

# ABSubscribe: Rapid Exploration & Organization of Multiple Writing Variations in Human-AI Co-Writing Tasks using Large Language Models

Mohi Reza  
mohireza@cs.toronto.edu  
University of Toronto  
Toronto, Ontario, Canada

Peter Dushniku  
peter.dushniku@mail.utoronto.ca  
University of Toronto  
Toronto, Ontario, Canada

Tovi Grossman  
tovi@dgp.toronto.edu  
University of Toronto  
Toronto, Ontario, Canada

Nathan Laundry  
nathan.laundry@mail.utoronto.ca  
University of Toronto  
Toronto, Ontario, Canada

Michael Yu  
michaelz.yu@mail.utoronto.ca  
University of Toronto  
Toronto, Ontario, Canada

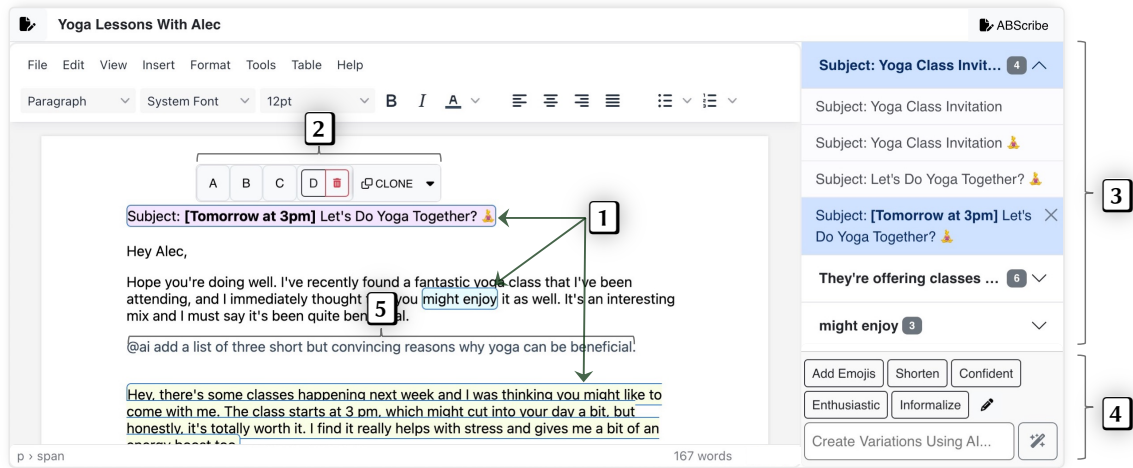
Michael Liut  
michael.liut@utoronto.ca  
University of Toronto Mississauga  
Mississauga, Ontario, Canada

Joseph Jay Williams  
williams@cs.toronto.edu  
University of Toronto  
Toronto, Ontario, Canada

Ilya Musabirov  
ilya.musabirov@mail.utoronto.ca  
University of Toronto  
Toronto, Ontario, Canada

Kashish Mittal  
kashish.mittal@mail.utoronto.ca  
University of Toronto  
Toronto, Ontario, Canada

Anastasia Kuzminykh  
anastasia.kuzminykh@utoronto.ca  
University of Toronto  
Toronto, Ontario, Canada



**Figure 1: The ABSubscribe Interface: (1) Variation Fields: Multiple variations are stored within text fields that do not break the flow of the draft. (2) Popup Toolbar: Users can swiftly choose and compare multiple variations by hovering over a dynamically placed toolbar located above the selected Variation Field, or clone and edit variations in-place. (3) Variation Sidebar: Users can navigate through multiple variations organized as an accordion-based interface. (4) AI Modifiers: Users can create variations using AI by typing modification instructions auto-converted into reusable buttons that can be applied to other Variation Fields. (5) AI Drafter: Users can draft text from GPT-4 directly into the document by typing '@ai <prompt>' and pressing enter.**

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CHI '24, May 11–16, 2024, Honolulu, HI, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0330-0/24/05...\$15.00  
<https://doi.org/10.1145/3613904.3641899>

## ABSTRACT

Exploring alternative ideas by rewriting text is integral to the writing process. State-of-the-art Large Language Models (LLMs) can simplify writing variation generation. However, current interfaces pose challenges for simultaneous consideration of multiple variations: creating new variations without overwriting text can be difficult, and pasting them sequentially can clutter documents, increasing workload and disrupting writers' flow. To tackle this, we present ABScribe, an interface that supports rapid, yet visually structured, exploration and organization of writing variations in human-AI co-writing tasks. With ABScribe, users can swiftly modify variations using LLM prompts, which are auto-converted into reusable buttons. Variations are stored adjacently within text fields for rapid in-place comparisons using mouse-over interactions on a popup toolbar. Our user study with 12 writers shows that ABScribe significantly reduces task workload ( $d = 1.20, p < 0.001$ ), enhances user perceptions of the revision process ( $d = 2.41, p < 0.001$ ) compared to a popular baseline workflow, and provides insights into how writers explore variations using LLMs.

## CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; *Empirical studies in HCI*; • **Computing methodologies** → **Natural language processing**.

## KEYWORDS

datasets, neural networks, gaze detection, text tagging

### ACM Reference Format:

Mohi Reza, Nathan Laundry, Ilya Musabirov, Peter Dushniku, Michael Yu, Kashish Mittal, Tovi Grossman, Michael Liut, Anastasia Kuzminykh, and Joseph Jay Williams. 2024. ABScribe: Rapid Exploration & Organization of Multiple Writing Variations in Human-AI Co-Writing Tasks using Large Language Models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA*. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3613904.3641899>

## 1 INTRODUCTION

*“The only kind of writing is rewriting”* – Ernest Hemingway, *A Moveable Feast* [42]

Revision is essential to the writing process [29, 42, 62, 101]. Professional writers often write and rewrite text hundreds of times [12, 87] and recommend rewriting as a core strategy for writing well [12, 101]. Effective revision goes beyond minor editorial changes, and may help writers rework ideas, and powerfully affect their knowledge [23, 87] as they explore alternative variations to find a line of argument [23]. The revision process is *iterative* [20, 23]: happens in repeated cycles, throughout the writing process, *granular* [62, 81]: happens at the word, sentence, or paragraph-level, and *non-linear* [13, 87]: requires constant reconsideration of potential variations of existing text throughout the passage.

Current writing interfaces tend not to support a non-linear revision process and predominantly support linear representations of revision history (e.g., revision history in popular word processing software such as Google docs or Microsoft Word). While these tools do support iterative and granular edits, it remains difficult for writers to simultaneously consider multiple writing variations and

to organize them without replacing earlier text, or cluttering documents when writers resort to pasting them in sequence. Insights from HCI and traditional design practice suggest that simultaneous consideration of multiple (at least 5) variations can lead to better ideas [89] and avoid fixation [10, 34, 44, 89]. We hypothesize that this may apply to writing, provided that writers are given adequate support in managing multiple variations with minimal workload.

Advanced Large Language Models (LLMs) such as ChatGPT [68], GPT-4 [69], PaLM 2 [32], LLaMA 2 [90], and Gemini[73] can enable writers to generate multiple variations of text via natural language prompting [8, 15, 26, 66], potentially reducing the workload of generating text variations. However, easier generation can exacerbate challenges surrounding the systematic organization, comparison, and modification of multiple variations with existing chat-based and in-place editing interfaces where users are required to find text variations in linear chat histories and store them in their document as comments or separate in-line text blocks. As text variations become easier to generate using AI, they become harder to manage.

