Supporting Text Entry in Virtual Reality with Large Language Models

Conference Paper · March 2024

DOI: 10.1109/VR58804.2024.00073

7 authors, including:


Liuqing Chen
Zhejiang University


Yu Cai
Zhejiang University


Preben Hansen
Stockholm University

2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR) | 979-8-3503-7402-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/VR58804.2024.00073

Supporting Text Entry in Virtual Reality with Large Language Models

Liuqing Chen* Zhejiang University Zhejiang-Singapore Innovation and AI Joint Research Lab

Yu Cai† Zhejiang University

Ruyue Wang‡ Zhejiang University

Shixian Ding§ Zhejiang University

Yilin Tang¶ Zhejiang University

Preben Hansen‖ Stockholm University

Lingyun Sun** Zhejiang University Zhejiang-Singapore Innovation and AI Joint Research Lab

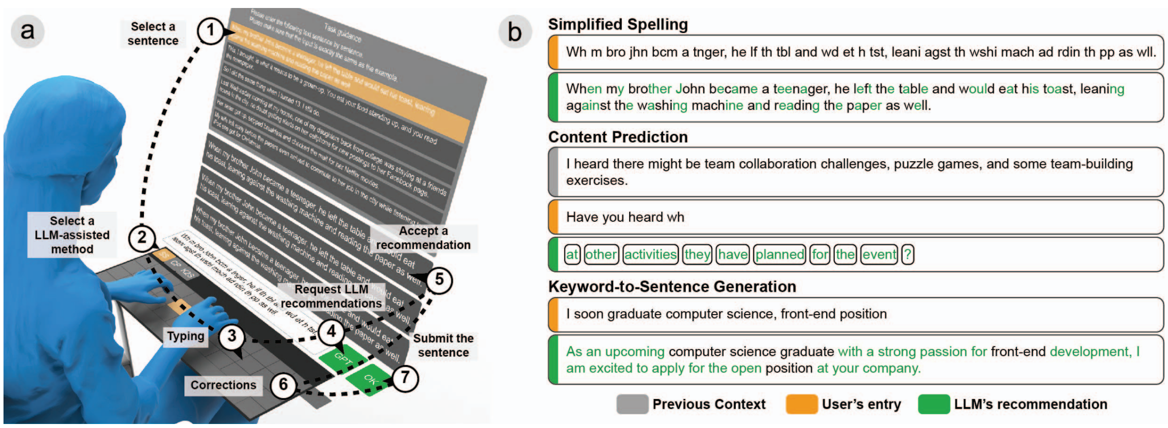

Figure 1: (a) The layout of the interfaces. Users can employ the index fingers of both hands to directly interact with all the buttons on the interface. Ideally, users follow the numerical order in the figure to complete text entry tasks. (b) Examples of the three LLM-assisted typing methods. These methods include predicting words with incomplete spelling (Simplified Spelling), continuing unfinished sentences (Content Prediction), and stringing together relevant keywords (Keyword-to-Sentence Generation). The LLM generates recommendations based on the user's incomplete entries and previous context.

Abstract

Text entry in virtual reality (VR) often faces challenges in terms of efficiency and task loads. Prior research has explored various solutions, including specialized keyboard layouts, tracked physical devices, and hands-free interaction. Yet, these efforts often fall short of replicating the efficiency of real-world text entry, or introduce additional spatial and device constraints. This study leverages the extensive capabilities of large language models (LLMs) in context perception and text prediction to enhance text entry efficiency by reducing users' manual keystrokes. Three LLM-assisted text entry methods - Simplified Spelling, Content Prediction, and Keyword-to-Sentence Generation - are introduced, aligning with user cognition and the contextual predictability of English text at word, grammatical structure, and sentence levels. Through user experiments encompassing various text entry tasks on an Oculus-based VR prototype, these methods demonstrate a 16.4%, 49.9%, 43.7% reduction in manual keystrokes, translating to efficiency gains of 21.4%,74.0%, 76.3%, respectively. Importantly, these methods do not increase manual corrections compared to manual typing, while significantly reducing physical, mental, and temporal loads and enhancing overall usability. Long-term observations further reveal users' strategies for using these LLM-assisted methods, showing that users' proficiency with the methods can reinforce their positive effects on text entry efficiency.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual Reality; Human-centered computing—Human computer interaction (HCI)—Interaction techniques—Text input

1 Introduction

Text entry stands as a pivotal interaction activity in virtual reality (VR), finding wide-ranging applications in social and working scenarios [13]. In the VR landscape, the dominant method for text entry typically involves using controllers or free hands to select characters from a virtual keyboard [60]. While these methods offer advantages in spatial and equipment constraints [2], they are plagued by inefficiencies that impact user experience, task loads, and productivity [13, 48]. Numerous research efforts have explored avenues to enhance text entry efficiency in VR, proposing optimized keyboard designs [29, 63, 72, 75], body metaphors [4, 6], and hands-free interaction methods [31, 32]. However, these enhancements remain far from achieving the efficiency of text input in the physical world [4, 6, 29, 31, 32, 75]. Some studies have introduced additional physical devices [7, 19, 24, 25, 27, 37, 44–46, 54] into VR systems to further improve text input efficiency, which usually compromise the flexibility and accessibility of VR environments. Considering these challenges and trade-offs, there is a pressing need for an assistive text input approach in VR that seamlessly integrates with existing text input techniques, significantly enhances text entry efficiency, and

*e-mail: chenlq@zju.edu.cn
†e-mail: cai_yu@zju.edu.cn
‡e-mail: wang_ry@zju.edu.cn
§e-mail: shixian.ding@outlook.com
¶e-mail: 22251376@zju.edu.cn
‖e-mail: preben@dsv.su.se
**e-mail: sunly@zju.edu.cn

979-8-3503-7402-5/24/$31.00 ©2024 IEEE
DOI 10.1109/VR58804.2024.00073

improves the overall user experience while preserving the flexibility of VR environments.

In response to these challenges, recent advances in large language models (LLMs) provide new opportunities. Trained on vast corpora of text, LLMs have demonstrated exceptional proficiency in understanding and generating human languages [8, 43, 51, 52], and are capable of capturing intricate linguistic patterns, context-based associations, and semantic nuances [30]. Leveraging these LLM features, previous studies have explored their applications in word correction [76], generating specific types of text [1], and facilitating high-quality textual communication for individuals with limited textual prompts, particularly those with disabilities [66]. In the context of VR text entry, LLMs exhibit the potential to significantly enhance efficiency and user experience by offering context-based text completion and predictions, thereby reducing the effort for manual input. Such predictive assistance could be particularly beneficial for text entry in VR compared to non-VR conditions, where cumbersome keyboard interactions and the absence of feedback severely challenge the alignment between users' thinking and typing process. Furthermore, LLM assistance can be provided in a logical and intuitive manner, closely aligning with human cognition during text input.

In this study, we introduce a novel LLM-assisted text input approach in VR to enhance the efficiency and user experience of text entry by reducing required manual inputs. Our approach leverages the predictive capabilities of LLMs across multiple linguistic levels, including individual words, grammatical structures, and sentences. Within this framework, we introduce three distinct LLM-assisted text input methods: Simplified Spelling, Content Prediction, and Keyword-to-Sentence Generation. To realize this concept, we have developed a functional VR prototype of our LLM-assisted text entry approach by integrating the GPT-3.5 model. We conducted comprehensive assessments across diverse text entry scenarios, which demonstrated substantial improvements in typing efficiency and reduced task loads while maintaining error rates comparable to those observed during manual input. Our study also unveils participants' preferences for these methods and how their strategies learned in longer-term practice enhance their effectiveness.

The main contributions of this study can be summarized as:

1. Introducing LLM as an assistive approach for text entry tasks in VR, and proposing three specific LLM-assisted methods to reduce users' manual keystrokes at the level of word, grammatical structure, and sentence, respectively.

2. Developing a prototype and demonstrating the effects of the LLM-assisted methods on text entry in three common long text entry scenarios, including text transcription, dialogue, and email writing.

3. Investigating user preferences and strategies when employing these LLM-assisted methods in various scenarios, as well as evaluating the impact of users' long-term proficiency with the LLM-assisted methods on text entry performance.

2 Related work

2.1 Text entry methods in VR

Significant research endeavors have been dedicated to enhancing text input efficiency in VR by optimizing interaction techniques. One prominent approach to address this challenge involves the reconfiguration of virtual keyboard layouts. Researchers have introduced a spectrum of inventive keyboard designs, such as split keyboards [69], cubic keyboards [70], and circular keyboards [29, 40, 75]. These redesigned keyboard interfaces have exhibited tangible improvements in text entry speed and have introduced distinct features tailored to specific application contexts. Some researchers have also explored the factors influencing the redesign of the virtual keyboards. According to Yildirim et al. [72], a flat user interface (UI) led to higher efficiency and fewer corrections than a curved UI. Tominaga et al. [63] discussed the effect of the position and angle of the virtual keyboard on typing efficiency and fatigue. Keyboard visibility also has an impact on VR text entry efficiency, where a minimalist representation of the user's fingertips may enhance keyboard visibility [16]. While these alterations have yielded notable improvements, they frequently fall short of achieving text entry efficiency on par with real-world conditions and require users to make extra efforts to acclimate themselves to these novel keyboard layouts.

