessions, text entry sessions involve more apps instead of just one, and users often avoid copy-pasting, opting instead for retyping or other convenient sharing methods like screenshots for interpersonal communication or data transfer, as reported in their study. This underscores that text input is merely a means to fulfill users' higher-level intentions.

Yet, the underlying motives of text input behaviors and their application in enhancing text input technologies remain unexplored. While Bemmann et al. propose a method for collecting richer keyboard logs using Android APIs and categorizing input motives based on input UI metadata [5], their work does not analyze user behavior or offer guidance on how these results can assist in text input. In contrast, our work is the first to combine a broader range of contextual smartphone factors in an open-ended scenario, and use LLMs to infer user intentions from real-time context. This enables our system to not only capture input motives but also guide intelligent text generation, providing personalized suggestions within a collaborative human-AI framework.

### 2.3 Large Language Models for Context-Aware Applications

Generative language models calculate the probability of text sequences and generate the most likely subsequent texts based on provided input. As transformer-based [57] language models like ChatGPT [41] increase in scale [29], they exhibit in-context learning capabilities, where they can learn new tasks via textual prompting without changing model parameters [9, 17]. This ability transcends traditional NLP tasks and generalizes to more complex challenges.

By converting information from different sources and modalities into descriptive text within prompts, LLMs can demonstrate strong context comprehension, supporting aspects like perception, reasoning, task planning, and execution [8, 25, 39, 46, 51].

LLMs enable new possibilities for context-aware applications that adapt services based on user context [15]. These applications often struggle with diverse and unforeseen scenarios because developers cannot predefine all potential user contexts [56]. Unlike machines, which represent contexts in a structured manner, humans convey and understand it through natural language, benefiting from its power and flexibility [13]. Thus, LLMs can express open-ended contexts in natural language and infer implicit information by leveraging their embedded general knowledge.

Several studies have explored using context to provide text services with LLMs. Some have leveraged external environment information sensed by devices. For example, PANDALens utilizes multimodal data (e.g., first-person view, location, time, audio) from smart glasses to provide passage-level text suggestions for auto-blogging in travel scenarios [10]. Others have focused on UI-based information, such as automatic form filling on desktops using web content and text field descriptions [4]. Notably, some studies have investigated how LLMs can analyze and utilize smartphone GUIs, such as for accessibility tasks like hint-text prediction [37] and for GUI testing to generate simulated user input [14, 36].

However, no studies have focused on context-aware text entry on smartphones without predefining the user's scenario or task. Our research fills this gap by combining device-perceivable context information on smartphones—including interface content, physical context, and other factors—and addressing the collaborative nature of user interaction with LLMs in an open-ended context.

## 3 CATIA: Context-Aware Text Input Assistant

Text entry occurs within the broader context of users engaging in daily activities on their smartphones. To explore how using large language models (LLMs) with the contextual information on devices can aid in inferring input text, we designed and implemented CATIA, a Context-Aware Text Input Assistant. The design of CATIA takes into account the following factors.

**Human-AI collaborative text entry.** Existing text entry methods typically involve a person actively expressing thoughts through typing or speaking. This process grants users more control but requires greater effort, such as precise and complete articulation. In contrast, AI assistants generating text using contextual information left by users during device usage represents a more implicit form of expression. Here, users are relieved from the burden of active expression, but the assistant might not always accurately guess the user's intentions. We believe that an ideal approach is a combination of both: the user's active expression and other contextual information complement each other, with both user and assistant collaboratively generating the input text [48]. This is not a replacement for existing text entry methods, but rather an enhancement in scenarios where the context is sufficient to infer text, aiming to reduce the input burden.

**Comprehensible interface.** Most existing commercial input methods include enhancements like auto-completion and error correction, but primarily consider the context of characters already typed in the text box. In contrast, CATIA considers more comprehensive factors such as time, location, activity, app, and screen content, offering text suggestions different from existing tools. This requires users to develop a mental model beyond conventional input methods, understanding CATIA's actions as a more human-like assistant. Therefore, we believe it is necessary to display the contextual information CATIA relies on and provide an explanation for each suggestion. Users may choose to ignore this information after becoming familiar with the system, but its presence is crucial for providing transparency, establishing trust in the AI assistant, and reducing metacognitive demands, particularly in complex or dynamic contexts [2, 13, 34, 35, 53].

Based on these considerations, we implemented CATIA as a collaborative text suggestion system with an understandable interface on Android smartphones. It leverages LLMs to offer text suggestions for the target text field based on context information collected in the short term and allows users to provide additional instructions to guide the generation of more appropriate text. In the following section, we will first introduce CATIA's interface and the collaborative interaction workflow between the user and CATIA. We will then detail the system design, including the device contextual information collected by CATIA and the method of generating suggestions using LLMs. Please refer to Appendix A for example use cases, and Appendix B for implementation details.

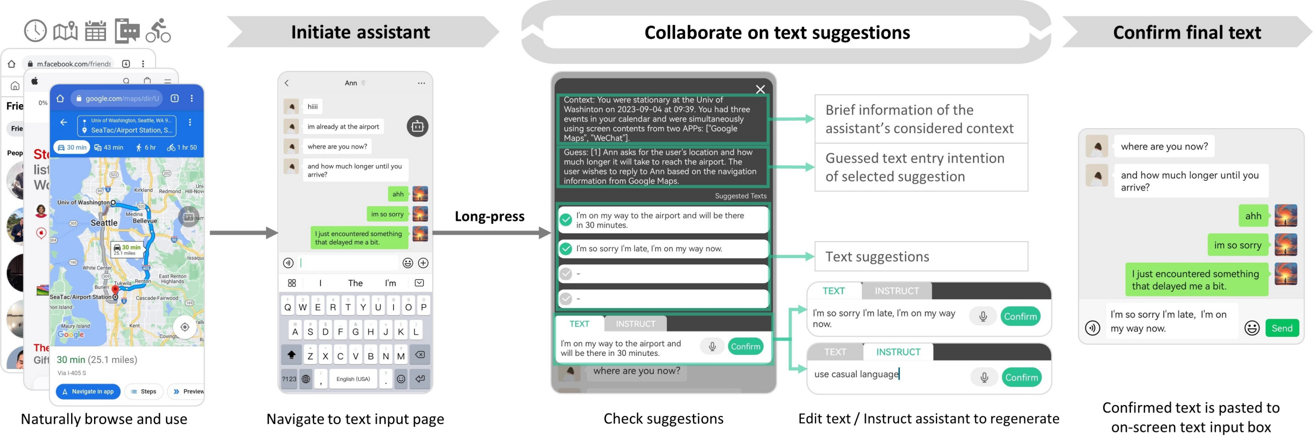
### 3.1 Collaborative Interaction Workflow

The interaction between the user and CATIA, referred to as a session, involves three steps: initiating the suggestion process, collaborating on the text, and confirming the final text. We illustrate this process with an example as shown in Figure 2.

When a user desires suggestions, they long-press the CATIA floating button on the page containing the input text box. The suggestion panel is a draggable floating overlay that can be moved up and down, allowing the user to freely adjust it to view the content below. The assistant simultaneously collects contextual information from the device in the background for text suggestions.

The suggestion panel displays four sections from top to bottom: a brief overview of the context used by the assistant, the assistant's guess of the user's intention, several suggested texts, and a text box for the user to edit input text or provide instructions to the assistant. The context overview briefly introduces the information captured by the assistant. The guessed intention appears when the user selects a suggested text, shown as natural language reflecting the assistant's interpretation of the user's underlying motivation. As previously discussed, these two parts are designed to make CATIA's suggestions more understandable to the user. The panel displays up to four suggested texts, which are dynamically presented to the user character by character to provide immediate feedback.

In the text input field at the suggestion panel's bottom, users can modify the text displayed on screen. Users can click on a suggested text for automatic copying to the `Text` tab for editing, or manually enter text if unsatisfied with the suggestions. To guide the assistant for new suggestions, users enter instructions in the `Instruct` tab and click confirm for regeneration. Notably, when editing text in the panel's text box, the user uses the existing input methods on the phone, allowing them to use familiar input methods, including


Initiate assistant
Collaborate on text suggestions
Confirm final text
Long-press
Brief information of the assistant's considered context
Guessed text entry intention of selected suggestion
Text suggestions
Naturally browse and use
Navigate to text input page
Check suggestions
Edit text / Instruct assistant to regenerate
Confirmed text is pasted to on-screen text input box
Ann
TEXT
INSTRUCT
Confirm
Send
Suggested Texts

hiiii

im already at the airport

where are you now?

and how much longer until you arrive?

ahh

im so sorry

I just encountered something that delayed me a bit.

Context: You were stationary at the Univ of Washinton on 2023-09-04 at 09:39. You had three events in your calendar and were simultaneously using screen contents from two APPs: ["Google Maps", "WeChat"].

Guess: [1] Ann asks for the user's location and how much longer it will take to reach the airport. The user wishes to reply to Ann based on the navigation information from Google Maps.

I'm on my way to the airport and will be there in 30 minutes.

I'm so sorry I'm late, I'm on my way now.

use casual language

**Figure 2: Interaction workflow of CATIA. A complete session involves three steps: capturing context, reviewing and collaborating on suggested results, and confirming the final text.**

typing and voice input. Additionally, the panel facilitates voice input through a press-and-hold voice button for convenience.

After text editing in the `Text` tab, users can click `confirm` for the assistant to copy text to the screen's pending input field, or click `cancel` to end the interaction.

## 3.2 Contextual Information

Upon user activation, CATIA gathers device contextual information in the background to inform text suggestions. This information includes date and time, day of the week, location, activity, connected Bluetooth devices and WiFi, calendar events, screen content, and the text input field to be filled.

Specifically, The location is a human-readable place name, converted from GPS coordinates. The activity, recognized by our custom deep learning model, is categorized as still, walking, running, cycling, or others. Bluetooth and WiFi data provide details on the type and name of currently connected devices and access points. Calendar events include the nearest three past and three future entries, relative to the current time. The text input field, where the suggestion panel is activated, includes three key descriptors: the source app name, a label indicating its intended function, and the pre-existing text content.

Different from other contextual information is the collection of screen content. The system continuously compiles a two-minute queue of recent screen interfaces in the background. Screen content, aggregated from all visible text via Android's Accessibility Service API, feeds into this collection. Exceptionally, for instant messaging apps' chat screens, we employ a recognition algorithm that structures page contents into chat lists pairing senders with messages, beyond mere text enumeration. This approach aids LLMs in capturing key chat information more effectively.

## 3.3 Generation of Text Suggestions

CATIA prompts LLMs to generate text suggestions. In this study, we use a general, widely-adopted method, *avoiding* specific assumptions about particular text input tasks due to the challenge of predicting each end-user's unique scenario with imprecise prior knowledge. This strategy enhances the broader applicability of our conclusions and provides a foundation for future refinements.