Recent HCI studies on LLM-based tool design have mainly focused on prompt engineering [6, 16, 46, 55, 94, 98] and exploring the generative capabilities of LLMs [50, 58, 63, 67, 97]. For example, Zamfirescu-Pereira et al. investigated ways to support non-AI experts with crafting effective prompts, and Dang et al. [16] found that users preferred choosing from multiple suggestions and using content from the draft to guide LLM usage rather than writing external prompts which require more effort. Yuan et al. [97] explored how users might use LLMs for creative writing and found that the output of the model did not need to be perfect to be useful to users. Many users found the output useful, even if they had to significantly revise the text or chose not to incorporate it into their final draft [97]. This highlights the potential value of designing affordances that help manage *imperfect* AI-generated variations. Even if these variations don't make it into the final text, they might still be valuable to consider.

In this paper, we present ABScribe<sup>1</sup>—a novel writing interface that supports the rapid exploration and organization of multiple writing variations in LLM-based human-AI co-writing tasks. We draw inspiration from a design framework by Kim et al. which shows the potential for object-oriented interactions with LLMs to encourage iteration and experimentation during writing [47], and propose an ensemble of five interface elements that support writers in swiftly working with multiple writing variations: (i) **Variation Fields** that store multiple human and AI-generated variations within flexible text segments in a non-linear manner, without overwriting text; the (ii) **Popup Toolbar** that reveal corresponding variations inside a Variation Field when users hover their mouse over buttons representing each variation, allowing for rapid comparisons without breaking text flow; (iii) the **Variation Sidebar** that organizes all variations in a navigable format within an accordion UI; (iv) **AI Modifiers** that automatically encapsulates LLM instructions into reusable buttons that can be applied across different Variation Fields; and (v) the **AI Drafter** which allows writers to insert LLM-generated text directly into the document (see Figure 1).

<sup>1</sup>We name our system *ABSCRIBE* to reference how we label variations using the alphabet. Note that the interface supports variations beyond just A and B. The Popup Toolbar can include multiple variations: A, B, C, D, E, etc.

To validate our design, we conducted a controlled evaluation study involving revision tasks and interviews with 12 writers comparing ABScribe with a widely-used baseline workflow consisting of an AI integrated rich text editor based on GPT-4, with a chat-based conversational AI assistant. Our findings demonstrate that ABScribe significantly reduces subjective task workload ( $d = 1.20, p < 0.001$ ), and enhances user perceptions of the revision process ( $d = 2.41, p < 0.001$ ), compared to the baseline. The key contributions of our work are as follows:

- (1) The design and implementation of ABScribe, an LLM-enhanced writing interface that supports the rapid exploration and organization of multiple text variations in human-AI co-writing tasks. We have made the full code for ABScribe open source and accessible via GitHub<sup>2</sup>.
- (2) The results of a 12-participant user study with writers demonstrating the efficacy of the ABScribe interface ensemble and its advantages over a commonly used baseline workflow, and user perspectives on how writers explore multiple variations in human-AI co-writing tasks using a linear and non-linear revision process.

## 2 RELATED WORK

We first review literature in HCI and traditional design practices on considering multiple alternatives, and discuss how this design approach is pertinent when revising text, informed by the theory on the revision process in writing (section 2.1). Then, we examine this work's alignment with the rapidly growing research in AI-assisted writing, focusing on the effects of exposing writers to diverse AI-generated suggestions, and explore some challenges that arise when trying to support multiple variation editing using existing interfaces (section 2.2). Finally, we contrast between *chat-based* and *in-place* interfaces to situate our design within a broader class of Human-AI writing interfaces (section 2.3).

### 2.1 Exploring Multiple Variations

HCI and traditional design practices encourage the parallel exploration of multiple variations to help avoid fixation on a singular idea [30, 44], to reduce the chances of eliminating rough but innovative ideas due to premature evaluation [10, 34], and to make us less prone to inflated subjective appraisals by giving us an opportunity to critically assess ideas in relation to each other [10, 89]. In this paper, we hypothesize that such parallel exploration of multiple variations may apply to the revision process during writing. Much like how a naive, linear implementation of an iterative design approach encourages the sequential refinement of ideas, when writers do not have a way to organize and work with multiple text variations, they may end up committing to ideas too early, and focusing too much on surface level edits to refine their draft.

This is problematic because when we turn to research on the revision and writing process, we see that revision goes beyond surface level edits [23], encompassing deeper writing subprocesses such as revising and evaluating ideas [24] and meaning discovery [87]. Experienced writers treat revision as a recursive, non-linear process [81, 87], and engage with the text in repeated cycles, with

multiple objectives including finding the form or shape of an argument [87], experimenting with vocabulary and style [43], and going back and forth between multiple composing activities during revision [22].

In addition to the rich-body of work underscoring the important and complex role of revision in writing, researchers have explored the benefits of adopting design language in writing pedagogy, such as characterizing writing pedagogy as a wicked [78] design thinking problem [52, 75]. There has also been some valuable work in HCI in designing novel editing practices, such as supporting constraints and consistency in maintaining domain-specific terms across complex documents [36] using persistent, reified [3] text selections that allow writers to store and compare two variations. However, further innovation in this space is needed to design affordances for writers that support the simultaneous consideration of multiple variations during the revision process.

We contribute to this line of work, and present a suite of interface elements that work together in supporting writers with the rapid exploration and organization of multiple text variations in a non-linear fashion, in alignment with the nature of the revision process, and offer empirical insights into the applicability of design ideas in HCI on the parallel consideration of variations to the specific task of revision in writing.

### 2.2 Working with Multiple Variations from Large Language Models

There has been an explosion of recent work [4, 9, 14, 16, 26–28, 31, 47, 48, 50, 56, 67, 82, 84, 97] on interfaces for writing with LLMs in tandem with rapid advancements in AI writing capabilities using LLMs [8, 18, 66, 76, 91], examining various aspects of the AI-assisted writing process including prompting [28, 94], text suggestions [9, 16], creative ideation [14, 26], and social dynamics [4, 27]. Among these, prior studies on the impact of showing multiple text suggestions from the AI [9, 16] are of particular relevance to our design. Buschek et al. [9] highlighted the benefits of multiple parallel suggestions for ideation but warned against the simultaneous cost to efficiency. Dang et al. [16] coined and distinguished between *diegetic* prompts that are part of draft with *non-diegetic* prompts that are external instructions to the LLM. They uncovered user behavioural patterns indicating a tendency to choose from multiple suggestions rather than writing non-diegetic prompts as that requires more effort, and found that users prefer to use the draft, i.e., diegetic information, to guide LLM prompts.

AI Modifiers, for instance, simplify prompt-writing by turning modification instructions into reusable buttons, working closely with Variation Fields to offer users multiple options and utilize diegetic information directly from the draft. We also incorporate Kim et al.'s concepts of object-oriented interactions with LLMs [37] and draw from existing revision control research [35, 36, 41] to create interface elements that effectively manage multiple LLM variations, reducing task workload and facilitating parallel exploration.

<sup>2</sup>Link to GitHub Repository: <https://github.com/mohireza/abscribe/>

## 2.3 Chat-Based and In-Place Human-AI Co-Writing Interfaces

To ground our interface design, we distinguish between two types of Human-AI Co-Writing interfaces: Chat-Based interfaces such as ChatGPT and In-Place interfaces that directly insert or modify text in a document.

*Chat-Based Interfaces:* Currently the dominant mode, chat-based interfaces, like ChatGPT [68], Bing Chat [65], and Bard [72], have gained immense popularity. These interfaces are highly intuitive, and mimic conversational interactions between humans, but lack scaffolding for crafting prompts, which can be difficult for novice AI users [49, 98]. In the context of writing with AI, simple turn-taking between the writer and the AI does not provide adequate support to writers in steering the iterative output from the LLM [14].

Another significant limitation is the linear chat-log structure. In contrast to the non-linear nature of how revision happens in writing [23, 87], the text-variations generated using a chat-based interface are buried within linear chat-logs, impeding parallel exploration of multiple variations in-place, where the writer is editing the document.

*In-Place Editing Interfaces:* This type offers closer integration between the human and AI writer during the text editing process by adopting a more *What You See Is What You Get* (WYSIWYG) [5] approach where AI-generated text modifies the human text and vice versa. This offers increased flexibility over the edited content compared to chat-based interfaces by allowing users to edit individual sections of the text.