Another widely explored approach to enhance text entry efficiency in VR involves utilizing additional physical devices, including physical keyboards [25, 37, 44, 45, 54], touchpads [28, 74], smartphones [7, 24, 46], and controllers with special forms [20]. Wearable devices have also been proposed to compensate for accessibility, offering options like body-wearable keyboards [27] or enabling users to tap on any flat surfaces to represent keystrokes [19]. Although these approaches have demonstrated certain improvements in the efficiency of text entry in VR, they often introduce additional devices or spatial dependencies to VR systems, which can diminish the accessibility and flexibility of virtual text entry environments.

Some studies have delved into the task loads and accessibility of VR text entry techniques. To address these issues, various hands-free interaction methods have been explored. Gaze-based [34, 53] and head-motion-based [31, 73] techniques are the most common approaches, where the user's eyes or head movements are continuously tracked by the VR device to indicate their text input intentions. Lu et al. [32] compared three typical hands-free text input methods, including blink-based, head-motion-based, and dwell-based approaches, and found that the blink-based method hold the best performance. There have also been experiments on text entry using brain-computer interfaces [33] or speech recognition [49], although practical applications for these methods remain relatively limited. While hands-free methods generally offer lower cognitive and physical loads and higher accuracy, their input efficiencies are often similar or only slightly improved over the typical VR text entry techniques, including controller selection or free-hand typing [5, 15].

2.2 Predictability in the English language and AI-assisted textual communication

Linguistic research has illuminated various aspects of English text predictability, encompassing lexical cohesion, contextual dependencies, grammatical structures, and collocations [12, 35, 36]. These predictabilities have been harnessed in natural language processing (NLP) techniques to aid tasks such as word prediction [71], autocompletion [9], spell-checking [47], abbreviation expansion [65], and predictive text messaging [42]. In the VR context, Sengupta et al. [55] implemented a gaze-based text entry system that computed probabilities for incomplete words with statistical, syntactic, and semantic predictive algorithms. However, these techniques have been constrained by their limited capabilities in context awareness and text prediction. They tend to provide assistance with fixed patterns at the word or short phrase level, falling short of significantly reducing users' manual input on a broader scale.

On the other hand, LLMs, benefiting from their extensive training data and advanced architecture, excel in generating high-quality context-aware content. The attention mechanisms enable them to assess the significance of individual words or letters within a sentence in relation to the broader context, resulting in more contextually appropriate outputs [51, 67, 77]. These advances empower LLMs to deliver high-quality text predictions based on immediate, document-level, and user-specific context, thus enabling them to take over a portion of the manual text entries [11]. Based on these advantages, LLMs have been applied to assist users in textual communication, including abbreviation expansion [10], text input with Chinese pinyin [62], and form filling for websites [3]. Additionally, LLMs have found applications in augmentative and alternative communication (AAC), utilizing conversation history, user profiles, and reduced user input with predefined options to generate daily communications for individuals with disabilities [57, 66]. Certain prompt engineering tools have been preliminarily suggested for text input assistance across various use cases, though without a specific emphasis on VR or input efficiency concerns [56]. LLM's ability to predict English text with minimal context cues makes them suitable for VR text input by reducing manual input requirements.

3 Proposed Method

We propose three distinct LLM-based methods to enhance VR text entry in Sections 3.1 to 3.3. These methods include Simplified Spelling for incomplete words, Content Prediction for sentence continuations, and Keyword-to-Sentence Generation for creating sentences or paragraphs using user-provided keywords. These methods, collectively referred to as the LLM-assisted approach, offer users different levels of control and efficiency improvements during text entry.

3.1 Simplified spelling

The Simplified Spelling method streamlines user input by permitting the frequent use of incomplete spellings to represent specific words. While this concept has been explored in prior studies related to spell-checking [50] and abbreviation expansion [58], these efforts have often focused on correcting only a limited number of characters within a vast array of contexts or predicting highly specific linguistic components like phrases and fixed collocations. Leveraging the capabilities of LLM in context comprehension and text prediction, we take this strategy to the next level. The LLM-assisted method empowers users to employ incomplete spellings extensively throughout their text input, resulting in highly accurate predictions for the complete state of nearly all words within a sentence. Remarkably, this accomplishment relies solely on the context within the sentence itself. Additionally, this feature aligns with users' cognitive tendencies during typing, which often involve recalling spellings through word pronunciation [41]. Consequently, it has the potential to significantly reduce the cognitive load associated with spelling.

As depicted in Fig. 1 (b), when employing this method, users input text with incomplete spellings to the LLM, which returns the text in its fully completed state. Users can then transfer the completed text to the input box and make any necessary adjustments. Users are required to enter a complete sentence or, at the very least, a sufficiently long clause into the input box to provide adequate intra-sentence context for the LLM. Furthermore, users are encouraged to adopt specific strategies for simplifying spellings, such as retaining consonants, initials, or both initials and final letters of each word, to optimize prediction accuracy [10].

3.2 Content Prediction

The Content Prediction method offers users an accurate preview of upcoming text content by leveraging sufficient previous context. This ability is inherent in LLMs, as they generate expected outputs by extending the word sequence of input text without the need for additional training [77]. Consequently, users can opt to select words suggested by the LLM rather than manually inputting them, particularly when dealing with sentences with distinctive grammatical structures and syntax. As demonstrated in Fig. 1 (b), this method involves supplying the LLM with the current sentence being typed, along with the two preceding sentences as the previous context. This provides the LLM with sufficient cues to predict subsequent content accurately. For this method only, the recommended text is segmented into words, allowing users to selectively accept a few words that align with their expectations and effortlessly integrate them into their input box.

As the predicted text becomes longer, situations may arise where new concepts surpass the predictive capacity of the prior context, leading to heightened uncertainty. Hence, with this method, users are encouraged to interact with the LLM more frequently and continue manual input after accepting several LLM-recommended words. This iterative approach ensures accurate predictions while retaining users' control over text input. It also aligns with users' understanding that grammatical structures and lexical cohesion are contextually interdependent.

3.3 Keyword-to-Sentence generation

The Keyword-to-Sentence Generation method allows users to input solely keywords, leaving the task of generating complete sentences to the LLM. This method relies on the recognition that sentences typically involve predictable interactions between keywords. It also aligns with psycho-linguistic theories emphasizing two stages in sentence production: conceptualization (selecting key concepts) and formulation (constructing sentences) [21].

In this method (see Fig. 1 (b)), users input keywords such as roles, places, objects, and time, while using commas and logical words to indicate clauses and connections. The LLM analyzes these keywords' interactions, skillfully stringing them into complete sentences. Users can manually refine the LLM-generated text or adjust keywords for regeneration. This method offers a wider range of sentence possibilities compared to the other two methods while enhancing input efficiency through keyword-only entry.

However, it is important to note that a single LLM-assisted method may not be universally suitable for practical text entry scenarios. The specific features of the sentence, the broader context, and the intended usage of the text can significantly affect the optimal choice of the LLM-assisted methods. Thus, users are free to select the most suitable method for completing each sentence during our experiments. The users' choices of LLM-assisted methods also emerged as a pivotal factor in our observations, highlighting the adaptability and user-centered nature of our approach.

4 Implementation

To evaluate the efficacy of our proposed methods, we created a functional prototype running on Oculus Quest 2 with Unity. The prototype incorporates hand-tracking capabilities using the XR Interaction Toolkit 2.3.2 and the XRHands package offered by Unity. This enables users to engage with the virtual interfaces by means of direct touch with virtual hands.

4.1 Interfaces and apparatus

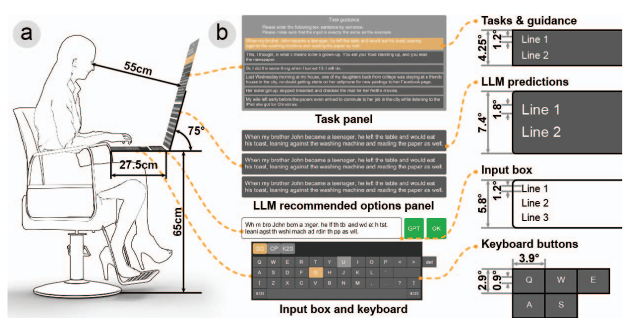

Figure 2: (a) The layout of the task panel, LLM-recommended options, and virtual keyboard in a 3D environment. (b) The detailed layout of the interface, where the size of the text and buttons are labeled according to the field of view (FOV). The task panel contains several text boxes, each containing a sentence to be entered. The box with the sentence the user is typing is marked yellow.

As shown in Fig. 2 (a), the virtual interface used for the prototype consists of three components: the task panel, the LLM-recommended options, and the virtual keyboard. The layout of the interfaces was designed to support a seated posture, which is the usual setting for extensive text entry experiments in previous studies [17, 22], with physical loads and similarity with real working scenarios as the main consideration. The virtual keyboard is placed horizontally at a fixed height of 65 cm above the ground, according to local ergonomic data [61]. The task panels and LLM-recommended options are situated approximately 55 cm from the user's eyes, inclined at a 15° angle from the vertical to replicate the distance and angle between the user and a computer interface in natural circumstances.

The task panel is situated at the upper part of the interfaces, presenting the task background, context, and the text to be entered. Several dark-colored text boxes are available to display the text to be entered or any associated instructions, with each box designated for a single sentence. Users can choose the sentence to be entered by clicking on the corresponding text box. Once a user finishes typing a sentence and clicks the "OK" button on the keyboard, the content in the text box is updated to reflect the user's final input.