The generation process is illustrated in Figure 3. Each suggestion includes an intention description that explains the possible motivation behind the recommendation. We use a chain-of-thought approach [61] to make the LLM sequentially produce the intention and the corresponding suggestion. This token-generation process facilitates more consistent and interpretable reasoning.

Specifically, we use a GPT-4 Turbo model `gpt-4-1106-preview`[1] from OpenAI's chat completions API[2]. The prompt used by CATIA for the initial suggestion primarily contains three parts: task description, input-output examples, and contextual input for this suggestion. The task description mainly informs the LLM of its role as a text suggestion assistant, introduces the content and format of the contextual variables (such as `context.location`, `input_field.app`, etc.), and requests the LLM to think step-by-step and output suggestions. The thinking steps instructed for the LLM include three stages. Firstly, to consider which contents in the contextual information are relevant to the user's input behavior. This is because the input contextual information might be abundant (especially the collected screen text), but not all of it is useful. Secondly, to analyze the possible input intentions of the user, which could be multiple. This is because solely relying on contextual information might not uniquely determine what the user intends to express. Finally, to generate a suggested text for each input intention. Under the guidance of these three steps, the model is required to output up to four intention-suggestion pairs, sorted in the order of what the model considers more likely. After the task description, we include several input-output examples in the prompt, all represented in JSON object format. The last part of the prompt is the collected contextual information for a particular suggestion, represented as a JSON object. Please refer to Appendix C.1 for the complete task description part of this prompt.

[1] GPT-4 Turbo: https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo
[2] OpenAI chat completions API: https://platform.openai.com/docs/api-reference/chat

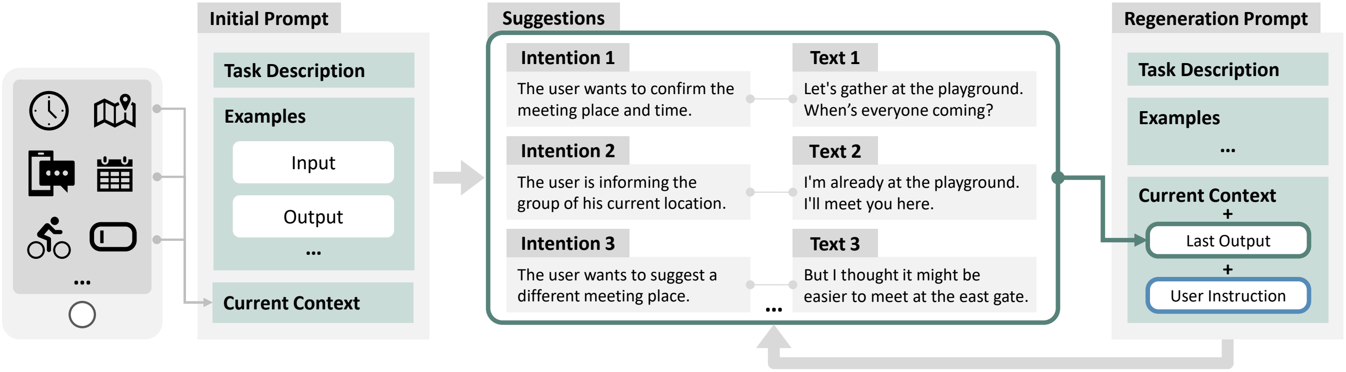

Initial Prompt
Task Description
Examples
Input
Output
...
Current Context
Suggestions
Intention 1
The user wants to confirm the meeting place and time.
Text 1
Let's gather at the playground. When's everyone coming?
Intention 2
The user is informing the group of his current location.
Text 2
I'm already at the playground. I'll meet you here.
Intention 3
The user wants to suggest a different meeting place.
Text 3
But I thought it might be easier to meet at the east gate.
...
Regeneration Prompt
Task Description
Examples
...
Current Context
+
Last Output
+
User Instruction

**Figure 3: The generation process of text suggestions.**

The prompt for regenerating suggestions (i.e., when the user inputs instructions on the panel and requests regeneration) has the same structure as the above one and also contains three parts. The difference is that each input example also includes the user instruction entered on the panel and the results of the previous suggestion. Thus, the task description informs the LLM that the task is to regenerate suggestions, adds introductions to `user_instruction` and `last_output`, and requests the LLM to think in two steps: to consider the intentions based on the provided information, and then output suggested text for each intention. Please refer to Appendix C.2 for the complete task description.

## 3.4 Delay and User Experience

The most time-consuming step in CATIA's suggestion process is the invocation of the LLM. To minimize user waiting time, the system streams the text suggestions on the interface, displaying them character by character as they are parsed from the LLM response. This allows users to start interacting with the suggestions as they appear. Based on preliminary testing, the *first character time* (the time from triggering the LLM call to displaying the first character of the suggestion) typically ranges from 2 to 4 seconds.

It is worth noting that during the LLM invocation delay, the suggestion panel also simulates a streaming animation of the captured context information (see Figure 2), providing continuous visual feedback to users and preventing a lagging experience. The total time for a complete suggestion depends on the number of suggestions and the length of the text. However, the actual delay experienced by the user can also be influenced by factors such as device performance and network communication time.

# 4 In-the-wild Study

To understand context-aware collaborative text entry in real-world scenarios, we conducted a 7-day in-the-wild study to explore how smartphone users interact with our system under natural conditions. In line with the research questions (RQs) mentioned earlier, our approach aimed to capture the following: (1) *real contextual data*, encompassing valuable information that researchers cannot anticipate or construct; (2) *authentic user needs*, by avoiding predefined tasks and supporting flexible, personalized usage in open-ended scenarios; and (3) *in-situ collaborative interactions*—since in real-world contexts, users are better able to perceive and express their needs and engage in collaboration. Considering the potential benefits, ethical approval was obtained from the Artificial Intelligence Ethics Review Board at the authors' university, which is responsible for evaluating AI-related human-subjects research.

## 4.1 Participants

We recruited participants who were native Mandarin speakers, used Android smartphones, and reported a high frequency of text input in their daily lives. Participants were free to withdraw from the study at any time, and any data collected from those who withdrew were deleted after the study. A total of 36 participants (20 males, 16 females) who completed the full study remained, aged between 18 and 27 ($M = 20.89$, $SD = 2.38$). All participants were undergraduate or graduate students from the same university in China. The CATIA system interface was provided in Chinese, and participants used the system in Chinese for text entry during the study. All participants signed an informed consent form and received compensation for their participation.

## 4.2 Procedure

Participants first attended our pre-study briefing session. We introduced the types of context information the assistant could collect and its ability to provide text suggestions in any text box, but we did not explain the underlying principles of how the assistant computed these suggestions. Then, participants installed the CATIA app on their own phones and learned how to use the assistant under the guidance of the experimenters. Data generated during this learning phase was not collected. Once we confirmed that the participants understood the assistant's functions and could use it correctly, we informed them that the 7-day in-the-wild study would begin at midnight the following day.

During the study period, participants were required to keep the app running in the background and integrate it into their daily smartphone activities. No specific tasks were assigned, and participants were free to use the system as needed. However, participants were reminded daily to engage with the system, ensuring consistent usage and minimizing the risk of forgetting.

At the end of the study, we conducted semi-structured interviews with participants via online voice, typically lasting no more than

20 minutes. The interviews focused on three main areas: (1) which scenarios participants found the system useful or not useful, (2) why participants felt the need to collaborate (edit or regenerate text), and (3) how participants understood the system and what their expectations were. Finally, participants removed the CATIA app from their phones.

## 4.3 Data Collection

The mobile service communicates with a remote server responsible for executing the suggestion process and returning the suggested results. Every time the user engages with the assistant, the server logs the captured context, suggestion results, and user interactions within the session. We filter out instances where users trigger and close the assistant multiple times consecutively within the same text field without changes to the page content, as this indicates improper usage. Only the final instance is recorded. All collected data is stored in text-only format, including the visible text on the screen accessed via the accessibility API.

Data collected during the study is stored on a dedicated remote server, with access restricted to a limited set of research team members. If the participant closes the assistant panel through cancellation (e.g., due to accidental touch), or if the session data is incomplete due to technical issues (e.g., unstable network connection), the data from that session will not be analyzed and will be deleted after the study to protect user privacy.

Participants were informed of the data collection process through the consent form and the briefing session. They understood that data collection occurred only when they triggered the assistant, and that they could cancel any session if they felt uncomfortable. They were assured that all data would be anonymized prior to publication. The university's AI Ethics Review Board considered these privacy concerns and approved the experimental procedure.

## 4.4 Post-hoc Analysis

We prompted the LLM to analyze two key aspects of the collected records: (1) the user's true intent behind each input, and (2) the key contextual factors that contributed to inferring the entered text. Although limited than human expert analysis, using LLMs ensures *consistency* in applying the same standards (or biases) across diverse scenarios, enabling more reliable comparative analysis [26]. Specifically, we used the same GPT-4 Turbo model `gpt-4-1106-preview` as in CATIA, with the same prompt structure and similar explanatory text. For each record, the contextual data collected during the study and the final ground truth provided by the user were considered. For the complete task description, see Appendix C.3.

**User Intent**. For each record, we provided the collected contextual information alongside the user's final input text, asking the LLM to infer the user's true intent in natural language. After obtaining the LLM's results for all records, two annotators (the authors) independently reviewed all inferred intents and discussed their categorization criteria. Each annotator then annotated the data according to these criteria and resolved any inconsistencies through consensus. Results are described in Section 5.1.1.

**Key Contextual Factors**. The categorization of contextual factors followed the variable definitions in the suggestion generation prompts. In particular, the screen content was represented as a time-ordered list variable `context.screen_content`, where each element corresponded to a page and included two attributes: page type (either *chat* or *non-chat*, determined by the page layout recognition algorithm) and page text content. The LLM was tasked with determining which page types or content in the list were critical for generating the suggestion. After analyzing all records, we quantified the occurrence of different factors and analyzed their influence on the generated suggestions. Results are described in Sections 5.1.2 and 5.3.

# 5 Results

We collected a total of 2,505 records of valid interactions with CATIA. In 2,063 cases (82.36%), users chose the system's suggestions without making any manual modifications. Among these, in 1,893 cases (75.57%), the system provided the selected suggestion in its initial round, without requiring further regeneration. On average, each session with the system lasted 32.47 seconds. Participants interacted with the system an average of 9.94 times per day, totaling approximately 5.38 minutes of daily interaction.

We identified two distinct patterns of user interaction with CATIA, primarily categorized into two major text entry scenarios: interpersonal communication and service-oriented tasks. In this section, we will explore the unique findings across these two scenarios through the lens of four key research questions: usage scenarios, user collaboration, off-screen contextual factors, and the choice of LLMs. All user texts in Chinese were translated into English.