Recent research prototypes like Wordcraft [97] and CoAuthor [50], as well as commercial tools such as Grammarly<sup>3</sup>, Notion AI, and Wordtune<sup>4</sup>, offer in-place editing features for LLM writing assistance. Wordcraft [97] combines an editing interface with options for narrative continuation and AI-modified text replacement, while CoAuthor [50] leverages GPT-3 for deep interaction and multiple edit suggestions. However, these tools typically overwrite previous text with new edits, obscuring the original content. Saved versions are often linear, as seen in Google Docs, or limited to undo/redo histories, complicating the management of multiple text variations simultaneously.

In our design, we adopt an in-place editing interface in a GPT-4 powered research prototype, offering a solution to overcome challenges surrounding the management of multiple text variations in human-AI co-writing tasks. We provide fresh empirical insights from interviews with writers on how our in-place interface design compares with existing chat-based writing workflows, and their impact on user perceptions of the revision process and task workload.

## 3 DESIGNING ABSCRIBE

In this section, we describe the design requirements for ABScribe and the interface elements that we developed to address those requirements.

## 3.1 Design Requirements

We surveyed literature on several key areas critical to our goal of facilitating the swift exploration of multiple writing variations using LLMs: the role and nuances of revision within the writing process [23, 25, 38, 57, 83, 87]; HCI design philosophies that emphasize the consideration of multiple ideas before evaluation [19, 34, 89]; principles on reification and reuse for designing visual interfaces [3, 37]; and the latest research on utilizing LLMs in writing interfaces to foster experimentation and creativity [9, 14, 16, 26, 47, 50, 97]. Based on this, we formulated an initial set of design requirements, which are summarized below.

**Requirement 1: Minimize task workload when exploring multiple text variations.** Drawing from HCI and traditional design principles, we emphasize the importance of exploring multiple ideas concurrently. Instead of refining a single solution to “get the design right”, these principles encourage the iteration and evaluation of multiple solutions in parallel to ultimately “get the right design” [89]. We hypothesized that parallel exploration of text segments could aid writers during the revision process, in alignment with recent studies on the benefit of showing multiple parallel phrase suggestions on ideation [9]. We also considered that during the writing process writers frequently struggle with cognitive overload [61] and that even small demands on working memory can lead to decreased fluency [77]. With this in mind, we hypothesized that an increase in writing variations could worsen this. As such, we aimed to design an editing interface that provides affordances for creating and comparing multiple writing variations without overwhelming the user.

**Requirement 2: Support visually-structured management of variations.** Documents can quickly become cluttered when trying to work with multiple variations of different text segments using current editing interfaces. Furthermore, the potential for LLMs to enhance writers’ ability to generate and revise multiple parallel variations of text further exacerbates the issue of clutter. Together, these factors highlight the need for a visually structured approach to managing variations. Our goal was to support writers in seamlessly incorporating LLM-generated variations into the draft without cluttering the document or erasing existing content, and to retain the ability to simultaneously consider multiple variations. This is in alignment with prior studies that show that while users prefer being able to choose from multiple options [16], having all of these options can negatively impact writing efficiency [9], indicating a need for option management support. Furthermore, having the ability to choose between options in an intuitive way can give users an opportunity to make more informed decisions as they compare and contrast between parallel ideas [10, 34, 89].

**Requirement 3: Support context-sensitive variation comparison and revision.** In a typical, linear document editing interface, we found it difficult to maintain a sense of the surrounding text as we created and added variations. This was particularly noticeable when trying to create and compare variations for smaller and embedded text segments—e.g., a sentence mid-paragraph or paragraph mid-section—which disrupted the text flow. Maintaining text flow is crucial since writers need to engage with information processing tasks such as ensuring the document maintains cohesion, which requires matching segments to the surrounding text

<sup>3</sup>grammarly.com

<sup>4</sup>wordtune.com

[60]. This becomes increasingly challenging as the document holds more and more variations of different text segments. Our objective was to design an interface that allows writers to systematically generate and edit multiple variations in context of the preceding and subsequent text in the document.

**Requirement 4: Supports revision-centric, reusable, and non-linear LLM usage.** Recognizing that revision is inherently nonlinear—with writers often revisiting earlier sections of a passage—and recursive, manifesting in repeated cycles throughout the writing process [23, 87], we aimed to align our LLM integrations with this fluid, iterative nature of revision. Our goal was for writers to be able to use LLMs to manipulate text segments of varying lengths and refine them as needed in a way that is natural to their non-linear and recursive process. To enable this, we drew inspiration from design principles for visual interfaces, focusing on reuse, polymorphism, and reification [3], and regarded LLM prompts as reusable, polymorphic commands that can be applied to targeted text segments of varying lengths. Recent research on designing LLM-powered writing interfaces has highlighted the value of viewing components of the LLM generation pipeline as interactive objects in supporting iteration and experimentation [47]. We adopt aspects of this approach in our design, such as reifying [3] LLM prompts into reusable AI Modifiers, and turning text segments into interactive Variation Fields akin to *cells* [47].

## 3.2 Interface Elements

We addressed these four design requirements by developing five interface elements using an iterative design process. To get the right design [89] the lead author iterated through multiple versions of each interface element, and tested them with five pilot users during in-depth brainstorming and design-review sessions over six months.

**3.2.1 Design Process.** The design process involved transitioning from low to high-fidelity prototyping via three overlapping processes: (i) *Paper-Prototyping* [80, 86]: We began by considering an in-place editing interface as a foundational concept, and sketched various design elements that could be integrated into this interface to enhance exploration of variations; (ii) *Cognitive Walkthroughs* [53, 59]: We used informal cognitive walkthroughs to rapidly iterate on our sketches and identify which potential solutions were intuitive; (iii) *Web-based Prototyping* [92]: We then developed functional versions of our intuitive design solutions in a high-fidelity prototype. There was considerable iteration between these stages. For example, if we saw issues in the high-fidelity, we would return to paper prototyping to devise refinements. Through this process, we developed five interface elements:

**3.2.2 Variation Fields.** Users can select any part of the text and create interactive Variation Fields that can hold multiple writing variations (Figure 2, Part 1). Newer variations can be added to an Variation Field without overwriting existing variations (Figure 2 Parts 2 and 3) and without breaking the flow of the passage.

**3.2.3 Popup Toolbar.** Variations are represented using buttons residing within a Popup Toolbar placed above the active Variation Field. The Popup Toolbar dynamically moves with the text caret,

and users can peek at different variations by hovering over the corresponding buttons to view them in the context of the surrounding passage (Figure 2, Parts 4 and 5). Moving the cursor away from a toolbar button reverts the Variation Field back to the selected variation, allowing users to quickly compare between the selected and the hovered variations. Users can click on a button to select the variation and edit them in place, or discard it by clicking on the trash icon (Figure 2, Part 6).

**3.2.4 Variation Sidebar.** To help writers view multiple text variations together, and navigate through them more easily, we pair the Popup Toolbar with a sidebar organized in an accordion structure where each Variation Field has its own header, and all corresponding variations are stored underneath. Clicking on the variations in the sidebar dynamically re-positions the Popup Toolbar above the active Variation Field, and vice versa, allowing users to navigate through multiple variations in a visually structured manner (Figure 3). Users can show or hide the sidebar from the editor menu.

The Variation Fields, Popup Toolbar, and the Variation Sidebar work together to address R1, R2 and R3. Note that the Popup Toolbar and Variation Fields have no AI capability, but support text revision by the user. To tackle R4, we developed two interface elements.