As depicted in Fig. 2 (b), the LLM recommended options panel displays three text predictions that the LLM considers most likely to align with the user's intent. For the Simplified Spelling and Keyword-to-Sentence Generation methods, users can transfer the LLM-predicted sentence into their input box with a simple click. In the case of Content Prediction, we break down complete sentences into individual words, allowing users to select and insert these words individually without replacing any existing content.

We utilized the virtual QWERTY keyboard provided by Microsoft's MRTK package. To enhance the feedback on users' mid-air typing behavior, we incorporated a visually prominent color and a sound effect to indicate keystrokes. This design approach would improve user experience and typing efficiency [26]. The input box, the "GPT" button for transferring text to the LLM, and the submit button are atop the keyboard.

For readability, text for reading occupies 1.2° of the field of view (FOV), while text used as interactive elements occupies 1.8°, following Microsoft's guidelines for MR interfaces with direct hand interactions [38].

4.2 Prompt engineering with GPT-3.5 model

We employed Open AI's GPT model with specific prompt engineering techniques to implement our three LLM-assisted methods. While the primary focus of the study was not on LLM optimization techniques, we conducted a technical analysis to evaluate various LLM-based solutions. This analysis, detailed in the Appendix, compared their generation quality and response times for the three LLM-assisted methods. The models we compared included GPT-3 and GPT-3.5 with few-shot learning, GPT-3.5 using diverse prompt templates, and GPT-4. Our findings led us to select the GPT-3.5-turbo model combined with specific prompt engineering strategies, as outlined in Table 1. This choice was based on a balance between generation quality—superior to GPT-3 with few-shot learning and nearly on par with GPT-4—and faster response time compared to both GPT-4 and the GPT-3.5 few-shot learning model. When the user enters incomplete textual content, the system will internally convert this content into a predetermined prompt format and transmit it to the model. We additionally adjusted the token limits and the temperature parameter when invoking the OpenAI API to control the diversity and length of the generated texts for each LLM-assisted method. We also implemented a rule-based judgment to guarantee that users receive only one sentence for each option at a time, addressing rare cases where the generated content does not match the expected format.

4.3 Interactions and user flow

Users interact with the virtual interface using the index fingers of both hands through gesture tracking throughout the system, including tapping on the keyboard, touching interface elements, and pressing buttons. This is the standard keyboard interaction technique recommended by the current Oculus system and Unity's XR Interaction Toolkit, and is considered more efficient than using ten fingers under free-hand conditions [14].

During a text entry task with our prototype, the user follows a specific user flow (see Fig. 1 (a)). The first step is selecting a sentence from the task panel to indicate their intention to enter it. Next, the user can choose one of the three LLM-assisted methods and enter incomplete text into the input box. When the user believes enough cues have been provided for the LLM to generate predictions, they can click the "GPT" button located at the top right corner of the keyboard to transfer the incomplete text to the LLM. The LLM offers three predictions based on the chosen method, displayed as interactive options within the LLM-recommended options panel. Users can click on an option to transfer the text to their input box. If the predictions are inaccurate, users can either modify the manual input to request LLM re-generations or manually adjust the text after accepting one of these options. Once satisfied, users can submit the content to the task panel by clicking the "OK" button, simultaneously clearing the input box.

5 Experiments

To assess how our proposed LLM-assisted approach can enhance text entry in VR, we conducted a series of controlled experiments under various text entry scenarios. The experiments aimed to investigate the following research questions:

RQ1: How do the three LLM-assisted methods impact text entry efficiency, error rates, and task loads for users across different text entry scenarios?

RQ2: How do users choose the appropriate LLM-assisted method in different text entry scenarios, and what motivates these choices?

RQ3: To what extent does users' long-term proficiency with LLM-assisted methods influence their performance, and what advanced strategies do long-term users employ to optimize their text entry efficiency?

5.1 Participants

The experiment employed a within-subjects design, involving a total of twenty-two participants (6 males and 6 females), aged between 19 and 26 years. All participants possessed normal vision and demonstrated proficient English communication skills as non-native English speakers, which is supported by their high performances in the College English Test, IELTS, or TOEFL. None of the participants had prior experience with text entry tasks in VR. Prior to the experiment, we assessed the participants' text entry speed (word per minute) on a physical QWERTY keyboard (M = 29.62, SD = 6.85) and used only their left and right index fingers (M = 25.90, SD = 4.61). Each participant provided informed consent by signing a consent form before the experiment, and compensation was remitted in accordance with the University's compensation structure.

5.2 Materials

The experiment aimed to compare text entry performance with and without the LLM assistance. In the experimental condition, participants utilized the prototype described in Section 4, which granted access to LLM-assisted methods. The non-assisted condition employed the same prototype, with the "GPT" button and LLM-recommended options panel disabled. These disabled interface elements were still visible within the virtual environment but were presented in a subdued grey color.

Table 1: Prompts for three LLM-assisted methods

Method	Prompts (System/User)	Temperature	Token limits
Simplified Spelling	**System**: You are an assistant who is very familiar with English conversation and writing. You'll replace words with incomplete spelling with their full state. **User**: (a whole sentence with incompletely spelled words)	1	–
Content Prediction	**System**: You are an assistant who is very familiar with English conversation and writing. Predict what will follow in the incomplete sentence. **User**: (the three most recent sentences as context and the user's unfinished sentence)	1	100
Keyword-to-Sentence Generation	**System**: You are an assistant who is very familiar with English conversation and writing. Generate one sentence for 'type' using the following keywords and an 'emoji' tone. **User**: (keywords, (optional) a selected text type and an emoji)	0.5	100

5.3 Tasks and procedure

We set up three tasks for text entry to cover common text entry needs in work scenarios. These tasks consisted of text transcription, simulated dialogue, and email writing (detailed descriptions are given in the Appendix). For the text transcription task, participants were instructed to enter the text provided to them accurately. The text material was collected from the English Gigaword corpus (LDC2003T05), containing 138 words with an average word character count of 5, which adheres to the standard setting for text entry tasks [22].

In the simulated dialogue task, participants engaged in a dialogue with a given context. They were tasked to assume one of two character roles within the conversation and compose responses for the character's four dialogue bubbles. While participants received concise prompts for each dialogue bubble, they retained the freedom to structure their own sentences. Meanwhile, the email writing task involved composing a cover letter, which is suitable for university participants. A 6-sentence template was provided to outline the main ideas of each sentence.

The experiment followed a within-subjects design (see Fig. 3). Participants first underwent tests for text entry speed under normal and index-finger-only conditions to ensure their proficiency. We then provided an introductory session covering the VR prototype's interfaces, interactions, and LLM-assisted methods, including usage tips. Subsequently, participants had a training phase of at least 15 minutes to interact with the interfaces and LLM-assisted methods for familiarity.

During the formal experiment, participants completed all three text entry tasks of transcription, simulated dialogue, and email writing consecutively, both with LLM assistance and in control conditions. Each task had a time limit of 10-15 minutes, determined based on average typing speeds reported by prior studies [22, 60] and the expected text length. The order of the two text entry conditions (assisted or control) was randomly assigned, ensuring an equal distribution of six participants for each order. Given our interest in RQ2, we allowed participants to freely choose the LLM-assisted method they deemed appropriate for completing each text entry task, and recorded their usage preferences. After each text entry condition, participants filled out questionnaires, followed by in-depth interviews to explore their behavioral and psychological experiences with the prototype.

Following the experimental procedure in Fig. 3, we invited participants to engage in a longer-term study to investigate whether their proficiency with the LLM-assisted methods could further enhance their effectiveness, while providing usage strategies and other insights on the proposed approach. Seven out of the twenty-two participants opted to participate in this extended observation, as detailed in Section 5.5.

5.4 Measurements

We captured the entirety of user interaction behaviors with the interface through the Unity back-end during our experiments. This

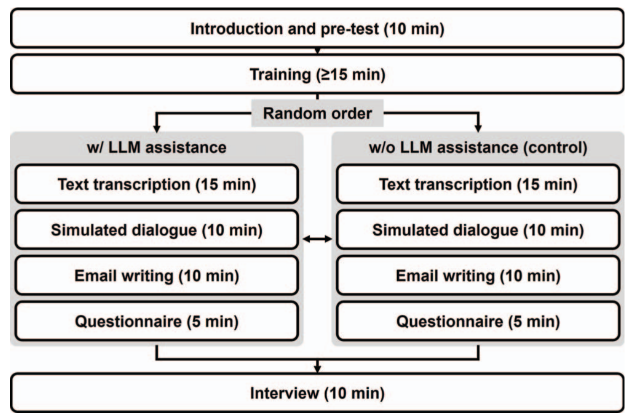

Figure 3: Experiment procedure for the twenty-two participants. Conditions with and without LLM assistance were given in randomized order. Outside of this procedure, seven participants joined a longer-term experiment.

encompassed actions such as typing, deleting, cursor movement, interaction with the LLM, acceptance of specific LLM predictions, and sentence submission. In accordance with prior studies [17, 60], we computed text entry speed in words per minute (WPM) to assess the participants' Manual Typing Speed (MTS).