## 5.1 RQ1: Scenarios and Performance

*5.1.1 Input Field Types and Intentions.* We analyzed the system usage across different text input fields, with the results summarized in Table 1. Overall, the majority of system-assisted text entries were used in interpersonal communication contexts, such as messages and comments, with the remaining entries concentrated on tasks related to information services, such as searches. In these two primary scenarios, the number of times users directly selected the system's suggestion without manual modification was 81.85% for interpersonal communication and 87.08% for service-oriented tasks. The number of correct suggestions received without the need for regeneration was 75.10% and 80%, respectively. These findings indicate that, in most cases, users were satisfied with the suggestions provided by the system.

The text input intentions obtained from the post-hoc analysis are shown in Table 2. Social text fields exhibited more diverse intents; for example, message and comment types could correspond to various social interaction behaviors. In contrast, the intentions for service-oriented input were more closely aligned with the function of the text field.

Social scenarios constituted the majority of cases in the study, which aligns with previous research indicating that most of users' daily text input is entered into communication apps [5, 33]. In interviews, 11 participants mentioned that the assistant's more formal tone was very suitable for certain social situations, such as communicating with elders or strangers, which is consistent with [20]. They also appreciated the diversity of the suggested texts, which were often more appropriate and comprehensive than their own expressions, encouraging them to use the assistant's

**Table 1: Statistics of total sessions, correct suggestions, average length, average manual edit distance, and example apps by input field type, grouped into interpersonal communication and service-oriented tasks.**

| Field Type | Total | Correct | Avg. Length | Avg. Edit Dist. | Example Apps |
|---|---|---|---|---|---|
| **Interpersonal Communication** | | | | | |
| Message | 2,099 | 1,713 | 22.02 | 2.10 | Weixin, QQ, WeCom, Douyin, Taobao |
| Comment | 164 | 140 | 23.18 | 1.28 | Bilibili, Weixin, Douyin, Xiaohongshu, Zhihu |
| Post | 1 | 0 | 15.00 | 33 | Weixin |
| Note | 1 | 1 | 12.00 | 0 | Meituan |
| **Total: 2265 (90.42%)** | | **Correct: 1854 (81.85%)** | | **Avg. Edit Dist.: 2.06** | |
| **Service-oriented Tasks** | | | | | |
| Search | 229 | 200 | 8.90 | 0.90 | Taobao, Pinduoduo, Bilibili, Browser, Ele.me, Amap |
| Form | 7 | 5 | 8.14 | 1.71 | Weixin |
| Chatbot | 4 | 4 | 73.00 | 0 | Wenxin Yiyan (ERNIE Bot), PaiPai Assistant |
| **Total: 240 (9.58%)** | | **Correct: 209 (87.08%)** | | **Avg. Edit Dist.: 0.91** | |

**Table 2: Categories of input intentions, corresponding input field types, and their examples.**

| Category | Field Type | Example |
|---|---|---|
| **Interpersonal Communication** | | |
| Share | comment, post, message | The user intends to share their music experience and recommend a song to a group. |
| Emotional | comment, message | The user is responding to a group chat member Alice who mentioned that they are currently learning to play a new hero, presumably in a game, and the user is offering encouragement. |
| Inquire | comment, search, message | The user intends to inquire about the process for ordering fruit and milk for the next day in a group. |
| Plan | comment, message, chatbot | The user intends to schedule a time to participate in an experiment with Alice by responding to a message. |
| Reply | comment, message | The user is responding to Alice's feedback on a document or presentation, indicating that they have made the suggested changes and are asking for a review or if there are any further additions needed. |
| Comment | comment, message | The user intends to comment on a friend's post, specifically mentioning the post about the theme education. |
| Greetings | comment, message | The user intends to introduce themselves to the group and express a desire for future communication regarding their graduation project preparations. |
| **Service-oriented Tasks** | | |
| Shopping | search | The user intends to search for "Nike shoes" on a shopping app. |
| Video or Song | search | The user intends to search for videos related to "Statue of David" after encountering a video or comment about the topic. |
| Location | search | The user intends to search for directions or information about a building using a navigation app. |
| News | search | The user intends to search for recent events or news related to McDonald's on an app. |
| Learning | search, chatbot | The user intends to search for information about the applications of QR decomposition after previously searching for related mathematical terms. |
| Command | search, note | The user intends to navigate to an app's homepage, likely for entertainment or information purposes. |
| Practice | chatbot | The user intends to deliver a speech and is likely preparing or practicing the speech using an LLM app. The speech emphasizes the importance of literature in enriching the soul alongside the foundation of scientific knowledge. |
| Alias | search, form | The user is adding a new contact on a social app, possibly someone they recently met or need to get in touch with for work or academic purposes. The user is setting a nickname for the contact, which includes the contact's name and affiliation. |

suggestions. Two participants stated that sometimes they didn't know what to say, but the assistant provided a good suggestion that happened to match their intentions. P5 mentioned: "*When responding to a notification about an event, the assistant helped me by asking, 'Where is this address? Could you send me the location?' and I realized that I actually didn't know where the place was.*" Three participants also mentioned that sometimes assistant responses inconsistent with their own style in informal situations created an entertaining effect and were also welcome. For example, P22 said: "*When commenting on posts, it's quite amusing. Sometimes I can't think of a comment, but the assistant's suggestion is unexpectedly clever.*"

The remaining scenarios are mainly in search scenarios, representing participants accessing information or services through queries. Eight participants also mentioned that they liked the assistant's suggestion of search queries based on their phone usage history. Although the specific proportions of scenarios vary from person to person, the typical usage patterns involved here are of reference significance.

### 5.1.2 *Usage of Screen Content.*

The usage of screen content was determined based on the key contextual information analyzed in Section 4.4. The most frequently used factors came from screen content and input field information. Within screen content, page type (chat or non-chat) and text content play the main roles. In the input field, the label describing its function is the most important. Other contextual factors are analyzed in section 5.3.

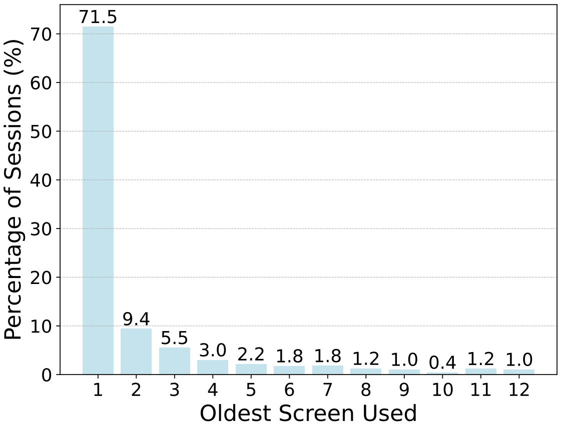


**Figure 4: Distribution of sessions by oldest screen used for text suggestion. Numbers on the horizontal axis represent the reverse order of the screens (i.e., how many screens back).**

Across all sessions, the number of screens the system used ranges from 1 to 7, with an average number of 1.04 ($SD$ = 0.27). We also considered the oldest screen needed for each session (how many screens back), and the distribution of the sessions is shown in Figure 4. In most cases, the important information is contained within the last screen. This is understandable in social scenarios, as message chats, social media interactions, etc., are mostly fully displayed within one screen.

Four participants mentioned that the assistant's ability to remember and summarize across screens can reduce the burden of human input. Five participants mentioned that in situations where the context is relatively consistent and matches their own intent, the assistant's suggestions are very accurate, such as when all messages in a group chat are talking about the same topic, or when the content they want to share is exactly the post they just saw.

## 5.2 RQ2: User Collaboration

User collaboration with the system occurred in two phases: one involved providing additional instructions to regenerate suggestions, and the other involved manually modifying the suggestions after selection or retyping the text.

### 5.2.1 *User Instructions for Regeneration.*

296 sessions with regeneration attempts were collected in the study, and a total of 354 instructions were proposed in these sessions, with each session having 1-4 instances of instruction ($Mean$ = 1.20, $SD$ = 0.52).

Two annotators (the authors) reviewed all the instructions provided by users, discussed, and categorized them into these five types: **Expressing Intention**: Briefly expressing the text's intent, leading the assistant to return an expansion matching the intention; **Emphasizing Existing Keywords**: Providing keywords to focus on corresponding information in the context; **Providing New Keywords**: Supplying keywords to add information not present in the context; **Giving Text Examples**: Directly providing text examples; **Tone / Writing Style Adjustment**: Making specific requests to modify the tone, style, or other characteristics of the text. The first three are instructions targeting intent, while the last two target the text itself. Statistics and examples of each instruction type are presented in Table 3.

In different scenarios, users had varying tolerance levels for the quality of regenerated suggestions. The average number of regeneration attempts in interpersonal communication and information service tasks was 1.21 and 1.12, respectively. In social scenarios, users were generally more willing to spend time interacting with the assistant to achieve a more satisfactory result through collaboration. This was confirmed in participant interviews, where they mentioned "*attempting multiple adjustments to the assistant's suggestions*" when sending messages or posting comments. However, in faster-paced tasks like searches, if the suggestions remained inaccurate after a retry, users tended to resort to manual input.

### 5.2.2 *Manual Edits.*

All participants agreed that, despite the need for manual edits in some cases, the assistant's ability to provide an initial draft for longer texts and social interactions reduced the burden of typing from scratch.

We calculated the Levenshtein distance (edit distance) for each modified record, measuring the minimum number of insertions, deletions, or substitutions required to transform a suggested text into the final version [32]. The average edit distance in interpersonal communication scenarios was 2.06, while in service-oriented tasks, the average edit distance was 0.91. This indicates that users made relatively more edits in social interaction scenarios.

We examined 442 sessions where users made manual edits to categorize the types of edits they performed. Two annotators (the authors) initially reviewed the collected contextual information, including the last selected text (if any) or all suggested texts (if none were chosen), and the final confirmed text. They then discussed and established classification criteria, and independently annotated all the data based on these criteria. Any discrepancies in their annotations were discussed to reach a consensus.

We identified three major categories of manual edits, each reflecting different types of user engagement with the system. Examples of each type are presented in Table 4. These categories provide a more nuanced view of how users interact with the system's suggestions and refine them to suit their needs.