**3.2.5 AI Modifiers.** Users can generate new variations by selecting a Variation Field and typing instructions to the AI (Figure 4, Part 1 and 2). Instructions are automatically converted into labeled buttons (Figure 4, Part 4). The labels are generated using the LLM. As users experiment with newer variations using the AI, they create a set of custom AI Modifiers that they can apply to different parts of the passage, making them reusable (Figure 4, Part 5 and 6). The prompts and labels for the buttons can be edited and improved over time (Figure 4, Part 7 and 8). This allows writers to not only create variations for a particular Variation Field, but also design buttons reflecting the kinds of variations they might want to generate for other parts of the text in the future, akin to a custom Swiss army knife for variations. As users add more variations, the interface organizes them into a scrollable list with a maximum height, so users do not run out of room. A more complete implementation could explore more sophisticated ways to group and search through AI Modifiers.

**3.2.6 AI Drafter.** Users can insert text from the AI model anywhere within the passage by prompting the system to invoke LLM usage in the following format: @ai <prompt>, and pressing enter. The text is generated and shown to the user in real-time, and they have the option to accept or discard the AI generated output, as well as revise the prompt to regenerate the output if it does not match what the user is looking for (Figure 5).

## 4 EVALUATING ABSCRIBE

To validate our design, we conducted a within-subjects evaluation study where we compared ABScribe to a carefully constructed baseline interface. Refer to Appendix A to see a screenshot. The Baseline interface featured the same rich-text editing capabilities as ABScribe without the ability to create and modify Variation Fields using the Popup Toolbar, Variation Sidebar and AI Modifiers. Instead, it featured a conversational AI assistant similar to ChatGPT, and the ability to insert AI generated text directly into the document

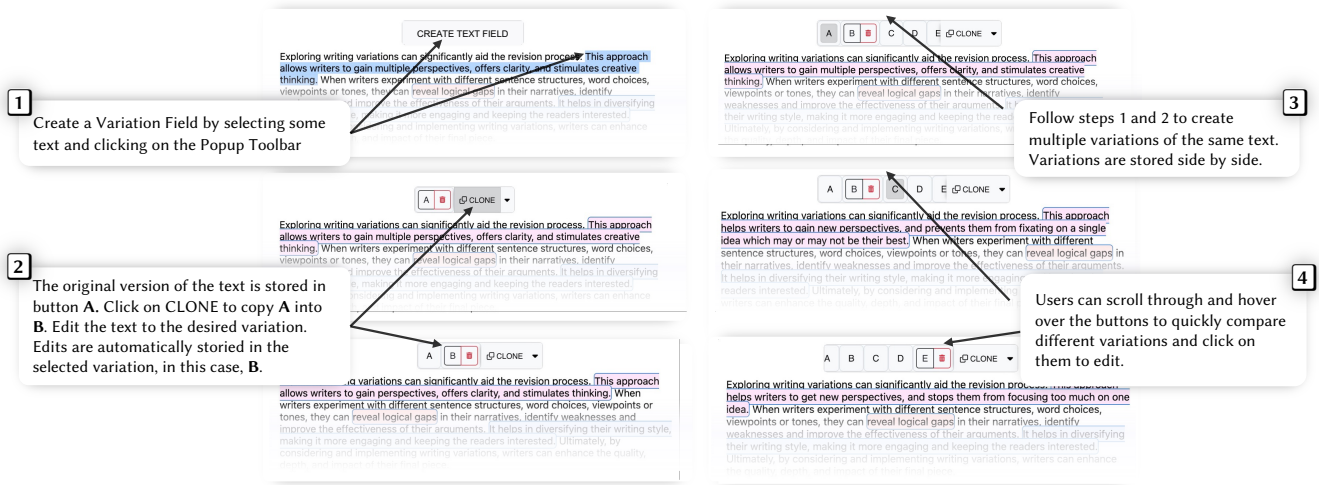


Figure 2: Variation Fields & Popup Toolbar: Users can choose, clone and edit multiple variations stored inside flexible text fields.

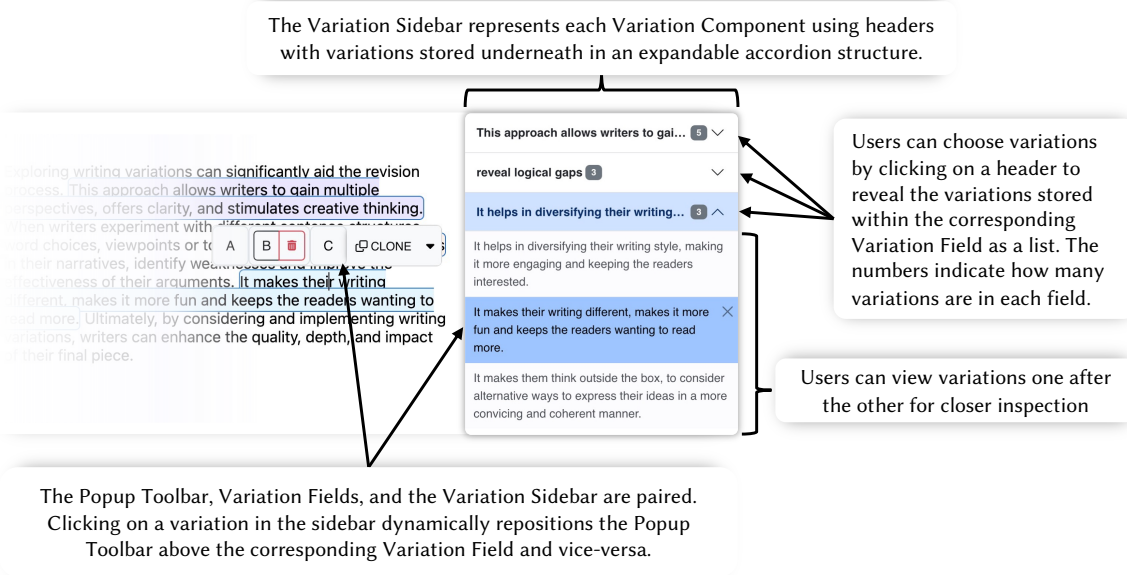
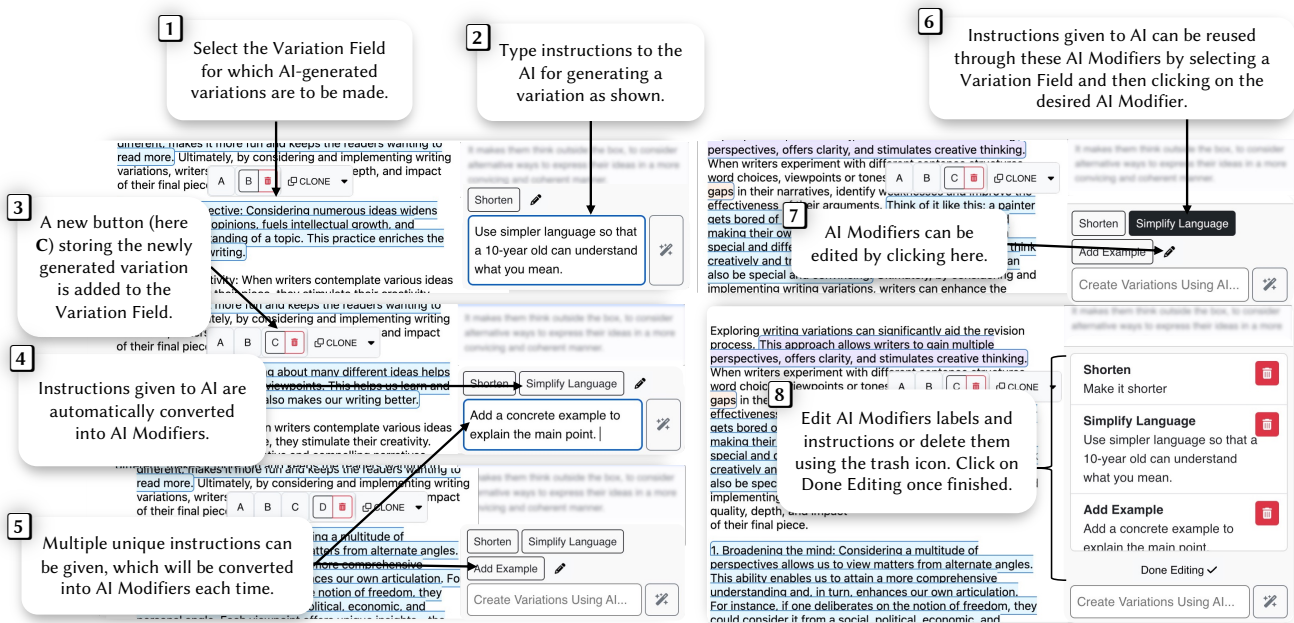


Figure 3: Variation Sidebar: Users can select, view, and navigate through multiple variations using an alternative accordion structure.