Given that our proposed methods integrate LLM assistance to enhance text entry, we introduced an additional metric known as Assisted Typing Speed (ATS), which was calculated using the following formula:

$$ATS = \frac{Text_{user} + Text_{LLM}}{Time_{total}} \tag{1}$$

In this formula, the total time encompasses the duration from the initiation of the first keystroke to the final interaction event that results in modifications to the sentence. We additionally introduced the Saved Keystrokes (SK) to describe the proportion of manual keystrokes reduced by LLM to total keystrokes:

$$SK = \frac{Text_{LLM}}{Text_{user} + Text_{LLM}} \tag{2}$$

and the Ideal Assisted Typing Speed (IATS) to represent the assisted typing speed under the ideal condition with zero response time or network delay, which could potentially consume 3-5 seconds for each interaction with the GPT model. The formulas of all metrics are detailed in the Appendix.

$$IATS = \frac{Text_{user} + Text_{LLM}}{Time_{total} - Time_{Response}} \tag{3}$$

We assessed text entry inaccuracies by examining error rates and prediction accuracies. Error rates include both human and language model (LLM)-assisted errors, split into corrected (CER) and non-corrected (NCER) categories [59]. CER counts user corrections, while NCER measures keystrokes to fix uncorrected errors. LLM-corrected characters are excluded from CER. Objective Prediction Accuracy (OPA) was analyzed by measuring the textual embedding similarities between user-input sentences with and without LLM assistance. OpenAI's text-embedding-ada-002 model was used to compute word embeddings for each user-entered sentence. A subjective evaluation of prediction accuracy (SPA) was also recorded utilizing the Augmentative and Alternative Communication Quality Scale [39], which measures user satisfaction with the text recommended by the LLM on a 4-point scale, ranging from "essentially the same" to "totally different and improper for the context".

In line with Grubert et al. [17], we employed the NASA Task Load Questionnaire (NASA-TLX) [18], the System Usability Scale (SUS) [64], and the Virtual Reality Sickness Questionnaire (VRSQ) [23] to assess the approach's improvements in terms of task load, system usability, and user's well-being.

5.5 Longer-term observations

Seven out of the twenty-two participants voluntarily extended their participation in the experiment to provide us with a more in-depth understanding of how their proficiency with the LLM-assisted methods influenced the methods' effectiveness. These participants dedicated half an hour daily for five consecutive days to engage in text entry tasks using our approach, enhancing their familiarity with its performance. Specifically, we allowed these participants to disregard the provided tips on using these methods to encourage them to explore different strategies. After the five-day period, we re-evaluated their utilization of LLM-assisted methods in the three typing scenarios, measuring text entry speed and error rates, and gathering their suggestions on method design and usage strategies.

6 Results

In this section, we present the results of our user study, which is primarily analyzed with ANOVA and paired samples t-tests. The Bonferroni correction was applied for post hoc comparisons. Unless explicitly stated otherwise, all data presented herein met the preconditions necessary for the application of these analyses.

6.1 Entry speed and reduced manual entries

As illustrated in Fig. 4, the LLM-assisted condition exhibited a significant 16.9% increase in ATS during the text transcription task when compared to the control group, $t(21) = 3.528$, $p < 0.001$. Our approach achieved a 72.7% speedup in ATS in the simulated dialogue task (LLM-assisted: $M = 15.12$, $SD = 6.14$; Control: $M = 8.76$, $SD = 2.15$, $t(21) = 5.360$, $p < 0.001$), and a similar 72.6% speedup (LLM-assisted: $M = 15.87$, $SD = 5.95$; Control: $M = 9.20$, $SD = 1.86$, $t(21) = 5.464$, $p < 0.001$) in email writing. The improvement was even more pronounced in IATS, with speedup rates of 22.7%, 115.7%, and 96.2% for the respective tasks, all with increased statistical significance. We also noted a minor reduction in MTS among participants who used LLM assistance, particularly in simulated dialogue and email writing tasks (simulated dialogue: $t(21) = -2.979$, $p = 0.004$; email writing: $t(21) = -2.413$, $p = 0.013$). This variance in MTS can be attributed to our calculation method, which accounts for the time spent on interactions with the LLM and waiting for responses.

We analyzed participants' utilization of the three methods and compared their impact on user's typing speed and the reduction in participants' manual keystrokes, as illustrated in Fig. 5 (a) and (b). An ANOVA revealed a significant effect of the utilization of LLM-assisted methods on ATS, $F(3,71) = 13.645$, $p < 0.001$, $\eta 2 = 0.366$. Post hoc comparisons further highlighted significantly improved ATS with Content Prediction ($M = 15.85$, $SD = 6.77$, $p < 0.001$) and Keyword-to-Sentence Generation ($M = 16.06$, $SD = 5.33$, $p < 0.001$) compared to the non-assisted condition ($M = 8.97$, $SD = 0.68$). Simplified Spelling ($M = 11.06$, $SD = 2.21$) might also have an improved ATS, with $t(21) = 4.278$, $p < 0.001$ under a paired sample t-test. Regarding the reduction in participants' manual keystrokes (SK), the three LLM-assisted methods achieved average reduction rates of 16.4%, 49.9%, and 43.7% respectively. A highly significant effect was found of the choice of these methods on the reduction rate of manual keystrokes, $F(2,50) = 19.488$, $p < 0.001$, $\eta 2 = 0.438$. Post hoc comparisons further highlighted significant differences between Simplified Spelling and Content Prediction ($p < 0.001$) and between Simplified Spelling and Keyword-to-Sentence Generation ($p < 0.001$). The overall response latencies observed with the three LLM-assisted methods are demonstrated in Fig. 5 (c), with average latencies of 1.68 ($SD = 0.52$), 1.77 ($SD=0.76$), and 1.54 ($SD = 0.34$) seconds, respectively, with a statistical difference ($F(2,340) = 4.808$, $p = 0.009$). Post hoc tests found differences between Simplified Spelling and Keyword-to-Sentence Generation ($p = 0.046$) and between Content Prediction and Keyword-to-Sentence Generation ($p = 0.031$). These differences in response latency generally align with the amount of text transferred through the GPT API in the three LLM-assisted methods.

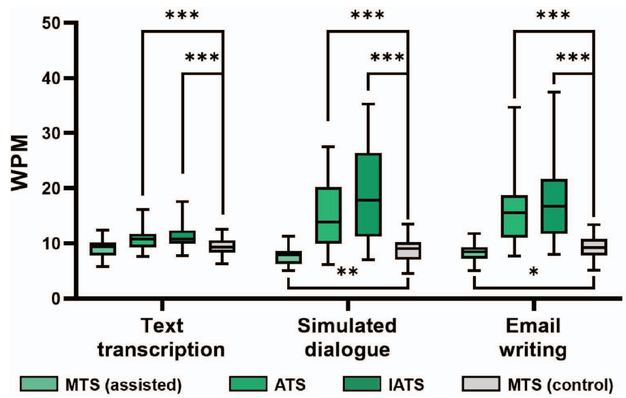

Figure 4: Improvement in user's text entry speed under the three text entry tasks. Significant improvements in ATS and IATS were observed in all tasks when using our approach.

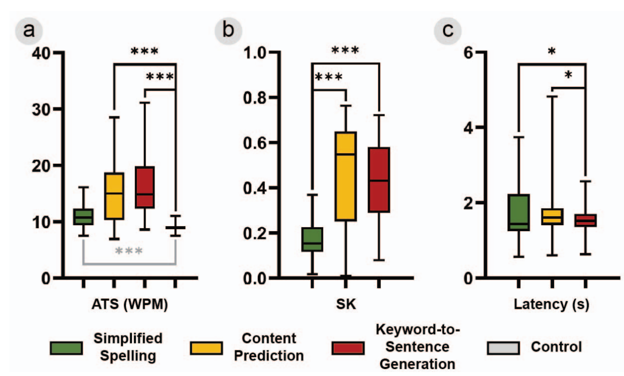

Figure 5: (a) The assisted typing speed (ATS) of the three LLM-assisted methods and the manual typing speed (MTS) of the non-assisted condition (control). (b) The ratio of saved keystrokes (SK) and (c) the response latency of the three methods.

6.2 Error rates and prediction accuracy

The corrected error rates (CER) and non-corrected error rates (CER) under the three LLM-assisted methods and the control condition are exhibited in Fig. 6 (a) and (b). According to ANOVA, the utilization of LLM-assisted methods did not have a significant impact on the

CER ($F(3, 71) = 0.824$, $p = 0.485$) and NCER ($F(3, 71) = 1.108$, $p = 0.352$). Post-hoc comparisons further indicated no statistically significant differences in the participants' manual error (either corrected or uncorrected) with each LLM-assisted method compared to the control group.

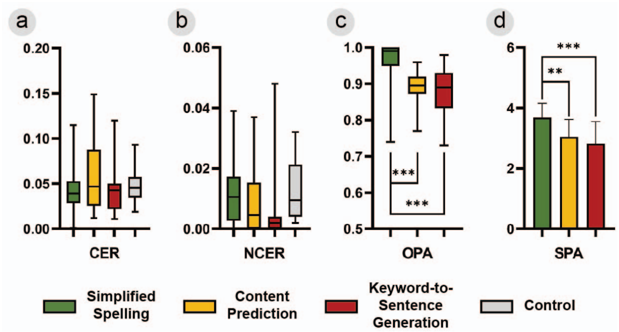

Figure 6: (a) The corrected (CER) and (b) non-corrected error rates (NCER) [59] of the three LLM-assisted methods compared with the non-assisted condition. (c) The objective prediction accuracy (OPA) of the three methods. (d) The subjective prediction accuracy (SPA) of the three methods.