The first category, **High-Quality Text with Minor Changes**, reflects cases where the system's suggestions were largely accurate, and users primarily made small adjustments to refine the tone or remove redundant information. Participants often found that the suggestions captured their intent but required stylistic changes to match their personal communication style. For example, P14 mentioned, "*It's easy for people to tell it's not me, like being too serious or the humor is off,*" indicating that minor modifications were necessary to make the text feel more natural and aligned with the user's tone. In these cases, users valued the suggestions as a

**Table 3: User instruction types, their percentages and examples.**

| User Instruction Types | Examples |
|---|---|
| Expressing Intention (259, 73.16%) | "*I want to help my classmate come up with a solution for a leave application*"; "*Express fear*" |
| Emphasizing Existing Keywords (38, 10.73%) | "*League of Legends*"; "*Pop Mart*"; "*Emergency department*" |
| Providing New Keywords (13, 3.67%) | "*Game*"; "*Doll*"; "*Rabbit*"; "*Riemann sum*" |
| Giving Text Examples (19, 5.37%) | "*Today is quite windy. Sure you want to bring badminton?*"; "*OK*" |
| Tone / Writing Style Adjustment (25, 7.06%) | "*Within three words*"; "*Be funnier*"; "*Don't be too polite*" |

**Table 4: Manual edit types and examples categorized by suggested text characteristics and user modifications. Green-highlighted text represents additions and red-highlighted text represents deletions.**

**A. High-Quality Text with Minor Changes (231, 52.26%)**

**A1. Redundant Information (93, 21.04%)**: The suggested text accurately conveys the intent but includes some redundant information, which the user easily removes.

- Modified " *Bob, you've worked hard, health is the most important; we'll handle the questionnaire, you just rest well.*" to " *You've worked hard, health is the most important, you just rest well*".
- Wanted to set a contact nickname and modified " *Alice - Dream Booster Activity*" *to* " Alice".
- Wanted to search for a Q&A post and removed " *related discussions*" in " *Nihilism related discussions*".

**A2: Style Differences (138, 31.22%)**: The suggested text perfectly matches the user's intent but differs in style, punctuation, or emphasis. Minor adjustments refine the text to suit the user's preferences.

- Modified " *Oh, good reminder, I'm a bit busy these days, might write it later.*" to " *Oh, good reminder, completely forgot* 😣😣😣*, might write it later.*"
- Modified " *The pre-defense meeting time at 11:30 am tomorrow is no problem for me.*" to " *Received,* no problem."
- Was suggested " *I'm going to sleep, hopefully, I'll feel better when I wake up tomorrow.*" and manually entered " Sleeping, hope everything's fine when I wake up."

**B. Reusable Text (158, 35.75%)**

**B1. Misaligned Intention but Useful Structure (43, 9.73%)**: The suggested text may convey an opposite intent, but the overall structure is useful and requires minimal changes.

- Modified " *Haha, we really have the mindset of the young but the lifestyle of the old*" to " *Haha, we really have the mindset of the old but the lifestyle of the young*".
- Modified " *If you have questions about the content in that picture, I can try to explain.*" to " *If you have questions about the content in that picture, there's nothing I can do about it*".
- Modified " *It's snowing? It started so early, the weather changes so much.*" to " *It also snowed a bit in Beijing yesterday, but just a little*".

**B2. Partially Accurate Content (115, 26.02%)**: The suggested text captures part of the user's intent but includes some erroneous or incomplete information. Most of the text is directly usable with minor edits.

- Modified " *I just checked the bonus calculation, no issues.*" to " *No issues on the science association's side.*"
- Modified " *Indeed, the portability of the Steam Deck and PC is not as good as handheld consoles, but the gaming experience and graphics will be much better.*" to " *Indeed, the portability of the Steam Deck and PC is not as good as handheld consoles, but the gaming experience and graphics will be somewhat better*".
- Wanted to search for a tutorial and was suggested " *GeoGebra Tutorial*", but manually entered " *How to draw a parametric equation using GeoGebra*".

**C. Text Requiring Extensive Modifications (53, 11.99%)**

The suggested text is not relevant to the user's intention, necessitating significant edits.

- Manually entered " *Good morning*" to initiate a new interaction, but the suggested texts were all responses to the group chat history.
- Manually entered " *Isn't the expected outcome just a bachelor's thesis?*", but the suggested texts were all other comments about the thesis proposal defense.
- Wanted to search for a location in the map app, but no relevant suggestions were provided.

useful first draft but felt the need to personalize the output to better reflect their unique voice.

The second category, **Reusable Text**, involved scenarios where the system provided a structurally or literally sound suggestion, but the generated text did not fully align with the user's intended meaning. While users reused the general framework of the suggestion, they needed to make more substantial content adjustments to better capture their exact intent. P10 highlighted this challenge, stating, "*Most of the time, it cannot capture what I want to say in conversations with peers, as the context might not be very relevant.*" This was particularly evident in information service tasks, where the system's suggestions were close to the desired output but required significant edits to convey the precise message. In these cases, the system helped users by providing a starting point, but the final expression of intent required further refinement.

The third category, **Text Requiring Extensive Modifications**, occurred when the system's suggestions were largely irrelevant to the user's intent, requiring major revisions or a complete rewrite of the text. In these cases, both the form and content of the suggestions missed the user's expectations, often due to misinterpreting the context or failing to capture the user's intent altogether. For example, P9 noted, "*After reading posts and then searching for shopping content, there is a lot of content in the posts, and the assistant fails to precisely capture what I want.*" Similarly, P16 commented, "*In multi-scenario situations, it tends to link completely unrelated scenarios together,*" indicating that in more complex, context-heavy tasks, the system struggled to produce relevant suggestions.

These findings suggest that while the system can act as a valuable drafting tool, the depth of user involvement varies significantly based on how well the suggestions align with the user's intent and the complexity of the task.

## 5.3 RQ3: Off-Screen Contextual Factors

All participants recognized CATIA's ability to incorporate contextual information into its text suggestions. While the most frequently used cues were on-screen, off-screen factors such as location, date and time, calendar events, day of the week, and user activity were also important in specific scenarios, as illustrated in Table 5.

Participants found off-screen cues particularly useful in situations where they needed to respond quickly without typing much. For example, P18 mentioned: "*Once a classmate asked me if I had arrived at the cafeteria. The assistant, using my location near the library, suggested responses like 'On the way' or 'Please wait.' This allowed me to quickly reply even though I was riding a bike and couldn't type easily.*" Similarly, cues such as the time of day or day of the week were helpful in shaping responses that aligned with daily schedules, such as reminders, brief status updates, or greetings.

Although these factors were less common overall, they provided valuable support by improving the speed and efficiency of routine tasks. They helped reduce the cognitive load on users, allowing them to send contextually appropriate messages with minimal effort.

## 5.4 RQ4: Perceived Effectiveness of Different LLMs

To implement LLM-assisted text entry in real-world environments, it is essential to consider the model's performance, real-time responsiveness, and deployability. We aim to explore how different LLMs may influence context-aware text suggestion tasks. Given the inherently subjective nature of suggestion quality evaluation, we conducted a post-study assessment where participants evaluated the effectiveness of various LLMs based on the data collected during the in-the-wild study. Participants' recall of the context was grounded in the recorded textual data. Although this evaluation cannot fully replicate the real-world setting of the study, we believe that this preliminary comparison still provides valuable empirical insights.

*5.4.1 Setup.* We primarily considered OpenAI's GPT series [42], Zhipu AI's GLM-4-9B [22], and Alibaba Cloud's Qwen2-7B-Instruct [64]. These models were chosen due to their strong performance in Chinese language understanding, as demonstrated in the 2024 August benchmark report from SuperCLUE[3] [63]. We used the platform-provided APIs for all tests and evaluated models smaller or faster than GPT-4 Turbo (`gpt-4-1106-preview`), which was used in the in-the-wild study in Section 4. We also utilized the provided interface to fine-tune the models with full parameters and deployed private instances when necessary. LLM calls were made using the same prompt and parameters as in CATIA.

**Dataset**. Given that the minimum context window for the tested LLMs is 8k tokens and that some models have input prompt length limitations, we selected 591 data samples from the study that met these criteria. We split these samples into training and testing sets in a 9:1 ratio, ensuring that the proportions of interpersonal communication and service-oriented tasks remained unchanged. As a result, the training set contained 533 samples, and the test set contained 58 samples. The training set was used for fine-tuning, with the user-confirmed final text serving as the sole ground truth in the output.

**Procedure**. After obtaining the predictions of different LLMs on the test set, we invited the original participants of each record to recall the context at the time and, based on consistent criteria, either select the best option for each anonymized model or choose none if unsatisfied. The contextual information was converted from JSON into a human-readable natural language list. Participants were also able to view the final text they confirmed during the in-the-wild study for better recall of the context.

**Metrics**. We considered top-1 acceptance, top-4 acceptance, first character time of the first suggestion, total response time, API cost, and fine-tuning cost. Top-1 acceptance was chosen because we requested the models to output suggestions in order of likelihood (consistent with CATIA), reflecting precision. Additionally, the fine-tuned models produced only a single suggestion. Top-4 acceptance was used because we requested a maximum of four suggestions, which covers the full set of possible options and is in line with the settings of in-the-wild study.

*5.4.2 Results.* Table 6 presents the performance of a range of different LLMs, highlighting the best-performing models and their respective metric results. The newer model `gpt-4o-2024-08-06` outperforms `gpt-4-1106-preview` in terms of the acceptance rates, demonstrating an evolved performance of available LLMs. While smaller base models show slightly reduced performance, their results are still competitive. Furthermore, while the fine-tuned models do not show a significant improvement in top-4 acceptance rate compared to the base models, they exhibit a notable increase in top-1 acceptance rate. This suggests that when the task limits the number of output options, fine-tuning has the potential to improve the precision of the suggestions.

As for cost, all models are cheaper than `gpt-4-1106-preview`. In terms of speed, all models, except for `glm-4-9b` and its fine-tuned version, perform faster. These results indicate that these models have the potential to significantly enhance user interaction responsiveness at a much lower cost.