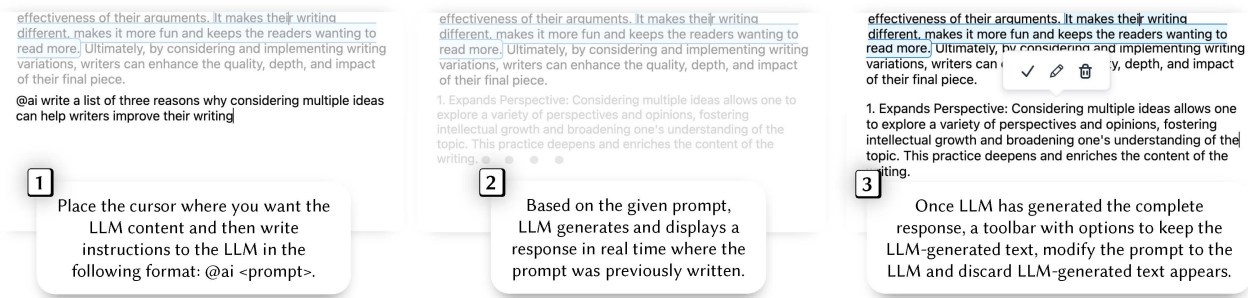
as some modern AI editors such as Notion AI<sup>5</sup> have that capability. To minimize potential confounding factors arising from tangential differences between the study conditions, we maintained the same overall layout for common UI elements. This included the width of the sidebar, placement and dimensions of buttons and text fields, font size, and color. Additionally, we used the same underlying LLM model, GPT-4, to implement the generative AI features for both conditions.

<sup>5</sup>notion.ai

- RQ1:** How does ABScribe influence **user perceptions of the revision process** for AI-assisted exploration of writing variations when compared to the AI-integrated Baseline interface?
- RQ2:** How does ABScribe influence **subjective task workload** for AI-assisted exploration of writing variations during text revision when compared to the AI-integrated Baseline interface?



**Figure 4: AI Modifiers:** Variation Fields can also be edited using the AI Modifiers, which let users specify custom alterations. Descriptive labels are automatically generated for each AI Modifier to turn them into reusable buttons.



**Figure 5: AI Drafter:** Users can draft LLM-generated text directly into the document, providing tighter integration between the Human and AI generated writing workflow. Users can see the AI generated content in real-time and choose to insert or delete the output, or revise the prompt, giving them control over what is included in their document.

### 4.1 Participants

We recruited 12 writers (5 women, 7 men, aged 18 to 34) through flyers and study invitation messages on social networking services, such as LinkedIn and Facebook. All participants reported proficiency in reading and writing in English and were screened for prior experience in a broad variety of both fiction and non-fiction writing genres. We included participants with a range of prior experience levels in AI tool usage. This ensured that our findings were not limited to specific genres or AI usage patterns. All participants had ample writing experience, enabling them to effectively

comment on the revision process and its relevance across various writing forms. Refer to Table 1 for writer profiles. Participants were compensated with 30 CAD. The institution’s research ethics board has approved the protocol.

### 4.2 Tasks

Each participant engaged in two guided writing tasks which were randomly paired with the two counter-balanced study conditions. Our choice of tasks—writing an email and a social media post—aimed to provide an ecologically valid writing experience that fits the

ID	Writing Experience	AI-Assisted Tools Usage
W1	<b>Moderate:</b> Worked as a staff writer for two political science publications. Writes fiction as a hobby.	<b>Moderate:</b> Uses ChatGPT to edit writing projects, as well as receive feedback and suggestions for further passages, primarily after completing a passage to identify areas for further revision.
W2	<b>Moderate:</b> Taught ESL courses to non-native English speakers, specializing in IELTS, TOEFL, and business English instruction.	<b>Experienced:</b> Uses ChatGPT by feeding it the main points of the article to generate a draft, and then editing the responses provided by ChatGPT.
W3	<b>Highly Experienced:</b> Has experience writing papers, specifically about writing tools for HCI. Has also published a novel, and has publications in many highly regarded literary venues (BOMB, LitHub, FENCE, and more). Participant also writes their own music.	<b>Moderate:</b> Uses ChatGPT for drafting messages, seeking feedback on fiction, and drafting small sections of research papers. Has experience with Respondable, a service in the Gmail plugin called Boomerang, for writing emails using AI.
W4	<b>Highly Experienced:</b> Writes fiction and published one novel, some sci-fi and fantasy short stories, and several articles for blogs, magazines, and satirical news sites. Worked as a staff writer as an undergraduate, a professional screenwriter for two independent studios. Also teaches two first-year writing classes in a liberal arts college. Achieved a MFA in Creative Writing.	<b>Limited:</b> Briefly experimented with ChatGPT to test its capabilities by asking it to write some scripts, essays, and articles. Found the results to be amusing, but lacking in perspective and personality.
W5	<b>Experienced:</b> Writes content for social media profiles for an NGO. Studied English Literature during both bachelor's and master's degrees. Writes music, having penned 65 songs, and promotes it through social media and music platforms such as YouTube and Spotify.	<b>Experienced:</b> Used Grammarly for on-the-go editing to write and ChatGPT much more extensively for both idea generation, as well as summation and synthesis of large bodies of text. Also found ChatGPT useful for helping figure out parts of creative works that may feel like they have gaps which can be prone to miss.
W6	<b>Experienced:</b> Worked for two national English language newspapers, including contributions to their weekend magazines, kid's sections, international section, in addition to also publishing fictional short stories for the newspaper. Completed a Creative Writing Certificate, and currently primarily writes about research.	<b>Limited:</b> Used ChaptGPT in a very limited capacity, mostly to brainstorm assignment structures and topic sentences when writing.
W7	<b>Experienced:</b> Writes academic articles, general interest articles and reviews for local newspapers. Also writes short stories for sharing with friends, and has a short story published in an locally published anthology. Expertise is primarily in creative nonfiction.	<b>Limited:</b> Briefly experimented with ChatGPT.
W8	<b>Moderate:</b> Written mostly technical papers, but also wrote some short stories as a hobby.	<b>Moderate:</b> Uses ChatGPT every other day mostly to proofread, create templates for texts, and find the right creative direction when writing.
W9	<b>Experienced:</b> Usually writes research papers in computing education and blogs. Blogs usually cover personal experiences at work as well as hobbies.	<b>Limited:</b> Used Grammarly and ChatGPT to edit writing.
W10	<b>Moderate:</b> Wrote some column articles for personal social media accounts, and several research papers over the past five years.	<b>Experienced:</b> Uses Notion.AI, ChatGPT, and GPT-4. Uses Notion.AI for generating bullet points and brainstorming ideas, ChatGPT for generating templates for writing, and sometimes summarizing related work for research purposes.
W11	<b>Highly Experienced:</b> Focuses on academic writing such as papers, scholarship applications, reviewing, etc.	<b>Advanced:</b> Has experience with Grammarly, Notion, Obsidian with GPT plugins, and ChatGPT. Mostly uses these tools to clean up sentences, and sometimes uses them to brainstorm titles for papers.
W12	<b>Experienced:</b> Engages in hobby novel writing, academic writing, blog writing	<b>None:</b> Has not used AI-assisted writing tools, but has experience with AI image generation using written prompts.