Objective prediction accuracy (OPA) and subjective prediction accuracy (SPA) of the three LLM-assisted methods are presented in Fig. 6 (c) and (d). OPA demonstrated the cosine similarities between word embeddings of the corresponding sentences a participant entered under LLM-assisted and non-assisted conditions, while SPA described participants' perceived prediction accuracy using the AAC scale [39]. An ANOVA revealed a significant difference between the OPA of the three methods, $F(2,271) = 97.329$, $p < 0.001$, $\eta 2 = 0.418$. Post-hoc test further demonstrated significant differences between Simplified Spelling ($M = 96.89\%$, $SD = 0.045$) and Content Prediction ($M = 89.39\%$, $SD = 0.041$), $p < 0.001$, and between Simplified Spelling and Keyword-to-Sentence Generation ($M = 88.11\%$, $SD = 0.060$), $p < 0.001$. No significant difference was observed between Content Prediction and Keyword-to-Sentence Generation ($p = 0.792$). This is generally acceptable as participants would naturally attempt to avoid entering the same content repeatedly under the two conditions by making some changes to the grammar and details. Similar differences were also observed on SPA, $F(2,63) = 12.077$, $p < 0.001$, $\eta 2 = 0.277$, where the three LLM-assisted methods achieved SPA scores of 3.68, 3.04, and 2.82, respectively, corresponding to the level between "essentially the same as their expectations" to "very similar, with minor differences." Though the latter two methods exhibited lower prediction accuracies, their error rates were not significantly higher. This indicates that participants were still able to trade off a controllable number of corrections for a significant improvement in the text entry efficiency for these two methods.

6.3 Method preferences under three scenarios

We analyzed participants' preferences for LLM-assisted methods in the three text entry tasks by examining the back-end records of user interactions with the LLM (see Fig. 7). In the text transcription task, Simplified Spelling was selected in 88% of LLM prediction events, while Content Prediction and Keyword-to-Sentence Generation were utilized in 8% and 4%. During simulated dialogue, Keyword-to-Sentence Generation was the preferred choice in 48% of the prediction events, while Simplified Spelling and Content Prediction both accounted for 26%, respectively. In the context of email writing, Keyword-to-Sentence Generation emerged as the predominant choice at 74%, followed by Simplified Spelling at 21%, and Content Prediction at 5%.

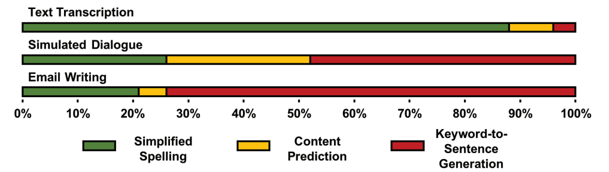

Figure 7: Participant preferences for LLM-assisted methods in the three text entry tasks, based on the number of prediction events using each method.

Among the three LLM-assisted methods, participants showed a lower preference for the Content Prediction method. They mentioned being overly cautious when selecting the grammatical structures for prediction, leading to increased cognitive load. Furthermore, participants had to manually trigger this method more frequently, which led to additional manual interactions and more waiting periods that might disrupt their continuous experience and thinking process.

6.4 Task loads and other self-reported metrics

Results from the NASA-TLX questionnaire (see Fig. 8) showed that the proposed approach ($M = 9.87$, $SD = 3.87$) significantly reduced the mental load compared to the control group ($M = 11.32$, $SD = 4.14$), $t(21) = -2.055$, $p = 0.026$. Our approach ($M = 10.05$, $SD = 4.15$) also obtained a highly significant advantage in the physical load relative to the control group ($M = 13.45$, $SD = 4.55$), $t(21) = -4.400$, $p < 0.001$. Significant improvements were also observed in the temporal demand ($t(21) = -2.531$, $p = 0.010$), performance ($t(21) = -2.057$, $p = 0.026$), effort ($t(21) = -3.056$, $p = 0.003$) and frustration components ($t(21) = -2.130$, $p = 0.023$). Participants also reported reduced cognitive effort in terms of spelling, grammar, and sentence structure, as well as significantly fewer manual keystrokes.

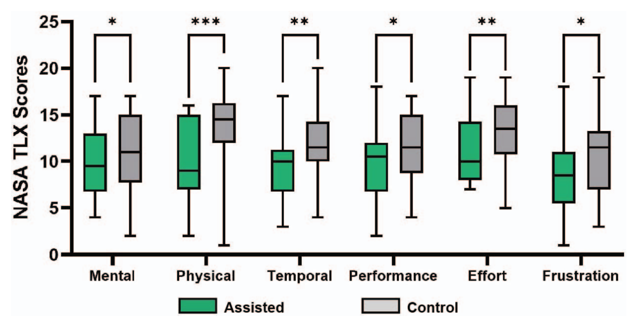

Figure 8: The results of the NASA-TLX questionnaire.

We collected self-reported metrics, including system usability (SUS), spatial presence (IPQ), and simulator sickness (VRSQ). Our method showed higher usability ($M = 51.36$, $SD = 8.25$) than the control group ($M = 46.03$, $SD = 7.49$), $t(21) = 3.064$, $p = 0.003$. It also resulted in less simulator sickness ($M = 47.84$, $SD = 12.14$) compared to the control ($M = 55.34$, $SD = 14.35$), $t(21) = -2.289$, $p = 0.016$. No significant differences were found in spatial presence, $t(21) = 0.793$, $p = 0.327$, possibly due to similar interfaces and environments in both text input conditions, resulting in minimal changes in spatial perceptions.

6.5 The effect of proficiency with LLM-assisted methods

In a more extended study involving seven participants (as depicted in Fig. 9), users' proficiency with LLM-assisted methods showed impacts on the methods' effectiveness. Over five days of usage, nearly all participants demonstrated improved ATS, IATS, and further reduced manual input. Participants experienced substantial enhancements in ATS across the three LLM-assisted methods, with improvements of 19.1% ($t(6) = 4.015$, $p = 0.003$), 38.6% ($t(6) = 3.276$, $p = 0.008$), and 41.8% ($t(6)= 2.441$, $p = 0.025$) compared to their initial performance using these methods, resulting in typing speeds of 12.8, 15.9, and 22.2 WPM, respectively. The ratios of improvements in IATS were 36.8% ($t(6) = 1.772$, $p = 0.063$), 33.9% ($t(6) = 3.276$, $p = 0.021$), and 52.9% ($t(6) = 2.859$, $p = 0.025$), leading to average ideal typing speed of 15.5, 22.7, and 26.6 WPM for the three LLM-assisted methods. The increased text entry efficiency could be attributed to a general reduction in manual input, with the SK metric increased from 15.9% to 21.5% in Simplified Spelling ($p = 0.034$), from 37.5% to 56.5% in Content Prediction ($p = 0.002$), and from 47.9% to 57.7% in Keyword-to-Sentence Generation ($p = 0.011$). Additionally, longer practice sessions led to enhanced manual entry speeds under certain circumstances, with MTS increased by 9.56% in Simplified Spelling ($t(6) = 1.687$, $p = 0.071$) and 19.95% in Keyword-to-Sentence Generation ($t(6) = 2.075$, $p = 0.042$). However, given the small sample size of this longer-term observation session, these significant differences should be interpreted with caution.

Extended observations revealed valuable insights into participants' strategies with the three LLM-assisted methods. In Simplified Spelling, participants primarily preserved consonants with occasional vowel retention. Notably, one participant achieved a remarkable 34% reduction in manual input and nearly 20 WPM typing speed after practice. This participant reported occasionally omitting individual consonants but adding the last letter to indicate tense or singular-plural forms. Meanwhile, trust in the LLM's prediction accuracy also led him to proceed without stopping for manual errors. In the Content Prediction method, some participants input initial words followed by the first letters of subsequent words to provide additional context to guide LLM predictions. Surprisingly, participants sometimes treated all words in the three LLM predictions as a relevant thesaurus, skillfully blending manual input with words from the thesaurus to construct their sentences. For Keyword-to-Sentence Generation, participants manually typed fewer verbs associating the key concepts and employed slight keyword abbreviations. However, we observed that not all participants showed increased MTS. Feedback indicated that some participants were primarily concerned with identifying optimal strategies to minimize text input. Despite the minor decrease in MTS, this focus positively impacted their ATS and IATS performance.

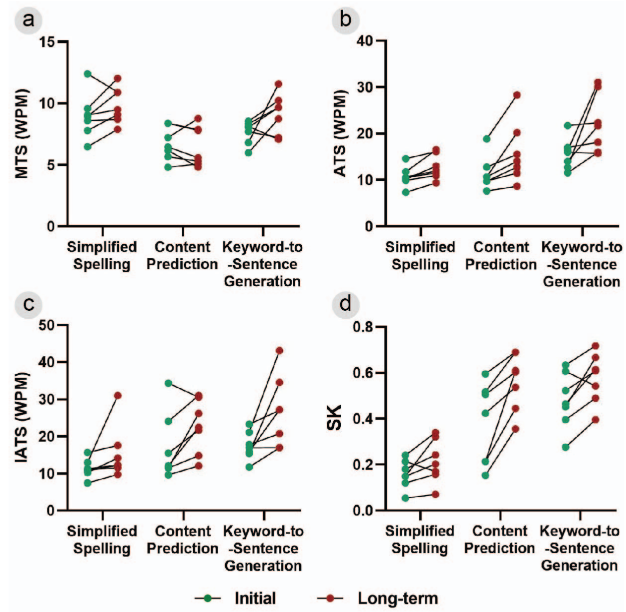

Figure 9: Changes in the seven participants' performances in (a) MTS, (b) ATS, (c) IATS, and (d) SK using the three LLM-assisted methods after 5 days of use.