[3] SuperCLUE: https://superclueai.com/

**Table 5: Use cases and corresponding examples of different off-screen contextual factors that influence text suggestions.**

| Context Factors | Use Cases | Examples |
|---|---|---|
| Location (158) | The user clearly expresses their current location or the place they are going to. | “*I am eating in the cafeteria.*” |
| | The information the user wants to convey can be inferred from the location. | “*It’s really noisy, I just want to escape!*” |
| Date and time (99) | The user’s expression is related to the season or date. | “*Let’s wait until the weather warms up to meet. No need to brave this cold!*” |
| | The user clearly expresses an approximate time. | “*It’s already late, I need to sleep. Let’s talk tomorrow.*” |
| Calendar (45) | The user mentioned plans for a specific time. | “*I’m free tomorrow afternoon during the first period, and I’m available in the evening as well.*” |
| | The user’s mood can be inferred from the schedule. | “*This week’s schedule is packed, and I feel like I can barely keep up.*” |
| Day of week (10) | The user clearly mentioned a specific day of the week. | “*It’s Monday! You can start testing the AI and see if it can pass, haha.*” |
| Activity (7) | The user clearly mentioned an activity. | “*I’m already biking on the road, I’ll be there soon.*” |
| | The user’s expression can be inferred from the activity. | “*Wait for me two minutes, I’ll be there soon.*” |

**Table 6: Performance and cost comparison of various LLMs with or without fine-tuning. Models with “(ft.)” indicate fine-tuned versions. Bold highlights the top 3 best-performing models in each acceptance rate metric. Asterisks (*) indicate that the top-4 acceptance rate for fine-tuned models matches the corresponding top-1 acceptance rate from the left. The values for the time metrics represent the mean, with standard deviations in parentheses. API cost is shown as the average cost per 1,000 calls.**

| Model | Top-1 Acpt. Rate (%) | Top-4 Acpt. Rate (%) | First Character Time (s) | Total Response Time (s) | API Cost ($ / 1k calls) | Fine-tuning Cost ($) |
|---|---|---|---|---|---|---|
| gpt-4-1106-preview | 37.93 | **60.34** | 2.99 (0.46) | 7.17 (1.80) | 38.97 | / |
| gpt-4o-2024-08-06 | 43.10 | **67.24** | 2.21 (0.55) | 3.20 (0.80) | 7.92 | / |
| gpt-4o-mini-2024-07-18 | 18.97 | 55.17 | 2.67 (2.74) | 3.70 (3.27) | 0.48 | / |
| glm-4-9b | 20.69 | 53.45 | 4.22 (0.98) | 7.68 (1.71) | 0.80 | / |
| qwen2-7b-instruct | 25.86 | 53.45 | 2.99 (0.73) | 4.02 (0.88) | 0.42 | / |
| gpt-4o-2024-08-06 (ft.) | **56.89** | **56.89** * | 2.54 (0.49) | 2.68 (0.54) | 11.26 | 113.19 |
| gpt-4o-mini-2024-07-18 (ft.) | **48.28** | 48.28 * | 2.83 (1.02) | 3.02 (1.04) | 0.91 | 13.58 |
| glm-4-9b (ft.) | 44.82 | 44.82 * | 3.30 (0.39) | 3.64 (0.63) | / | 22.93 |
| qwen2-7b-instruct (ft.) | **50.00** | 50.00 * | 2.62 (0.55) | 2.75 (0.55) | / | 2.91 |

## 6 Discussion

Based on the results of our user study, we identify two distinct mental models—*facilitator* and *collaborator*—that reflect how users perceive and interact with AI-driven text entry systems. These models inform several design dimensions and choices. In this section, we first introduce these models and relate them to prior work, then explore the design space for context-aware collaborative text entry on smartphones, and conclude with a discussion of potential ethical issues.

### 6.1 Mental Models: Facilitator and Collaborator

Our findings suggest that users adopt two different mental models when interacting with collaborative text entry systems (Figure 5). These models inform how AI systems should be designed to meet varying user needs based on the task at hand.

**Facilitator**. In this model, the AI serves as a supportive tool that automates tasks or reduces user effort, without requiring significant manual input. This model is commonly applied in task-driven contexts, such as filling out forms, entering structured data, or performing searches. Our study also revealed that the Facilitator model extends to certain social scenarios, particularly in cases where users face situational impairments (e.g., walking, biking) or when speed is prioritized over collaboration. For example, in situations requiring quick replies, users do not expect to engage in a creative process

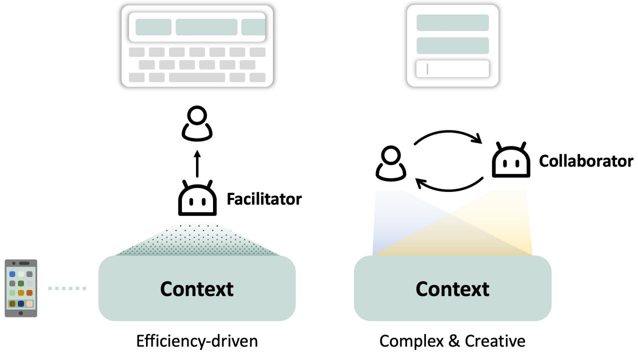


**Figure 5: Mental models of human-AI collaborative text entry.**

with the AI; instead, they value precise, contextually aware suggestions that can be easily selected with minimal effort.

**Collaborator**. In contrast, the Collaborator model represents a more dynamic interaction where the AI acts as an equal partner, offering diverse suggestions to inspire or assist the user in creative or exploratory tasks. This model is more relevant in scenarios where the user's intent is *flexible or evolving*, such as drafting social messages, brainstorming ideas, or composing longer texts. In these contexts, users expect the AI to generate multiple options, allowing them to explore different possibilities. Unlike the Facilitator model, which focuses on precision, the Collaborator model thrives on diversity and creativity. Users may trigger the AI to generate suggestions even when they do *not* have a fully formed intent, using the AI as a source of inspiration to guide the collaboration. The focus here is on offering varied options, allowing the user to select, refine, or adapt the AI's suggestions as needed.

These distinct models emerge not only due to the different nature of tasks but also because the advanced capabilities of LLMs have made users more aware of AI's potential to intervene in a wider range of complex scenarios. Despite our study employed the same system and explicitly prompted users with "this is the same assistant," users in open-ended, daily environments instinctively adjusted their expectations based on the direction and function of the text. Furthermore, these models influence both how users interact with AI and the cognitive effort required in these interactions. Depending on the model adopted, users experience different levels of metacognitive demand. For instance, users in the Facilitator model generally engage in lighter metacognitive activities, while the Collaborator model requires more active involvement in refining and monitoring suggestions, guiding the AI through iterative processes. This aligns with previous work emphasizing the metacognitive demands posed by generative AI [50, 53].

Beyond individual text composition, CATIA was predominantly used for interpersonal communication, accounting for 90.42% of cases in our study. This underscores the growing role of AI in helping users articulate their thoughts and refine their messages in social interactions. Consistent with Fu et al.'s findings [20], users in the Collaborator model expressed a desire to leverage AI to better articulate their thoughts and produce text beyond their usual capabilities. Users also noted that AI excels in formal communication. Additionally, our Facilitator model aligns with Fu et al.'s suggestion that AI can serve as a "*communication expediter,*" providing rapid replies based on the communication context.

While these findings align with Fu et al.'s work, we also identified notable differences. In contrast to their observation that AI use may be "unnecessary and undesired" in informal, low-stakes contexts (e.g., casual chats), users in our study, despite expressing concerns about inauthentic replies, still relied heavily on the system in these contexts. We hypothesize three possible reasons for this. First, given the ubiquitous nature and inherent constraints of text input on smartphones, the system's benefits—whether for quick replies or generating initial drafts—seem to outweigh concerns about authenticity. Second, unlike Fu et al.'s study, where participants had to manually provide context for the AI, our system eliminated this requirement. This reduced barrier likely increased users' willingness to adopt the system. Finally, the sample size and duration of our study may have introduced potential biases. These factors warrant further empirical investigation.

## 6.2 Design Space for Context-Aware Collaborative Text Entry

Building on our exploration of the mental models, we propose the following key dimensions and design considerations.

*6.2.1 Contextual Cues for Model Differentiation.* The distinction between the Facilitator and Collaborator models can be effectively inferred from various elements on the smartphone, such as text field hint text and interface structure [5, 37], making this differentiation feasible in practical system design. Structured fields like search bars, form inputs, or notes typically suggest the need for Facilitator-like suggestions—precise, concise, and directly embedded into the workflow. In contrast, text fields like message composition areas are better suited to the Collaborator model, where the AI can provide a wider array of creative or exploratory options.

Moreover, the system can leverage off-screen contextual factors, such as the user's location, activity, or time of day, alongside on-screen interface cues. For instance, if the user is on the move or in a meeting room with potential situational impairments, the system might prioritize quick, task-oriented suggestions (Facilitator). Conversely, in less constraint scenarios, such as when the user is engaged in a creative or brainstorming activity, the system could incorporate the user's physical context to offer suggestions that broaden their thoughts (Collaborator). Different users may prefer different ways of leveraging contextual cues to inform their mental model, which suggests that the system should continuously learn from user interactions and adapt over time to better align with individual preferences.

*6.2.2 Initiation: System vs. User.* A critical design dimension is who initiates the interaction—the system or the user. In CATIA's current implementation, suggestions are manually triggered by the user. However, based on our interviews, whether an additional trigger step is necessary may vary depending on the scenario.

**System-Initiated Suggestions** (Facilitator). In task-oriented activities, users expect the AI to provide automatic and unobtrusive suggestions. These suggestions should be contextually relevant, minimizing the need for manual intervention. Unlike existing input methods and app-based suggestions, which are often binary (on or off), users in these situations prefer the system to offer recommendations only when there is high confidence in their accuracy. Any incorrect or irrelevant suggestions risk breaking user trust, especially in contexts where speed and precision are crucial (e.g., quickly filling a form or executing a search). To minimize distractions, the system can predict user intent to input text based on past digital traces and only offer automatic suggestions in user-specified text fields. Additionally, non-intrusive icons can indicate available suggestions, allowing users to expand them as needed.

**User-Initiated Suggestions** (Collaborator). In open-ended tasks, where users seek creativity, exploration, or multiple options, they are more inclined to actively engage the AI to generate suggestions. In such cases, users may first want to observe how the AI responds before shaping their own intentions. The system, however, cannot anticipate whether the user has a fully formed idea or is looking for inspiration, and premature autonomous AI suggestions may disrupt the user's flow or prematurely steer the direction of the task. By allowing users to initiate suggestions themselves, a calmer design approach can help them retain control over the interaction, ensuring that the AI's contributions align with their creative process.

*6.2.3 Suggestion: Precision vs. Diversity.* Depending on the context, the system should either prioritize precision or embrace diversity.

**Precision-Oriented Suggestions** (Facilitator). Users need *fewer but more accurate* suggestions in efficiency-driven tasks. Given that users have limited cognitive resources to evaluate numerous options, it is essential for the AI to filter out the most relevant contextual information to provide precise predictions. For example, in our study, certain apps offered multiple in-app search suggestions, but CATIA was able to refine these suggestions by leveraging cross-app history to present more accurate search terms. Section 5.4 also suggests that such an approach is feasible by showing fine-tuned models can improve accuracy with fewer options.

**Diversity-Oriented Suggestions** (Collaborator). In exploratory or creative tasks, users seek *diverse and informative* suggestions. Precision is less critical in these contexts; instead, users value having access to a broader range of possibilities. Our study found that by leveraging *additional contextual data*, such as location or past interactions, the system was able to generate options that, while unexpected, were still acceptable and useful to users. This ability to present a diverse array of suggestions enhances collaboration by offering new possibilities, thereby supporting users in refining their ideas and engaging in more meaningful creative exploration.

*6.2.4 Interface: Single-Action vs. Conversational.* The design of the interface is key to supporting different modes of interaction. While CATIA is currently designed as a popup panel, varying tasks may demand different interface approaches.

**Single-Action Interface** (Facilitator). In the Facilitator model, users expect the AI to integrate seamlessly with existing input methods, such as the smartphone keyboard, and to apply suggestions with a single tap. They prefer minimal disruption to their workflow, with no additional learning or interaction costs. In this case, the single-action interface is ideal, as it embeds suggestions directly into the user's input field, allowing for quick, effortless integration of AI-generated text.