**Table 1: Self-reported experience with writing and using AI-assisted tools. Expertise labels for writing range from very limited to highly experienced, and labels for the AI tools ranges from none to advanced. Details on prior writing experience and AI tool usage is also included for each participant.**

timing constraints of our study and served as realistic use-cases for LLMs, in alignment with recent HCI studies on human-AI co-writing [9, 31, 47], and commonly advertised applications of commercial AI-assisted writing tools, such as Grammarly Go, Respondable, and Copy.ai. We selected scenarios by considering situations that: (i) would be easy for users to imagine, such as emailing a professor or seeking a job as a copywriter; (ii) offer opportunities for exploring variations, such as devising multiple alternatives for a subject line or altering a sentence to maximize reader engagement. We wrote and tested the prompts to ensure the output was reasonable and of consistent length. For example, after initial rounds of testing, we found that the output generated by the AI was too long, and so we added 'Keep it within three paragraphs' to both

prompts. We asked participants to generate a draft using the same prompts for all scenarios. This approach was designed to keep their focus on exploring and organizing variations, rather than engaging in prompt engineering on the initial draft. It also helped maintain relative consistency in the generated output across participants and study conditions. Refer to Appendix B for additional details on the task scenario descriptions and prompts.

Then, with either ABSubscribe or the Baseline interface, participants were asked to use AI to explore eight variations (increasing or decreasing length, formality, word diversity, adding emojis, and two variations of their choosing) of three distinct text segments (title/subjectline, third sentence of the second paragraph, entire



third paragraph), summing up to twenty-four variations. This approach ensured variation exploration of consistent number, size and variety across study conditions, while affording some scope for creativity as, for two out of the eight variations for each segment, participants had the autonomy to craft variations based on their preferences. We selected the variations by considering potential modifications writers might want to apply to their drafts, including changes in length, tone, and word choice. We then tested these modifications to ensure their feasibility with both ABScribe and the baseline interface.

### 4.3 Measures

To assess subjective task-workload, we used the widely used NASA-TLX [39] procedure with weighting. To quantify the specific aspects of the LLM-assisted revision process that we aimed to improve, we also asked participants to rate their agreement on a 7-point Likert scale (1 = Strongly Disagree, 7 = Strongly Agree), similar to prior work [47, 50, 51]:

- (1) **Variation Granularity:** *I felt like I could work with multiple (more than 5) writing variations of different fine grained parts of the text (e.g. word, sentence, paragraph) using this tool.*
- (2) **Variation Search:** *I felt like after creating all these variations, I could find previous variations when I needed them (e.g. when trying to create a new variation based on an existing variation I created earlier in the writing process).*
- (3) **Prompt Reuse:** *I felt like after creating all these variations, I could reuse my previous instructions or prompts to the LLM without having to rewrite them often.*
- (4) **Variation Comparison:** *I felt like I could identify fine-grained differences between multiple variations using this tool.*
- (5) **Variation Editing:** *I felt like I could systematically edit new variations without losing existing variations or cluttering the document using this tool.*
- (6) **Variation Control:** *I felt like I had control over which variations I wanted to keep, discard or change.*
- (7) **Variation Divergence:** *I felt like exploring multiple variations using this workflow will help me come up with variations that are surprisingly different.*
- (8) **Draft Quality:** *I felt like exploring multiple variations using this workflow will help me have better final draft.*
- (9) **Intent Match:** *I felt like exploring multiple variations using this workflow will help me come up with variations that are closer to what I want to say.*
- (10) **Variation Diversity:** *I felt like I could create variations with a lot of variability in word choice, style, and tone of voice using this tool.*
- (11) **Document Clutter:** *I felt like after creating all these variations, the document became cluttered.*

### 4.4 Procedure

Participants began by signing a consent form and completing a survey that captured demographic data, prior writing experience, and familiarity with AI-assisted writing tools. The entire study lasted approximately 1.5 hours, and was conducted via video-conferencing software. Participants accessed the prototypes through their web browsers, mirroring how they typically access popular AI editing

tools like ChatGPT and Grammarly. Conducting the study online enabled us to engage a diverse group of writers beyond Canada. After a brief introduction outlining the study’s objective—to investigate the exploration and organization of multiple writing variations using LLM-integrated editing tools—participants undertook the two 15-minute writing tasks. Before starting each task, we demonstrated how the tools functioned and gave participants an opportunity to try them, ensuring they felt comfortable. After each task, participants completed the NASA-TLX assessment and 11 Likert-scale measures. These measures offered insights into the writers’ perceptions of the revision process and prompted them to reflect on specific aspects of the process that we sought to improve through our design. The evaluation concluded with a recorded 30-minute semi-structured user interview on their experience with each interface.

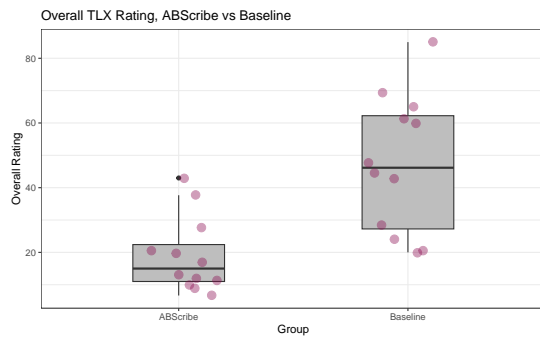
### 4.5 Analysis

Our data comprised interview transcripts, task observation notes, and the NASA-TLX and Likert-scale ratings for each condition. We coded and analyzed the interview transcripts and task observation notes using reflexive thematic analysis [7] through an inductive-deductive lens. The theory on revision-focused exploration of writing variations served as a pre-existing code guiding our interpretations. In a within-subjects design, we use pairwise one-sided t-tests to compare the sum of scores of NASA-TLX and our Likert-scale measures on the revision process. T-test was shown to be robust for aggregated data of this kind [17]. We aimed to determine if ABScribe presented significant improvements over the baseline interface, prompting us to select a one-tailed test with hypothesis  $B < A$  for task workload and  $B > A$  for level of agreement on the efficacy of the revision process. We performed an *a priori* power analysis for a pairwise one-sided t-test, showing that we can detect an effect size of at least  $d = 0.8$  with 80% power and a sample size of  $N = 12$  participants for a significance level of  $\alpha = 0.05$ .

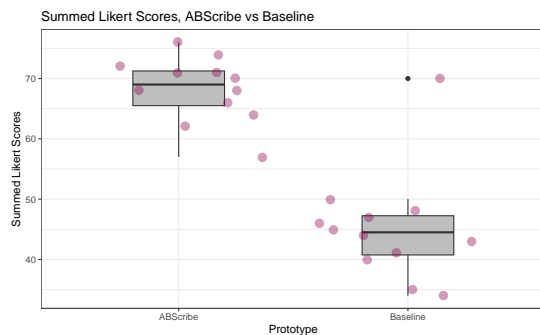
## 5 RESULTS

*“The user interface for A [ABScribe] made comparisons, easy storage and access of those variations much easier than B [Baseline]. The fact that after you wrote a prompt, it instantly assigned a button to it that you could access later, was incredibly useful. It made it such that you could actually play with a more precise number of variations than I previously could and the fact that you could manually edit them and then again, quickly, have a way to play with the variation made it much more practical as a writing tool, there was a lot less physical effort involved to streamline that process. It was a wonderful, very smart way of dealing with the problem of clutter on the page. Even absent the AI generation part of it’s an excellent way of storing variations.”*  
– W1

The overall response to ABScribe, as exemplified by W1’s comment and Figures 6, 7 and 8, was positive, with a significant increase ( $d = 2.41, p < 0.001$ ) in the summed agreement levels on the efficacy of the revision process (RQ1), and a significant reduction ( $d = 1.20, p < 0.001$ ) in TLX rating for subjective task workload



**Figure 6: Overall results on NASA TLX subjective task workload**



**Figure 7: Summed Likert ratings for users' perceptions of the revision process**

(RQ2) when compared to the Baseline interface. To gain deeper insights into the factors behind the reduction in task workload and the increase in user perceptions, we conducted a reflexive thematic analysis on the semi-structured user interviews. *User perceptions on the revision process* (section 5.1) groups findings relating to how the users perceived changes to the process and outcome of creating and managing variations as well as their interactions with the LLM. *Subjective task workload* (section 5.2) groups findings relating to the ease of specific tasks during the writing and revision process as well as the ease of specific interactions with the interface to accomplish those tasks.