7 Discussion

7.1 Enhancing text entry with LLM assistance

Regarding RQ1, the experimental results strongly support the effectiveness of our LLM-assisted approach in improving text entry efficiency and task loads. Participants achieved an average MTS of 8.98 WPM, in line with previous studies using free-hand typing [2, 60, 68]. This result validates our control group's design as an effective benchmark, confirming that minor interface differences did not introduce deliberate interaction challenges. Novice users experienced efficiency gains of 21.4%, 74.0%, and 76.3% in ATS, as well as 28.0%, 147.7%, and 96.2% in IATS when using our three LLM-assisted methods. These efficiency levels surpass common text input techniques mentioned in prior literature, including raycasting, controller tapping, eye tracking, and head movements [29, 55, 60, 73]. The speed improvements achieved by our approach are comparable to or better than other text input enhancement approaches, such as gesture-based input (22% [73]), specialized keyboard layouts (26.2% with drum-like keyboards [5]), and physical interfaces (34.7% with smartphones [7], 31.8% with touchscreen keyboards [17], and 32.1% with tracked physical surfaces [14]). Some studies achieved higher efficiencies, but they typically involved additional devices [25, 44]. Extended practice with our assisted methods led to further improved input efficiencies of 12.8, 15.9, and 22.2 WPM. Flexibly combining these methods in practical scenarios would allow participants to approach their real-world input speeds (as measured before the experiment), which partially answers RQ3.

Our proposed approach effectively grants users control over the textual content, allowing for swift modifications of generated text through minimal interactions. Experiments revealed the error rates and prediction accuracies of three LLM-assisted methods, with CER and NCER demonstrating no significant differencing from the controlled conditions, and considerable OPA performances ranging from 88.11% to 96.89%. Despite occasional predictions that did not precisely match user expectations, the LLM-assisted approach maintained error rates comparable to traditional text entry methods, suggesting that users can correct LLM-generated text without increasing the number of corrections. These results underscore the strong connection between the LLM predictions and users' manual input, highlighting user control over the text entry process. Moreover, back-end records showed that the LLM typically does not introduce entirely new concepts (like key objects, tasks, scenes, and activities) into the predictions. Instead, it focuses on enhancing existing content in predictable ways, such as spelling, grammar, and objects that are either related to existing concepts or previously mentioned by users. This observation confirms our design approach that while users retain primary control in the conceptualization phase of language generation, deciding "what to write", the specific organization of grammar and structures can be efficiently delegated to the LLM to varying extents, under user supervision.

Participants consistently reported reduced task loads when utilizing the proposed approach. Interviews revealed how these LLM-assisted methods alleviated cognitive and linguistic challenges stemming from sluggish typing interactions. In conventional VR text entry systems, users often need to expend significant effort to ensure accuracy, as identifying and correcting errors is notably difficult. This detracts from their ability to concentrate on structuring their language effectively. Our Simplified Spelling approach resolves this by leveraging the LLM's capability to automatically rectify clear errors and proficiently handle simplified, ambiguous inputs. Additionally, the inherent inefficiency of traditional VR text entry methods results in prolonged periods of typing out the main concepts of sentences after their conceptualization. Feedback from participants indicates that this often results in extended gaps when entering several concepts, necessitating reliance on short-term memory to maintain and recollect formulated concepts. Contrastingly, our Keyword-to-Sentence Generation and Content Prediction techniques markedly diminish these intervals, facilitating quicker and smoother input completion. This aligns the input process more closely with the real-world experience of seamless coordination between thought and typing. Such feedback underscores the effectiveness of our methods in addressing the unique cognitive challenges associated with VR text entry.

Another significant advantage of the LLM-assisted text entry approach is its generalizability. It adapts seamlessly across diverse text entry scenarios, enhancing efficiency consistently. As an assistive technique, it has the potential to complement a wide range of existing VR text entry techniques, including specialized keyboards, physical devices, and hands-free interactions, further approaching the efficiency of real-world typing without introducing external constraints. Its broad applicability extends to scenarios like education in VR environments, and facilitating text-based communication for populations with physical disabilities.

7.2 User behavior and usage strategies

Back-end records and interviews conducted with 22 participants shed light on novice users' ability to effectively select the most suitable LLM-assisted method based on sentence features and the specific text entry scenario, addressing RQ2. These findings highlighted significant variability in participants' preferences for the three LLM-assisted methods across different text entry scenarios, aligning with the intended design of these methods. According to the experimental results, the methods of Simplified Spelling, Content Prediction, and Keyword-to-Sentence Generation showed a progressive increase in ATS and a decrease in prediction accuracy. This pattern aligns with the varying degrees of manual input and control in each method, influencing users' usage preferences. In tasks that demanded precision and accuracy, participants leaned toward the use of Simplified Spelling due to its rich context clues. Conversely, in scenarios where users sought greater freedom in crafting their language, Keyword-to-Sentence Generation emerged as the favored method, valued for its efficiency and the diversity of results it offered, albeit with less user control. The Content Prediction method presented a more neutral choice, providing enhanced efficiency compared to Simplified Spelling with reduced manual input, while offering additional context and frequent interactions compared to Keyword-to-Sentence Generation, which leads to higher control. Notably, Keyword-to-Sentence Generation visibly garnered more popularity than Content Prediction in practical text entry scenarios, signifying participants' strong willingness to trade off some accuracy for a smoother, more efficient typing experience with fewer interactions and lower latencies.

Despite our training encouraging trust in LLM predictions, some participants displayed conservative behaviors, requesting short phrase or clause predictions with Simplified Spelling instead of predicting entire sentences, resulting in reduced efficiency and accuracy. Some participants may tend to provide the LLM with overly complete text clues, which hinders the LLM's ability to reduce manual input effectively. Low confidence when applying Content Prediction is also mentioned by some participants, as they found it difficult to intuitively determine when to activate this method. This highlights the critical impact of alignment between LLM-assisted methods and users' cognitive processes on user confidence and engagement.

Longer-term experiments with the seven participants offered advanced usage insights for the proposed methods and addressed our RQ3 on user proficiency. For instance, users mastered the technique of optimizing manual input in Simplified Spelling by preserving consonants and word endings. With Content Prediction, they alternated between prediction and manual input, while strategically providing clues like initials or abbreviations for challenging words. In Keyword-to-Sentence Generation, participants used punctuation and conjunctions to demarcate clauses, omitting verbs and object interactions. These advanced strategies emphasize the methods' underexplored potential in creating a natural human-AI collaborative working environment.

7.3 Limitations and future work

While our approach significantly improved text entry efficiency and reduced workload, it necessitated trade-offs in interaction design due to online LLMs' response times and network latency. Given that each LLM interaction added a waiting period averaging between 1.54 to 1.77 seconds, frequent activation of LLM-assisted methods could result in regular delays in system responses, potentially diminishing efficiency and creating disjointed user experiences. This concern was especially evident in the low preference for the Content Prediction method, where participants expressed dissatisfaction due to increased wait times and more interactions with the LLM. To mitigate this, our prototypes employed a trigger-based design, where participants were guided to activate LLM assistance only after entering substantial text content. Future improvements could involve deploying local LLMs or developing pruned LLMs optimized for text entry. Participants also noted the benefits of automatic LLM assistance during interviews, highlighting fewer interaction steps and a better sense of control.

Participants, particularly those with extensive experience, expressed a preference for a mixed LLM assistance approach that integrates various methods, noting that pre-designed LLM methods can sometimes restrict flexibility in language formulation. This calls for a more complex experimental setup with careful fine-tuning of the LLM to better align with users' cognitive processes in language formation. Additionally, some participants highlighted the potential for LLMs to adapt to individual linguistic styles, such as preferred sentence structures and vocabulary. This could be achieved through personalized prompt engineering templates or integrating users' linguistic profiles into the LLMs, potentially enhancing prediction accuracy and improving user satisfaction and engagement.

8 Conclusion

In this study, we propose an innovative approach to enhance text entry efficiency in VR while concurrently alleviating user's task loads by harnessing the contextual understanding and text generation capabilities of LLMs. Grounded in the contextual predictability of English text at the word, grammatical structure, and sentence levels, we introduce three LLM-assisted methods: Simplified Spelling, Content Prediction, and Keyword-to-Sentence Generation. These methods are integrated into a VR prototype with a free-hand typing technique. Comprehensive user experiments, encompassing various text-entry scenarios and both short- and long-term observations, demonstrate the effectiveness of these LLM-assisted methods. For novices, manual input is reduced by approximately 16.4%, 46.7%, and 43.7%, while long-term trained users achieve reductions of about 21.5%, 56.2%, and 57.7% across the three methods. This translates to significantly improved text entry efficiencies ranging from 15.50 to 26.65 WPM, along with significantly reduced task loads. Importantly, the well-prompted LLM displays commendable predictive accuracy and does not yield a higher correction rate compared to purely manual input. Users' cognitive processes and strategies when utilizing different LLM-assisted methods are also discussed. The proposed approach consistently yields positive outcomes across diverse text entry scenarios and holds significant promise for synergistic collaboration with other VR text input techniques to further elevate efficiency to levels akin to real-world text entry. Future research on the user's language organization process will further assist our approach in understanding the complementary features between LLMs and human users during textual communication, facilitating the design of a human-AI collaboration approach in VR workplaces.