**Conversational Interface** (Collaborator). Users in the Collaborator model may benefit from a more interactive, conversational-style interface that fosters deeper engagement with the AI. In this setup, a dedicated panel or dialogue interface enables a back-and-forth exchange, where users can iteratively refine and regenerate suggestions. Unlike purely text-based interactions, *the reasoning behind suggestions or the AI's thought process* can be optionally displayed, as suggested by [53] and [50], helping to inspire users' intent beyond just the literal text. Moreover, users sometimes have a clear intent to express but may be constrained by mobile conditions. In such cases, the AI can proactively highlight and inquire about the missing parts of the user's intended message, as proposed by [67]. This shared interaction space facilitates a richer, more meaningful collaboration [48], as users gain insights into the AI's underlying logic, thus encouraging creative exploration and providing more control over the final output.

## 6.3 Ethical Issues

*6.3.1 Over-reliance and Cognitive Manipulation.* Although context-aware text suggestions can benefit users, they may also shift situational reasoning to the AI, potentially influencing human thoughts and decisions [53]. Moreover, AI suggestions may inherit biases from their training data [40, 60]. If the system prioritizes content from specific sources or favors certain types of queries, it could inadvertently reinforce existing biases or limit information diversity. For example, search suggestions based on a user's past purchases or interests might narrow their perspective or unintentionally favor specific companies or products. Another critical issue is the subtle integration of advertisements or promotional content into AI-generated suggestions. While users may trust the system, prioritizing sponsored content or subtly nudging users toward certain choices risks manipulating their decisions without their full awareness.

To mitigate these risks, technology providers should work to reduce inherent biases in LLMs and introduce external oversight mechanisms. At the same time, active user involvement is equally important. Systems should offer clear explanations for each suggestion, outlining its source and rationale. In high-stakes situations, such as financial decisions, or user-designated critical contexts, users should be required to pause and review these explanations. This aligns with the "seamful" design approach proposed in prior work [18, 27, 53].

*6.3.2 Data Privacy and User Control.* Collecting user context information comes at the cost of privacy, and participants in our study expressed corresponding concerns. In our study, we used a dedicated LLM API, with all data securely stored on a dedicated server. This approach ensured that sensitive information was protected. However, we acknowledge that in practical deployments, stronger privacy measures will be necessary.

A key avenue for addressing these concerns is through on-device LLM deployment, which would enable the processing of user data locally, without needing to upload sensitive contextual information to the cloud. Our findings (Section 5.4) indicate that smaller models have the potential to perform well in such scenarios, thus supporting the feasibility of this direction.

Another critical privacy issue is screen content capture. Our findings (Section 5.1.2) suggest that the information on a single screen is often sufficient for providing text suggestions. As a result, future systems could reduce the need for continuous screen content collection.

Additionally, providing users with granular control over what data is collected can empower them to make informed decisions about their privacy. Optional data collection policies could include requirements on specific sensors or types of data, as well as the duration of data collection. Users could also specify more refined control rules for data collection.

# 7 Limitations and Future Work

## 7.1 Data Collection and Contextual Factors

Our study collected one week of in-the-wild data from Chinese student participants. While the data volume and participant diversity are limited, we do not claim that our study encompasses all possible use cases or is generalizable to all populations. Nevertheless, we believe that this case study provides valuable insights for other research exploring Human-AI collaboration and the application of LLMs in everyday contexts for end users.

The contextual variables we collected were diverse but restricted to textual data, leaving room for future expansion and improvement. For instance, beyond capturing screen content through accessibility APIs, future studies could integrate image or video data, which would be particularly relevant for richer social media platforms.

## 7.2 Choice of LLMs and Prompting Methods

This study employed GPT-4 Turbo, which, while effective, is relatively costly and slower compared to some newer models (see Section 5.4). Due to the time constraints of the study, we chose this model, though it may not be the optimal choice for real-world deployment. Future implementations should consider more lightweight, faster, and cost-effective LLMs that are better suited for seamless integration into everyday use.

Additionally, we fed contextual information into the prompt in JSON format, which may not necessarily be the most optimal structure [52]. The screen content was input as a list of text fragments without specific organization. There may be more effective ways to represent this data, which could enhance the LLM's ability to reason about the contextual information and generate more accurate suggestions.

## 7.3 Future Directions

We are focusing on refining the system to better align with the proposed mental models. Future studies could involve more diverse participant groups with varied backgrounds, including different cultures, occupations, and age ranges, to enhance the generalizability of our findings. An interesting direction is exploring how non-native speakers use similar systems on mobile devices, where the need for effective human-AI collaboration may be even greater. This could offer insights into overcoming language barriers in diverse real-world settings. Additionally, long-term studies will be necessary to observe how users adapt to AI collaboration over time and how AI systems can evolve their suggestions based on sustained interaction.

# 8 Conclusion

This paper investigates human-AI collaborative text entry on smartphones using large language models (LLMs). We developed a context-aware text input assistant (CATIA), which provides text suggestions based on contextual factors such as screen content, time, location, and user activity.

In a 7-day in-the-wild study with 36 participants, we found that the system provided appropriate suggestions in over 80% of cases, primarily in two key scenarios: interpersonal communication and information services. The collaboration between users and the system demonstrated its effectiveness in reducing cognitive load. While the system mainly relied on screen-based information to infer input text, off-screen factors also proved useful in specific contexts. Additionally, an offline evaluation of various LLMs on the collected dataset showed that smaller, faster, and more cost-effective models could potentially achieve results comparable to the larger model used in our study, making them more practical for real-world applications.

We identified two distinct mental models of human-AI collaborative text entry: in efficiency-driven tasks, the AI is expected to act as a supportive facilitator, while in more complex, creative tasks, it is viewed as an equal collaborator. We also outlined design options across different dimensions to support these models.

Our work provides empirical evidence for human-AI collaborative text entry and offers insights into the design and implementation of LLM-based systems for real-world end-user applications.

# Acknowledgments

This work is supported by the National Key Research and Development Plan of China under Grant No. 2024YFB4505500 & No. 2024YFB4505502, Beijing Key Lab of Networked Multimedia, Institute for Artificial Intelligence, Tsinghua University (THUAI), Beijing National Research Center for Information Science and Technology (BNRist), 2025 Key Technological Innovation Program of Ningbo City under Grant No. 2022Z080, Beijing Municipal Science and Technology Commission, Administrative Commission of Zhongguancun Science Park No. Z221100006722018, and Science and Technology Innovation Key R&D Program of Chongqing.

# References

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2021. Context-aware Target Apps Selection and Recommendation for Enhancing Personal Mobile Assistants. *ACM Transactions on Information Systems* 39, 3 (2021), 29:1–29:30. doi:10.1145/3447678

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–13. doi:10.1145/3290605.3300233

[3] Apple. 2024. Apple Intelligence. https://www.apple.com/apple-intelligence/

[4] Timothy J. Aveni, Armando Fox, and Björn Hartmann. 2023. Bringing Context-Aware Completion Suggestions to Arbitrary Text Entry Interfaces. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 1–3. doi:10.1145/3586182.3615825

[5] Florian Bemmann, Timo Koch, Maximilian Bergmann, Clemens Stachl, Daniel Buschek, Ramona Schoedel, and Sven Mayer. 2024. Putting Language into Context Using Smartphone-Based Keyboard Logging. doi:10.48550/arXiv.2403.05180 arXiv:2403.05180 [cs].

[6] Priyanka Bhatele and Mangesh Bedekar. 2023. Survey on Smartphone Sensors and User Intent in Smartphone Usage. In *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*. 1–9. doi:10.1109/I2CT57861.2023.10126192

[7] Xiaojun Bi and Shumin Zhai. 2016. IJQwerty: What Difference Does One Key Change Make? Gesture Typing Keyboard Optimization Bounded by One Key Position Change from Qwerty. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 49–58. doi:10.1145/2858036.2858421

[8] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance,

Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the Opportunities and Risks of Foundation Models. doi:10.48550/arXiv.2108.07258 arXiv:2108.07258 [cs].

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[10] Runze Cai, Nuwan Janaka, Yang Chen, Lucia Wang, Shengdong Zhao, and Can Liu. 2024. PANDALens: Towards AI-Assisted In-Context Writing on OHMD During Travels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–24. doi:10.1145/3613904.3642320

[11] Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *Comput. Surveys* 44, 1 (2012), 1:1–1:50. doi:10.1145/2071389.2071390

[12] Juan Pablo Carrascal and Karen Church. 2015. An In-Situ Study of Mobile App & Mobile Search Interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2739–2748. doi:10.1145/2702123.2702486

[13] Weihao Chen, Chun Yu, Huadong Wang, Zheng Wang, Lichen Yang, Yukun Wang, Weinan Shi, and Yuanchun Shi. 2023. From Gap to Synergy: Enhancing Contextual Understanding through Human-Machine Collaboration in Personalized Systems. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3586183.3606741

[14] Chenhui Cui, Tao Li, Junjie Wang, Chunyang Chen, Dave Towey, and Rubing Huang. 2024. Large Language Models for Mobile GUI Text Input Generation: An Empirical Study. doi:10.48550/arXiv.2404.08948 arXiv:2404.08948 [cs].

[15] Anind K. Dey. 2001. Understanding and Using Context. *Personal and Ubiquitous Computing* 5, 1 (Feb. 2001), 4–7. doi:10.1007/s007790170019 Publisher: Springer.

[16] Trinh Minh Tri Do, Jan Blom, and Daniel Gatica-Perez. 2011. Smartphone usage in the wild: a large-scale analysis of applications and context. In *Proceedings of the 13th international conference on multimodal interfaces (ICMI '11)*. Association for Computing Machinery, New York, NY, USA, 353–360. doi:10.1145/2070481.2070550

[17] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A Survey on In-context Learning. doi:10.48550/arXiv.2301.00234 arXiv:2301.00234 [cs].

[18] Upol Ehsan, Q. Vera Liao, Samir Passi, Mark O. Riedl, and Hal Daumé. 2024. Seamful XAI: Operationalizing Seamful Design in Explainable AI. *Proc. ACM Hum.-Comput. Interact.* 8, CSCW1 (2024), 119:1–119:29. doi:10.1145/3637396

[19] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of Language Modeling and its Personalization on Touchscreen Typing Performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 649–658. doi:10.1145/2702123.2702503

[20] Yue Fu, Sami Foell, Xuhai Xu, and Alexis Hiniker. 2024. From Text to Self: Users' Perception of AIMC Tools on Interpersonal Communication and Self. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3613904.3641955

[21] Nestor Garay-Vitoria and Julio Abascal. 2006. Text prediction systems: a survey. *Universal Access in the Information Society* 4, 3 (March 2006), 188–203. doi:10.1007/s10209-005-0005-9

[22] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. doi:10.48550/arXiv.2406.12793 arXiv:2406.12793.