### 5.1 RQ1: User Perceptions on the AI-Assisted Revision Process

ABSCRIBE provides features targeted at improving *Variation Management* and thoughtful use of AI in text revision. To disentangle results of the two as well as identify when findings are a result of both those features, findings in the following section are tagged. We use *Variation Management*, *AI Integration*, or *Mixed* to indicate whether a finding relates purely to the management of text variations, or how participants interact with the AI, or a mixture of both features. When a finding is *Mixed*, comments are provided to exemplify what users attributed to *Variation Management* elements, and what was due to *AI Integration* elements.

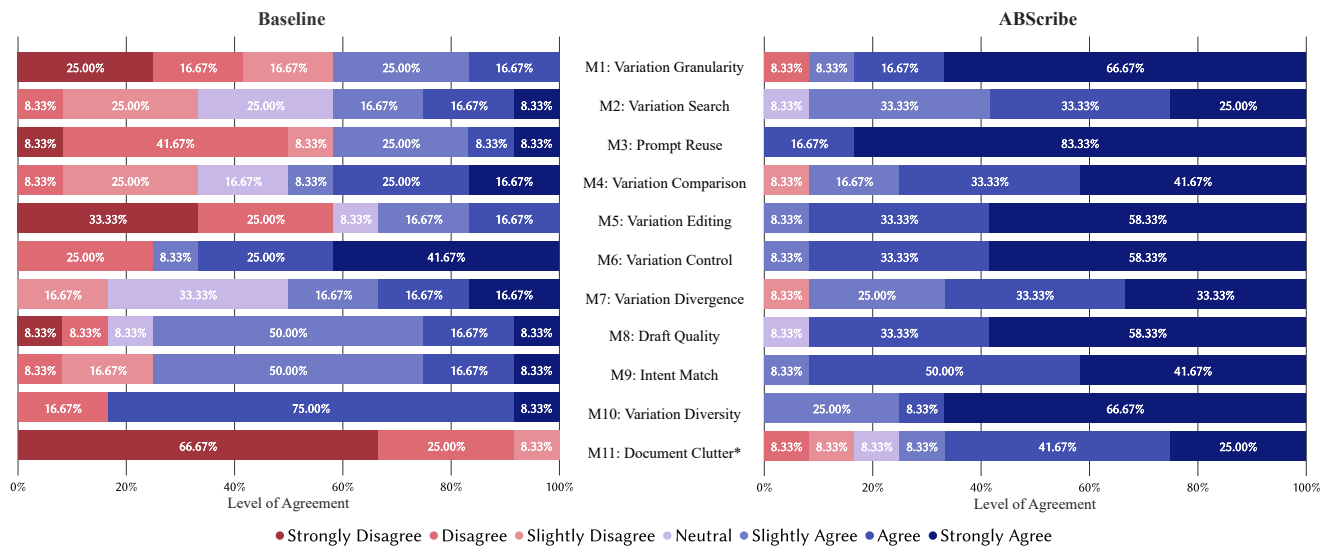
**F1: Variation Management**—ABSCRIBE lessens pressure to commit early to an initial idea and nudges users to explore a greater

quantity of variations than the Baseline workflow. The non-linear approach to *variation storage* within a *Variation Field* and the ability to interact with them via the *Popup Toolbar* and *Variation Sidebar* without cluttering the document made some participants feel less pressured to commit to an initial variation before considering multiple options. In contrast, participants indicated that in traditional text editing settings, they maintained and iterated on a single variation. During the study, we noticed baseline interactions favoured expansion of variations for a single passage, revision, then collapse to a single variation to avoid clutter, whereas with ABSCRIBE, multiple variations were maintained for considerations. For example, W3 notes how the difficulty of *variation storage* in Baseline pushed them to maintain just one variation: “I would probably feel more pressured to just kind of work on one sentence and come up with a couple of variations and change it immediately. I would feel like I have to commit pretty early on, rather than generating a number of variations, trying a bunch of different tracks and sort of different timelines, seeing how each of them turns out and performing a master comparison at the end. So I think it had a significant effect, or would have a significant effect on my behavior, certainly doing the same task for both conditions where I was trying to deal with a bunch of different versions and trying to change them and revise them differently. It was vastly more difficult in B and a nonlinear approach seems to make much more sense.”

W2, 3, and 12 specifically commented that the lack of interface clutter due to the way the *Variation Fields*, *Popup Toolbar*, and *Variation Sidebar* handle *variation storage* meant that they could create, revise and evaluate a larger number of variations with ABSCRIBE. W12 noted that *variation storage* in a linear flow is “just very clunky”, leading to a “higher cost of keeping multiple versions”, “more scrolling in the document”, and “taking a longer time to find anything specific”. While exploring variations with Baseline, W12 noted that: “I sort of gave up about halfway because it was taking too long to vaguely remember, oh there was this version that I thought was good, but I wasn’t able to find it because the document became so long, so I just grabbed whatever to just finish.”

**F2: Mixed**—ABSCRIBE enhances writers’ ability to explore more granular variations in context of the surrounding passage. W1, 3, and 9 mentioned that they could work with smaller text-segments more easily when using ABSCRIBE than Baseline. Participants indicated that this was due to both the *Popup Toolbar* enabling in-context comparison of variations (meaning they could switch between variations in-place observing how they would fit in to the passage) and the *AI Modifiers* allowing for less effortful use of AI on finer-grained text segments.

With regards to *variation management*, a concern participants presented when editing fine-grained text segments within traditional text editing interfaces is that the overall coherence of the passage would be more difficult to maintain while performing *variation comparison*. W9 said “I fear that if I edit small chunks, then the tone of different chunks end up becoming different. Whereas I kind of want to change the tone of the whole thing to one specific thing.” W3 shared a similar concern but noted that the ability to view and edit variations within the context of the surrounding passage reduced their concern: “ABSCRIBE is vastly superior for any kind of fine-grained edits which become incredibly difficult to deal with [in Baseline] if you’re trying to do different edits on smaller variations



\*Likert scores for M11 are reversed to account for negative valence of wording. Higher scores correspond to more positive user perceptions in each measure.

**Figure 8: Responses to Likert-Scale Measure on the Revision Process for Exploring Multiple Writing Variations. Higher the agreement level, the more positive the user perception.**

of the text, unless you want to perform a single edit and immediate make that change.”

With regards to AI Integration, When using the chat-based LLM interface in Baseline, W3 said they had “never even considered [editing] on a sentence level because it would be so hard to go into ChatGPT and say that in the third sentence of the third paragraph...I don’t even know if it [ChatGPT] has a sense of where that is.” This sentiment was echoed by W9 who noted: “Usually, [I] would be [editing] at least a whole paragraph and see edits from that and then paste some of those edits in [to the document] but I wouldn’t put one sentence in...so there isn’t very fine-grained control [in Baseline]...whereas here, [with ABScribe], because it takes less effort, I’m like okay, I can do one sentence.”

The combination of AI Modifiers and Popup Toolbar enhances writers’ ability to explore more granular variations by enabling the writer to remain within-context as they edit.