ACKNOWLEDGMENTS

This work was supported by a grant from the National Natural Science Foundation of China (62207023) and by the ZJU-SUTD IDEA Grant (188170-11102) of the Ng Teng Fong Charitable Foundation.

REFERENCES

[1] A. Acharya, B. Singh, and N. Onoe. LLM based generation of item-description for recommendation system. In Proc. RecSys, p. 1204–1207. ACM, 2023. doi: 10.1145/3604915.3610647

[2] J. Adhikary and K. Vertanen. Text entry in virtual environments using speech and a midair keyboard. IEEE Transactions on Visualization and Computer Graphics, 27(5):2648–2658, 2021. doi: 10.1109/TVCG.2021.3067776

[3] T. J. Aveni, A. Fox, and B. Hartmann. 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, pp. 1–3, 2023.

[4] M. A. Bakar, Y.-T. Tsai, H.-H. Hsueh, and E. C. Li. Crowbarlimbs: A fatigue-reducing virtual reality text entry metaphor. IEEE Transactions on Visualization and Computer Graphics, 29(5):2806–2815, 2023. doi: 10.1109/TVCG.2023.3247060

[5] C. Boletsis and S. Kongsvik. Controller-based text-input techniques for virtual reality: An empirical comparison. International Journal of Virtual Reality, 19(3):2–15, oct 2019. doi: 10.20870/IJVR.2019.19.3.2917

[6] C. Boletsis and S. Kongsvik. Text input in virtual reality: A preliminary evaluation of the drum-like VR keyboard. Technologies, 7(2), apr 2019. doi: 10.3390/technologies7020031

[7] S. Boustila, T. Guégan, K. Takashima, and Y. Kitamura. Text typing in VR using smartphones touchscreen and HMD. In Proc. VR, pp. 860–861. IEEE, 2019. doi: 10.1109/VR.2019.8798238

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020. doi: 10.18653/v1/2021.mrl-1.1

[9] L. Burgueño, R. Clarisó, S. Gérard, S. Li, and J. Cabot. An NLP-based architecture for the autocompletion of partial domain models. In Advanced Information Systems Engineering, pp. 91–106. Springer International Publishing, jun 2021. doi: 10.1007/978-3-030-79382-1_6

[10] S. Cai, S. Venugopalan, K. Tomanek, A. Narayanan, M. R. Morris, and M. P. Brenner. Context-aware abbreviation expansion using large language models. In Proc. NAACL, pp. 1261–1275. ACL, Seattle, United States, jul 2022. doi: 10.18653/v1/2022.naacl-main.91

[11] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, and N. A. Smith. All that's 'human' is not gold: Evaluating human evaluation of generated text. In Proc. ACL — IJCNLP, pp. 7282–7296. ACL, aug 2021. doi: 10.18653/v1/2021.acl-long.565

[12] M.-C. de Marneffe and J. Nivre. Dependency grammar. Annual Review of Linguistics, 5(1):197–218, 2019. doi: 10.1146/annurev-linguistics-011718-011842

[13] T. J. Dube and A. S. Arif. Text entry in virtual reality: A comprehensive review of the literature. In Human-Computer Interaction. Recognition and Interaction Technologies, pp. 419–437. Springer International Publishing, 2019. doi: 10.1007/978-3-030-22643-5_33

[14] J. Dudley, H. Benko, D. Wigdor, and P. O. Kristensson. Performance envelopes of virtual keyboard text input strategies in virtual reality. In Proc. ISMAR, pp. 289–300. IEEE, 2019. doi: 10.1109/ISMAR.2019.00027

[15] S. Fallah and S. Mackenzie. H4VR: One-handed gesture-based text entry in virtual reality using a four-key keyboard. In Proc. CHI EA, pp. 1–7. ACM, 2023. doi: 10.1145/3544549.3585876

[16] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. Effects of hand representations for typing in virtual reality. In Proc. VR, pp. 151–158. IEEE, 2018. doi: 10.1109/VR.2018.8446250

[17] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. Text entry in immersive head-mounted display-based virtual reality using standard keyboards. In Proc. VR, pp. 159–166. IEEE, 2018. doi: 10.1109/VR.2018.8446059

[18] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In Advances in psychology, vol. 52, pp. 139–183. Elsevier, 1988. doi: 10.1016/s0166-4115(08)62386-9

[19] Z. He, C. Lutteroth, and K. Perlin. Tapgazer: Text entry with finger tapping and gaze-directed word selection. In Proc. CHI, pp. 1–16. ACM, 2022. doi: 10.1145/3491102.3501838

[20] B. Jackson, L. B. Caraco, and Z. M. Spilka. Arc-type and Tilt-type: Pen-based immersive text input for room-scale VR. In Proc. SUI. ACM, New York, NY, USA, 2020. doi: 10.1145/3385959.3418454

[21] G. Kempen. Conceptualizing and formulating in sentence production. In Sentence production: Developments in research and theory, pp. 259–274. Erlbaum, jan 1977.

[22] F. Kern, F. Niebling, and M. E. Latoschik. Text input for non-stationary XR workspaces: Investigating tap and word-gesture keyboards in virtual and augmented reality. IEEE Transactions on Visualization and Computer Graphics, 29(5):2658–2669, 2023. doi: 10.1109/TVCG.2023.3247098

[23] H. K. Kim, J. Park, Y. Choi, and M. Choe. Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment. Applied ergonomics, 69:66–73, 2018. doi: 10.1016/j.apergo.2017.12.016

[24] Y. R. Kim and G. J. Kim. HoVR-Type: Smartphone as a typing interface in VR using hovering. In Proc. ICCE, pp. 200–203. IEEE, 2017. doi: 10.1109/ICCE.2017.7889285

[25] P. Knierim, V. Schwind, A. M. Feit, F. Nieuwenhuizen, and N. Henze. Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands. In Proc. CHI, p. 1–9. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173919

[26] A. Krasner, J. L. Gabbard, and G. Burnett. Musikeys: Investigating auditory-physical feedback replacement technique for mid-air typing. In Proc. ISMAR-Adjunct, pp. 510–512. IEEE, 2021. doi: 10.1109/ISMAR-Adjunct54149.2021.00124

[27] F. Kuester, M. Chen, M. E. Phair, and C. Mehring. Towards keyboard independent touch typing in VR. In Proc. VRST, p. 86–95. ACM, New York, NY, USA, 2005. doi: 10.1145/1101616.1101635

[28] S. Kuś and R. Szmurło. CNN-based character recognition for a contextless text input system in immersive VR. In Proc. CPEE, pp. 1–4. IEEE, 2021. doi: 10.1109/CPEE54040.2021.9585252

[29] J. Leng, L. Wang, X. Liu, X. Shi, and M. Wang. Efficient flower text entry in virtual reality. IEEE Transactions on Visualization and Computer Graphics, 28(11):3662–3672, nov 2022. doi: 10.1109/TVCG.2022.3203101

[30] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What makes good in-context examples for GPT-3? arXiv preprint arXiv:2101.06804, jan 2021. doi: 10.48550/arXiv.2101.06804

[31] X. Lu, D. Yu, H.-N. Liang, X. Feng, and W. Xu. Depthtext: Leveraging head movements towards the depth dimension for hands-free text entry in mobile virtual reality systems. In Proc. VR, pp. 1060–1061. IEEE, 2019. doi: 10.1109/VR.2019.8797901

[32] X. Lu, D. Yu, H.-N. Liang, W. Xu, Y. Chen, X. Li, and K. Hasan. Exploration of hands-free text entry techniques for virtual reality. In Proc. ISMAR, pp. 344–349. IEEE, 2020. doi: 10.1109/ISMAR50242.2020.00061

[33] X. Ma, Z. Yao, Y. Wang, W. Pei, and H. Chen. Combining brain-computer interface and eye tracking for high-speed text entry in virtual reality. In Proc. IUI, p. 263–267. ACM, New York, NY, USA, 2018. doi: 10.1145/3172944.3172988

[34] P. Majaranta, U.-K. Ahola, and O. Špakov. Fast gaze typing with an adjustable dwell time. In Proc. CHI, p. 357–360. ACM, New York, NY, USA, 2009. doi: 10.1145/1518701.1518758

[35] Z. Malah. Lexical cohesion: A brief review on theoretical emergence, development and practical application in discourse studies. International Journal of Linguistics, 8(2):46–61, dec 2020. doi: 10.15640/ijlc.v8n2a6

[36] K. R. McKeown and D. R. Radev. Collocations. Handbook of Natural Language Processing. Marcel Dekker, pp. 1–23, jan 2000.