[23] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N. Patel, and Jacob O. Wobbrock. 2013. ContextType: using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2795–2798. doi:10.1145/2470654.2481386

[24] Joshua T. Goodman. 2001. A bit of progress in language modeling. *Computer Speech & Language* 15, 4 (Oct. 2001), 403–434. doi:10.1006/csla.2001.0174

[25] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents. In *International Conference on Machine Learning*. PMLR, 9118–9147. https://proceedings.mlr.press/v162/huang22a.html ISSN: 2640-3498.

[26] 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 (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–19. doi:10.1145/3544548.3580688

[27] Sarah Inman and David Ribes. 2019. "Beautiful Seams": Strategic Revelations and Concealments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3290605.3300508

[28] Maryam Kamvar and Shumeet Baluja. 2007. The role of context in query input: using contextual signals to complete queries on mobile devices. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services (MobileHCI '07)*. Association for Computing Machinery, New York, NY, USA, 405–412. doi:10.1145/1377999.1378046

[29] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. doi:10.48550/arXiv.2001.08361 arXiv:2001.08361 [cs].

[30] Per Ola Kristensson. 2009. Five Challenges for Intelligent Text Entry Methods. *AI Magazine* 30, 4 (Sept. 2009), 85–85. doi:10.1609/aimag.v30i4.2269 Number: 4.

[31] Per Ola Kristensson, Stephen Brewster, James Clawson, Mark Dunlop, Leah Findlater, Poika Isokoski, Benoît Martin, Antti Oulasvirta, Keith Vertanen, and Annalu Waller. 2013. Grand challenges in text entry. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. Association for Computing Machinery, New York, NY, USA, 3315–3318. doi:10.1145/2468356.2479675

[32] Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics-Doklady* 10 (1966), 707–710. https://api.semanticscholar.org/CorpusID:60827152

[33] Toby Jia-Jun Li and Brad A. Myers. 2021. A Need-finding Study for Understanding Text Entry in Smartphone App Usage. doi:10.48550/arXiv.2105.10127 arXiv:2105.10127 [cs].

[34] Q. Vera Liao, Daniel Gruen, and Sarah Miller. 2020. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3313831.3376590

[35] Q. Vera Liao and Jennifer Wortman Vaughan. 2023. AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap. doi:10.48550/arXiv.2306.01941 arXiv:2306.01941.

[36] Zhe Liu, Chunyang Chen, Junjie Wang, Xing Che, Yuekai Huang, Jun Hu, and Qing Wang. 2023. Fill in the Blank: Context-Aware Automated Text Input Generation for Mobile GUI Testing. In *Proceedings of the 45th International Conference on Software Engineering (ICSE '23)*. IEEE Press, Melbourne, Victoria, Australia, 1355–1367. doi:10.1109/ICSE48619.2023.00119

[37] 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 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–20. doi:10.1145/3613904.3642939

[38] I. Scott MacKenzie and R. William Soukoreff. 2002. Text Entry for Mobile Computing: Models and Methods,Theory and Practice. *Human–Computer Interaction* 17, 2-3 (Sept. 2002), 147–198. doi:10.1080/07370024.2002.9667313 Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/07370024.2002.9667313.

[39] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented Language Models: a Survey. (Feb. 2023). http://arxiv.org/abs/2302.07842 arXiv:2302.07842.

[40] Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 5356–5371. doi:10.18653/v1/2021.acl-long.416

[41] OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt

[42] OpenAI. 2024. OpenAI Models. https://platform.openai.com/docs/models

[43] Antti Oulasvirta, Anna Reichel, Wenbin Li, Yan Zhang, Myroslav Bachynskyi, Keith Vertanen, and Per Ola Kristensson. 2013. Improving two-thumb text entry on touchscreen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2765–2774. doi:10.1145/2470654.2481383

[44] Antti Oulasvirta, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. 2005. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. Association for Computing Machinery, New York, NY, USA, 919–928. doi:10.1145/1054972.1055101

[45] Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014. An in-situ study of mobile phone notifications. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI '14)*. Association for Computing Machinery, New York, NY, USA, 233–242. doi:10.1145/2628363.2628364

[46] Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-rong Wen. 2025. Tool learning with large language models: a survey. *Frontiers of Computer Science* 19, 8 (Jan. 2025), 198343. doi:10.1007/s11704-024-40678-2

[47] Philip Quinn and Shumin Zhai. 2016. A Cost-Benefit Study of Text Entry Suggestion Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 83–88. doi:10.1145/2858036.2858305

[48] Jeba Rezwana and Mary Lou Maher. 2023. Designing Creative AI Partners with COFI: A Framework for Modeling Interaction in Human-AI Co-Creative Systems. *ACM Trans. Comput.-Hum. Interact.* 30, 5 (2023), 67:1–67:28. doi:10.1145/3519026

[49] Tapio Soikkeli, Juuso Karikoski, and Heikki Hammainen. 2011. Diversity and End User Context in Smartphone Usage Sessions. In *2011 Fifth International Conference on Next Generation Mobile Applications, Services and Technologies*. 7–12. doi:10.1109/NGMAST.2011.12 ISSN: 2161-2897.

[50] Hari Subramonyam, Roy Pea, Christopher Pondoc, Maneesh Agrawala, and Colleen Seifert. 2024. Bridging the Gulf of Envisioning: Cognitive Challenges in Prompt Based Interactions with LLMs. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–19. doi:10.1145/3613904.3642754

[51] Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. 2023. Cognitive Architectures for Language Agents. doi:10.48550/arXiv.2309.02427 arXiv:2309.02427 [cs].

[52] Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models. doi:10.48550/arXiv.2408.02442 arXiv:2408.02442 [cs].

[53] Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The Metacognitive Demands and Opportunities of Generative AI. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–24. doi:10.1145/3613904.3642902

[54] Jaime Teevan, Amy Karlson, Shahriyar Amini, A. J. Bernheim Brush, and John Krumm. 2011. Understanding the importance of location, time, and people in mobile local search behavior. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. Association for Computing Machinery, New York, NY, USA, 77–80. doi:10.1145/2037373.2037386

[55] Michael Terry, Chinmay Kulkarni, Martin Wattenberg, Lucas Dixon, and Meredith Ringel Morris. 2024. Interactive AI Alignment: Specification, Process, and Evaluation Alignment. doi:10.48550/arXiv.2311.00710 arXiv:2311.00710.

[56] Shiu Lun Tsang and Siobhan Clarke. 2007. Mining User Models for Effective Adaptation of Context-Aware Applications. In *The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*. 178–187. doi:10.1109/IPC.2007.108

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[58] Keith Vertanen, Mark Dunlop, James Clawson, Per Ola Kristensson, and Ahmed Sabbir Arif. 2016. Inviscid Text Entry and Beyond. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 3469–3476. doi:10.1145/2851581.2856472

[59] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating Fast Mobile Text Entry using Sentence-Based Decoding of Touchscreen Keyboard Input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 659–668. doi:10.1145/2702123.2702135

[60] Yixin Wan, George Pu, Jiao Sun, Aparna Garimella, Kai-Wei Chang, and Nanyun Peng. 2023. "Kelly is a Warm Person, Joseph is a Role Model": Gender Biases in LLM-Generated Reference Letters. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 3730–3748. doi:10.18653/v1/2023.findings-emnlp.243

[61] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. doi:10.48550/arXiv.2201.11903 arXiv:2201.11903 [cs].

[62] Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain text entry on mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2307–2316. doi:10.1145/2556288.2557412

[63] Liang Xu, Anqi Li, Lei Zhu, Hang Xue, Changtai Zhu, Kangkang Zhao, Haonan He, Xuanwei Zhang, Qiyue Kang, and Zhenzhong Lan. 2023. SuperCLUE: A Comprehensive Chinese Large Language Model Benchmark. https://arxiv.org/abs/2307.15020v1

[64] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. doi:10.48550/arXiv.2407.10671 arXiv:2407.10671.

[65] Shumin Zhai, Michael Hunter, and Barton A. Smith. 2002. Performance Optimization of Virtual Keyboards. *Human–Computer Interaction* 17, 2-3 (Sept. 2002), 229–269. doi:10.1080/07370024.2002.9667315 Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/07370024.2002.9667315.

[66] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. doi:10.48550/arXiv.2303.18223 arXiv:2303.18223 [cs].

[67] Chen Zhou, Zihan Yan, Ashwin Ram, Yue Gu, Yan Xiang, Can Liu, Yun Huang, Wei Tsang Ooi, and Shengdong Zhao. 2024. GlassMail: Towards Personalised Wearable Assistant for On-the-Go Email Creation on Smart Glasses. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference (DIS '24)*. Association for Computing Machinery, New York, NY, USA, 372–390. doi:10.1145/3643834.3660683