[37] T. Menzner, A. Otte, T. Gesslein, J. Grubert, P. Gagel, and D. Schneider. A capacitive-sensing physical keyboard for VR text entry. In Proc. VR, pp. 1080–1081. IEEE, 2019. doi: 10.1109/VR.2019.8797754

[38] Microsoft. Typography in mixed reality, January 2021.

[39] M. Mitchell and R. Sproat. Discourse-based modeling for AAC. In

Proc. SLPAT, pp. 9–18. ACL, USA, 2012.

[40] A. Nguyen, S. Bittman, and M. Zank. Text input methods in virtual reality using radial layouts. In Proc. VRST. ACM, New York, NY, USA, 2020. doi: 10.1145/3385956.3422114

[41] T. Ocal and L. C. Ehri. Spelling pronunciations help college students remember how to spell difficult words. Reading and Writing, 30:947–967, may 2017. doi: 10.1007/s11145-016-9707-z

[42] N. A. Omoregbe, I. O. Ndaman, S. Misra, O. O. Abayomi-Alli, R. Damaševičius, and A. Dogra. Text messaging-based medical diagnosis using natural language processing and fuzzy logic. Journal of Healthcare Engineering, 2020:1–14, sep 2020. doi: 10.1155/2020/8839524

[43] OpenAI. GPT-4 technical report. arXiv, pp. 2303–08774, mar 2023. doi: 10.48550/arXiv.2303.08774

[44] A. Otte, T. Menzner, T. Gesslein, P. Gagel, D. Schneider, and J. Grubert. Towards utilizing touch-sensitive physical keyboards for text entry in virtual reality. In Proc. VR, pp. 1729–1732. IEEE, 2019. doi: 10.1109/VR.2019.8797740

[45] A. Otte, D. Schneider, T. Menzner, T. Gesslein, P. Gagel, and J. Grubert. Evaluating text entry in virtual reality using a touch-sensitive physical keyboard. In Proc. ISMAR-Adjunct, pp. 387–392. IEEE, 2019. doi: 10.1109/ISMAR-Adjunct.2019.000-4

[46] Z. Pan, Z. Pan, T. Luo, and M. Zhang. Exploring the use of smartphones as input devices for the mixed reality environment. In Proc. VRCAI. ACM, New York, NY, USA, 2023. doi: 10.1145/3574131.3574451

[47] H. Patel, B. Patel, and K. Lad. Jodani: A spell checking and suggesting tool for Gujarati language. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 94–99, 2021. doi: 10.1109/Confluence51648.2021.9377072

[48] D.-M. Pham and W. Stuerzlinger. Hawkey: Efficient and versatile text entry for virtual reality. In Proc. VRST. ACM, 2019. doi: 10.1145/3359996.3364265

[49] S. Pick, A. S. Puika, and T. W. Kuhlen. Swifter: Design and evaluation of a speech-based text input metaphor for immersive virtual environments. In Proc. 3DUI, pp. 109–112. IEEE, 2016. doi: 10.1109/3DUI.2016.7460039

[50] T. A. Pirinen and K. Lindén. State-of-the-art in weighted finite-state spell-checking. In Computational Linguistics and Intelligent Text Processing, pp. 519–532. Springer Berlin Heidelberg, 2014. doi: 10.1007/978-3-642-54903-8_43

[51] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. OpenAI Blog, 2018.

[52] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

[53] V. Rajanna and J. P. Hansen. Gaze typing in virtual reality: Impact of keyboard design, selection method, and motion. In Proc. ETRA. ACM, New York, NY, USA, 2018. doi: 10.1145/3204493.3204541

[54] D. Schneider, A. Otte, T. Gesslein, P. Gagel, B. Kuth, M. S. Damlakhi, O. Dietz, E. Ofek, M. Pahud, P. O. Kristensson, J. Müller, and J. Grubert. Reconviguration: Reconfiguring physical keyboards in virtual reality. IEEE Transactions on Visualization and Computer Graphics, 25(11):3190–3201, nov 2019. doi: 10.1109/TVCG.2019.2932239

[55] K. Sengupta, R. Menges, C. Kumar, and S. Staab. Gazethekey: Interactive keys to integrate word predictions for gaze-based text entry. In Proc. IUI Companion, pp. 121–124. ACM, New York, NY, USA, 2017. doi: 10.1145/3030024.3038259

[56] J. Shen, J. J. Dudley, J. Zheng, B. Byrne, and P. O. Kristensson. Promptor: A conversational and autonomous prompt generation agent for intelligent text entry techniques. arXiv preprint arXiv:2310.08101, 2023.

[57] J. Shen, B. Yang, J. J. Dudley, and P. O. Kristensson. Kwickchat: A multi-turn dialogue system for AAC using context-aware sentence generation by bag-of-keywords. In Proc. IUI, p. 853–867. ACM, New York, NY, USA, 2022. doi: 10.1145/3490099.3511145

[58] S. M. SHIEBER and R. NELKEN. Abbreviated text input using language modeling. Natural Language Engineering, 13(2):165–183, jun 2007. doi: 10.1017/S1351324906004311

[59] R. W. Soukoreff and I. S. MacKenzie. Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric. In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 113–120, 2003.

[60] M. Speicher, A. M. Feit, P. Ziegler, and A. Krüger. Selection-based text entry in virtual reality. In Proc. CHI, p. 1–13. ACM, 2018. doi: 10.1145/3173574.3174221

[61] State Bureau of Technical Supervision. Human dimensions of chinese adults. GB/T 10000-1988, 1988.

[62] M. Tan, Y. Dai, D. Tang, Z. Feng, G. Huang, J. Jiang, J. Li, and S. Shi. Exploring and adapting Chinese GPT to pinyin input method. In Proc. ACL, pp. 1899–1909. ACL, Dublin, Ireland, may 2022. doi: 10.18653/v1/2022.acl-long.133

[63] K. Tominaga, S. Fujita, R. Takakura, and B. Shizuki. Investigating the effects of position and angle of virtual keyboard on text entry performance and workload. In Proc. ASIAN-CHI, p. 25–27. ACM, New York, NY, USA, 2021. doi: 10.1145/3429360.3468174

[64] T. Tullis and B. Albert. Chapter 6 - self-reported metrics. In T. Tullis and B. Albert, eds., Measuring the User Experience (Second Edition), Interactive Technologies, pp. 121–161. Morgan Kaufmann, Boston, second edition ed., 2013. doi: 10.1016/B978-0-12-415781-1.00006-6

[65] J. Tóth, A. Kondelová, and G. Rozinaj. Natural language processing of abbreviations. In Proceedings ELMAR-2011, pp. 225–228. IEEE, 2011.

[66] S. Valencia, R. Cave, K. Kallarackal, K. Seaver, M. Terry, and S. K. Kane. "The less I type, the better": How AI language models can enhance or impede communication for AAC users. In Proc. CHI. ACM, New York, NY, USA, 2023. doi: 10.1145/3544548.3581560

[67] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In Proc. NIPS, p. 6000–6010. Curran Associates Inc., Red Hook, NY, USA, 2017. doi: 10.48550/arXiv.1706.03762

[68] Y. Wang, Y. Wang, J. Chen, Y. Wang, J. Yang, T. Jiang, and J. He. Investigating the performance of gesture-based input for mid-air text entry in a virtual environment: A comparison of hand-up versus hand-down postures. Sensors, 21(5), 2021. doi: 10.3390/s21051582

[69] E. Whitmire, M. Jain, D. Jain, G. Nelson, R. Karkar, S. Patel, and M. Goel. Digitouch: Reconfigurable thumb-to-finger input and text entry on head-mounted displays. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 1(3), sep 2017. doi: 10.1145/3130978

[70] N. Yanagihara and B. Shizuki. Cubic keyboard for virtual reality. In Proc. SUI, p. 170. ACM, New York, NY, USA, 2018. doi: 10.1145/3267782.3274687

[71] J. Yang, H. Wang, and K. Guo. Natural language word prediction model based on multi-window convolution and residual network. IEEE Access, 8:188036–188043, 2020. doi: 10.1109/ACCESS.2020.3031200

[72] C. Yildirim and E. Osborne. Text entry in virtual reality: A comparison of 2D and 3D keyboard layouts. In Proc. HCII, p. 450–460. Springer-Verlag, Berlin, Heidelberg, 2020. doi: 10.1007/978-3-030-59990-4_33

[73] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. Shi. Tap, dwell or gesture? Exploring head-based text entry techniques for HMDs. In Proc. CHI, p. 4479–4488. ACM, New York, NY, USA, 2017. doi: 10.1145/3025453.3025964

[74] C. Yu, K. Sun, M. Zhong, X. Li, P. Zhao, and Y. Shi. One-dimensional handwriting: Inputting letters and words on smart glasses. In Proc. CHI, p. 71–82. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858542

[75] D. Yu, K. Fan, H. Zhang, D. Monteiro, W. Xu, and H.-N. Liang. PizzaText: Text entry for virtual reality systems using dual thumbsticks. IEEE Transactions on Visualization and Computer Graphics, 24(11):2927–2935, nov 2018. doi: 10.1109/TVCG.2018.2868581

[76] X. Zhang, X. Zhang, C. Yang, H. Yan, and X. Qiu. Does correction remain an problem for large language models? arXiv preprint arXiv:2308.01776, aug 2023. doi: 10.48550/arXiv.2308.01776

[77] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A survey of large language models. arXiv preprint arXiv:2303.18223, 2023. doi: 10.48550/arXiv.2303.18223