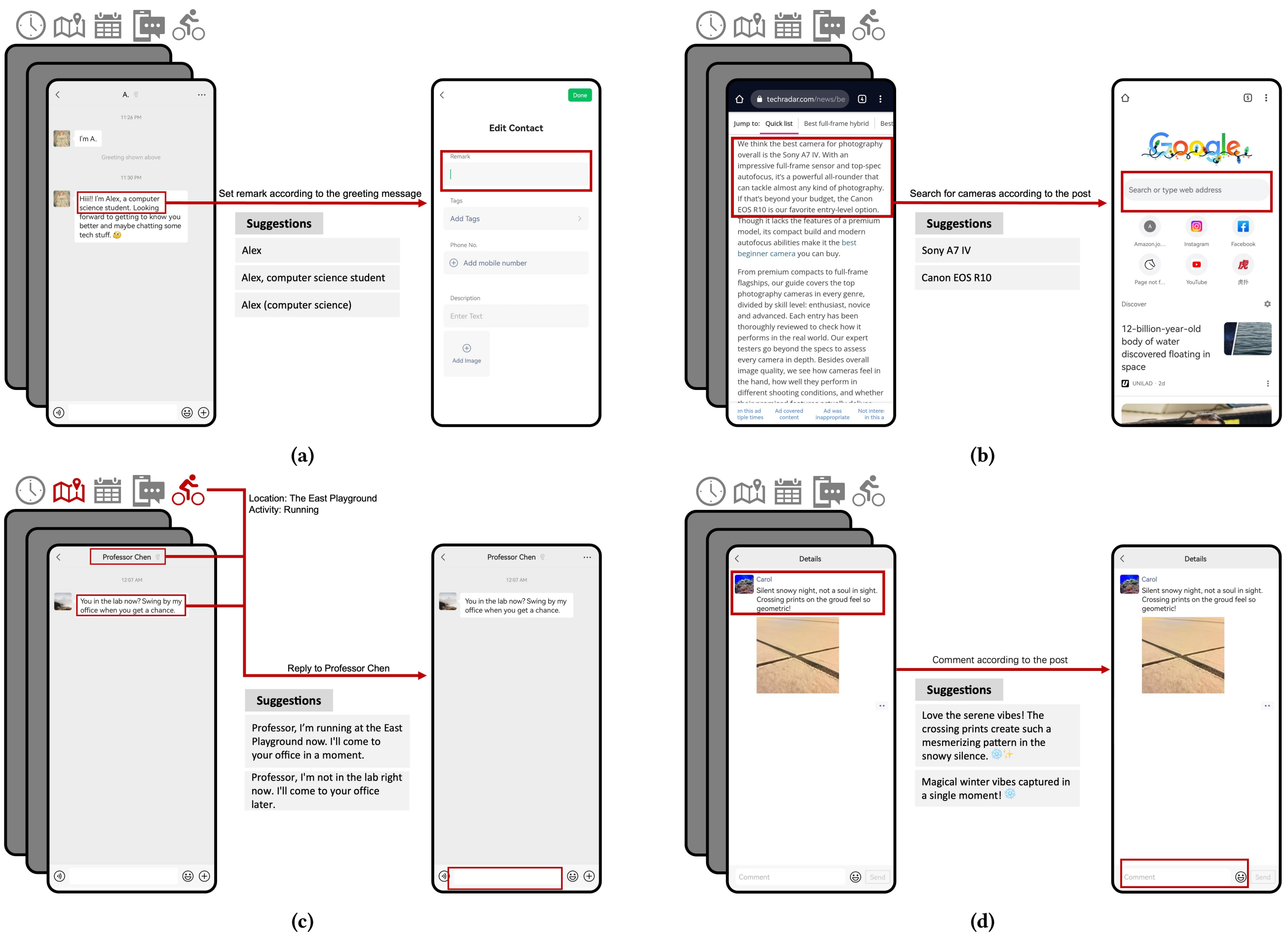
# A Example Use Cases

We present several use cases of CATIA in Figure 6. The marked content in red represents the key information for text suggestion.

# B Implementation Details

We present the system architecture diagram of CATIA in Figure 7. The system's Android mobile app communicates with the remote server via Socket.IO. When the user activates the assistant, the Android app establishes communication with the server, collects and sends contextual information, and waits for the server to execute the suggestion computation results. The suggestion panel on the Android side displays the results in real time.

For the collection of contextual information on the phone, recent screen content is continuously updated in the background. The app always maintains a queue of pages from the last two minutes, which is managed by a screen stability algorithm that decides whether to collect screen text and add it to the queue. This screen stability algorithm takes screenshots at regular intervals of 200 ms and detects the similarity between adjacent screenshot images at the pixel level. When the similarity is below 0.8, the screen is considered to be changing. Once the similarity exceeds 0.8 and remains stable for 400 ms, the screen is considered to have entered a new stable state, and the text on the screen at that moment is collected through the accessibility service and placed in the queue. In addition, if the



Hiii!! I'm Alex, a computer science student. Looking forward to getting to know you better and maybe chatting some tech stuff. 😉

Set remark according to the greeting message

Suggestions

- Alex
- Alex, computer science student
- Alex (computer science)

We think the best camera for photography overall is the Sony A7 IV. With an impressive full-frame sensor and top-spec autofocus, it's a powerful all-rounder that can tackle almost any kind of photography. If that's beyond your budget, the Canon EOS R10 is our favorite entry-level option. Though it lacks the features of a premium model, its compact build and modern autofocus abilities make it the best beginner camera you can buy.

From premium compacts to full-frame flagships, our guide covers the top photography cameras in every genre, divided by skill level: enthusiast, novice and advanced. Each entry has been thoroughly reviewed to check how it performs in the real world. Our expert testers go beyond the specs to assess every camera in depth. Besides overall image quality, we see how cameras feel in the hand, how well they perform in different shooting conditions, and whether

12-billion-year-old body of water discovered floating in space

Search for cameras according to the post

Suggestions

- Sony A7 IV
- Canon EOS R10

Location: The East Playground
Activity: Running

You in the lab now? Swing by my office when you get a chance.

Reply to Professor Chen

Suggestions

- Professor, I'm running at the East Playground now. I'll come to your office in a moment.
- Professor, I'm not in the lab right now. I'll come to your office later.

Silent snowy night, not a soul in sight. Crossing prints on the groud feel so geometric!

Comment according to the post

Suggestions

- Love the serene vibes! The crossing prints create such a mesmerizing pattern in the snowy silence. ❄️✨
- Magical winter vibes captured in a single moment! ❄️

**Figure 6: Four use cases of CATIA. Marked content in red represents the key information for text suggestion. (a) The user added a new friend on a social media platform, and the friend sent a greeting message; the assistant suggested suitable remarks. (b) The user came across a tweet about camera shopping and wanted to search for cameras on Google; the assistant suggested suitable keywords. (c) The user was running on the playground when the professor sent a message asking if the user was in the lab and requested him to visit their office when available; based on the physical context (location and activity) and the message content, the assistant suggested suitable responses. (d) The user was browsing a friend's post; the assistant suggested suitable comments.**

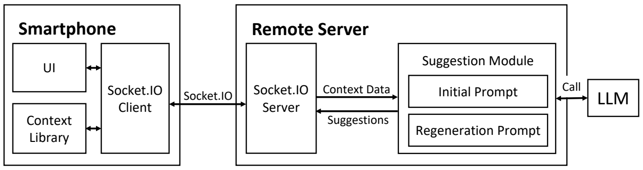


**Figure 7: System architecture diagram of CATIA.**

text on adjacent pages in the queue has a similarity greater than 0.75, they will be merged into a single page.

We use `gpt-4-1106-preview` from OpenAI's Chat Completions API as the driving LLM. The API was configured with the following parameters: `top_p` set to 1.0, `max_tokens` limited to 512, and the response format specified as a JSON object.

## C Prompts

Our prompts follow a dialog format, where the "system" part presents the core requirements of each task, and the first "user" part provides a detailed instruction. Following parts contain several input-output examples. Here we only present the prompts without their input-output examples. For complete prompts used in the paper, please refer to the supplementary material.

### C.1 Initial Suggestion

```
system
You are a text suggestion assistant.

user
```

~~~~
[context] information is automatically sent following a user's request for text suggestions in an input field. This includes temporal, physical, social, and other digital information collected on the phone (such as context.date_time, context.location, context.screen_content, etc.).
Among these, context.screen_content is a list of screen pages captured in chronological order. If the type attribute of a screen_content page is 'chat', it represents a chat history; if it is 'screen', it contains text snippets extracted from that particular screen.
Your task is to deduce the user's possible intents for initiating text input and suggest appropriate texts to the user. This involves analyzing provided context and the active input field:
```
input_field.app: (the APP that the field is in),
input_field.label: (the label of the field),
input_field.content: (existing user input, you can infer the attitude the user is trying to convey based on this entered content)
```
You need to follow these steps:
1. From [context], filter out which elements are pages the user actually want to browse. Then based on the [input_field], further filter out which elements may be relevant to the user's input action.
2. Based on the filtered [context] from step 1 and [input_field], analyze the [intention] (why user opens the input field and what the user wants to express?) of the user. Remember, since elements in context.screen_content are the pages user browsed over a certain period of time, so the elements (context.screen_content[index]) with a smaller index may not be relevant or useful, and you should pay more attention to elements (context.screen_content[index]) with a greater index. If there are multiple possible intentions, list them in the keys of [intention_suggestion_pair]. Different possible attitudes of users under the same topic can also be counted as multiple intentions.
3. Give [suggestion] to the user based on each [intention] and list them in the values of [intention_suggestion_pair].
You need to output [output.intention_suggestion_pair] in JSON format (up to 4). Preferentially output the intention_suggestion_pair that is more likely in the given contexts.
~~~~

## C.2 Suggestion Regeneration

**system**

~~~~
You are a text suggestion assistant and regenerate suggestions based on history results and user's instruction.
~~~~

**user**

~~~~
[context] information is automatically sent following a user's request for text suggestions in an input field. This includes temporal, physical, social, and other digital information collected on the phone (such as context.date_time, context.location, context.screen_content, etc.).
Among these, context.screen_content is a list of screen pages captured in chronological order. If the type attribute of a screen_content page is 'chat', it represents a chat history; if it is 'screen', it contains text snippets extracted from that particular screen.
Your task is to deduce the user's possible intents for initiating text input and suggest appropriate texts to the user. This involves analyzing provided context and the active input field:
```
input_field.app: (the APP that the field is in),
input_field.label: (the label of the field),
input_field.content: (existing user input, you can infer the attitude the user is trying to convey based on this entered content)
```
You will also be provided:
- last_output: A dict of texts generated last time and the guessed user intention.
- user_instruction: User demand for generated text.

You need to follow these steps:
1. Based on [context], [input_field], [last_output] and [user_instruction], analyze the [intention] (why user opens the input field and what the user wants to express?) of the user. If there are multiple possible intentions, list them in the keys of [intention_suggestion_pair]. Different possible attitudes of users under the same topic can also be counted as multiple intentions.
2. Give [suggestion] to the user based on each [intention] and list them in the values of [intention_suggestion_pair].
You need to output [output.intention_suggestion_pair] in JSON format (up to 4). Preferentially output the intention_suggestion_pair that is more likely in the given contexts.
~~~~

## C.3 Data Analysis

**system**

~~~~
You are a text suggestion assistant. Your role is to analyze a user's smartphone text entry intention and identify key information crucial for inferring the text a user has input. This analysis occurs post-text entry.
~~~~

**user**

~~~~
You need to consider the contextual data captured by the device and the groundtruth text entered by the user.
~~~~

```
[context] information is captured when a user opens an input
      field. This includes temporal, physical, social, and
     other digital information collected on the phone (such
     as context.date_time, context.location, context.
     screen_content, etc.).
Among these, context.screen_content is a list of screen
     pages captured in chronological order. If the type
     attribute of a screen_content page is 'chat', it
     represents a chat history; if it is 'screen', it
     contains text snippets extracted from that particular
     screen.
[input_field] is the target input field, which contains the
     following attributes:
- input_field.app: the APP that the field is in,
- input_field.label: a label providing a description of the
     field's function or purpose,
- input_field.content: existing user input.
[entered_text] is the final text entered by the user in the
     current context.

You need to follow these steps:
1. Analyze the [intention] (why the user opens the input
     field and what the user wants to express?) of the user
     based on [context], [input_field] and [entered_text].
2. Identify the [key_information] (elements in [context] and
      [input_field], not in [entered_text]) that is crucial
     to infer the [entered_text]. Focus on the most relevant
      details that directly influence the user's text entry
     process. If the user uses only part of context.
     screen_content, you only need to output the part of the
      content that is used.
You need to output [output.intention], [output.
     key_information] in JSON format.
```