**F3: AI Integration–ABScribe nudges writers toward an imperative LLM prompt writing style in contrast to the conversational style in Baseline.** W1, 3, 6 and 8 noted how the automatically generated AI Modifiers that captured prompts they previously wrote influenced their prompt writing style to be more direct. W1 mentioned that in ABScribe, “you’re conscious of the fact that you are designing a button, and that forces you think within that framework. You’re not really asking someone to do something [like in the chat-based interface]. You’re giving it instructions to make a button...which make things very simple, very easy, very quick...making it much easier to create prompts, and then use them subsequently, when they’re being instantly converted to a new use-case.” Some examples of the short imperative style from the participants include “Make more professional, Make more casual, write this so a fifth grader could understand it” Note the imperative style and general nature of the prompts.

In contrast, the conversational approach sometimes led users to anthropomorphize the AI, using superfluous words that do not influence the quality of the variation generated. For example, we observed that W8 would say “Can you please make this variation shorter?” or “Could you make a version with more varied vocabulary” when instructing the chatbot, but opt for a more direct “make it shorter” prompt for ABScribe. When asked why, they said the conversational AI assistant felt “similar to Clippy”, referencing the Office Assistant from a discontinued intelligent interface from Microsoft with an interactive animated character [2].

**F4: AI Integration–ABScribe nudges writers toward composing generalizable and atomic LLM prompts in contrast to the complex and variation-specific prompts in the conversational approach in Baseline.** W3 and 12 shared that with ABScribe, they were intentionally trying to simplify their prompt design to make them generalizable and reusable across different text segments. W3 said “I felt more like I was going to create a generalized prompt, that I will probably reuse later. So it felt like it ought to be something that’s simple that could be applied to a variety of situations rather than something that’s specific to a single piece of text.” In contrast to W3, W9 found that due to the ease of prompt reuse, they were more likely to create longer prompts that would generalize to other contexts, when compared to the chat interface. “I was okay with writing longer prompts, for example, to imitate the style of a character because it took less effort, and it was fun for me to do, and I knew I could use it again in other sentences. Whereas in [Baseline], I wasn’t as excited because it would take more time to type, and I knew that if I had to reuse it, I would need to type it again.”

However, because participants were actively thinking of instructions they could reuse across different text segments, they were less likely to write prompts that were tied to specific characteristics of

particular text segments, as exemplified by W3's comment: "I categorically preferred A [ABSScribe] to B [Baseline]...the only advantage to B might be that it slightly encouraged me to have more nuance in the prompts I gave to ChatGPT...it just made me realize I could do that. I know I could do that in ABSScribe too, but the button interface kind of guided me to more naturally consider simplistic prompts rather than prompts which might be more suited to particular tasks for those in conversation with a bot." A prompt pattern in the conversational approach was to copy paste a section of text then ask for a modification such as the following by W3 "Comprehending the essence of crisp, contemporary material ..." Can you write this in a way that maximized the number of linkedin likes?"

With ABSScribe, participants tended to decompose specific prompts that only apply to a given text segment into more general, atomic prompts that apply across variations. To achieve similar specificity, participants combined multiple prompts by clicking selecting multiple AI Modifiers. W3 considered mimicking the functionality of longer, more complex prompts by "stacking" multiple of their general, more atomic prompts to explore variations: "I think I would also be more willing to try out a variety of permutations of different prompts, rather than trying to apply one prompt globally to an entire email, or full prompts to one or two sentences. So seeing how compound prompts might help at various points would be a lot better in A [ABSScribe]." W8 demonstrates this stacking of atomic prompts in the following use of the AI Drafter "@ai tell me a joke, make it more creative, personalize!, add a cough or a speech utterance between every few words" Note the use of several generic and atomic prompts "make it more creative" followed by "personalize!"

## 5.2 RQ2: Subjective Task Workload

Participants pointed to four aspects of the revision workload in ABSScribe that led to the significant reduction in subjective task workload: reduction in clutter, ease of *variation management* which referenced both *variation storage* and *variation comparison*, access to surrounding text context, and the reduction of context switching during LLM use.

**F5: Variation Management**—ABSScribe reduces task workload by reducing document clutter. W4, 5, 7, 10, 11, commented that the general lack of clutter in the ABSScribe interface when dealing with multiple versions affected various aspects of their writing and revision process. Specifically, the way ABSScribe handles *variation storage* in-place (i.e., all variations of a segment of text are stored and can be accessed at the *Variation Field* using the *Popup Toolbar*) as opposed to in-sequence, by storing both versions in the main text editing interface, reduced clutter. W11 summarizes this in the following quote "I think definitely the biggest difference would have been the fact that your your document is not as cluttered. Actually, it doesn't get cluttered basically because you can just switch between versions and the text is on in the same location. So that's already a huge boost in terms of not having a mess on my hand. And you notice at the very start I was already organized to think that way." When comparing ABSScribe to the linear storage of variations (one after the other within the main document) in Baseline, W5 and 8 noted that the linear interface inevitably led to cluttered and messy documents. W5 pointed out the difficulties of managing variations in the linear Baseline approach: "So the linear approach was more

difficult, because there was just walls of text, like they began piling up very quickly. And I tried to segment them, right, I believe, if I numbered them, it would have been better. But at the same time, it doesn't get rid of the root problem, where more and more text is, is being added to the whole draft."

A noteworthy and surprising trade-off is that although all participants agreed that the ABSScribe interface produced less clutter, two of them perceived clutter as unproblematic or even beneficial. W2 and W6, were both comfortable with the workflow afforded by Baseline as it was similar to what they were used to doing in their naturalistic writing tasks - paper, article, and book writing. They noted that clutter didn't matter as much to them during the revision process. W2 mentioned that Baseline "definitely" felt more familiar, and that "...you probably think cluttering is one of the one of the important factors to consider when people are writing, I actually don't think that's the case. That's why I don't care about whether it looks [messy during revision]." W2 also mentioned when doing *variation comparison*: "In terms of comparison, I think this is just, I don't know, it just, it looks convenient. But actually, when you're actually writing things that matter you don't want to just use this to compare the sentences. ... For example, I guess most people use ChatGPT to write emails, that's the most popular application, when they write emails to clients and professors, people—they don't want to offend them. But sometimes you want to pay attention to your wording and stuff, in that case, I don't think anyone wants to just compare like this. You want to be careful." and concluded with "I would not compare them on this page [via the *Popup Toolbar* and *hover feature*], I would compare them in the *Idea Bucket* [*Variation Sidebar*]."

W6, who described themselves as a "a messy editor", a "hoarder" of various made copies of older text, felt "really conflicted" about the reduction in clutter. They commented: "I feel like it would be a little bit easier to do that [variation comparison], because the text is already there [in the main document editor] ... you can't see two versions side by side unless you were to scroll in the box [*Variation Sidebar*], right. And so I feel like version B would be just like, it's easier there to do it. [variation comparison]" and that they liked being able to "mishmash multiple versions together" in a messy document. In describing the workflow in Baseline, they said it was "more familiar to me than like doing it the way that you would in ABSScribe. Even though in an abstract way, the non-linear approach makes a lot of sense...I feel like it just feels like that's how the design should be...but I feel like [I'm] a messy editor. And so it's, it's almost easier for me to edit, in Baseline."

**F6: Variation Management**—ABSScribe reduces task workload by enhancing variation management. Two major activities in the exploration of multiple variations were *variation storage*, or tracking *variation history*, as W7 referred to it, and *variation comparison*. W3, 6, 7, and 11 noted that ABSScribe's non-linear storage of variations necessitated less overhead to track *variation history* and do *variation comparisons*. W3 commented that by having the ABSScribe interface manage *variation storage* for them as opposed to manually placing text variations in the document, they were more able to focus on the writing task: "You're fully focused on the writing changes you're trying to make, as opposed to managing the state and managing your document and managing like, that kind of stuff. So that was the biggest difference for me. So I really liked that feature. And that made a huge difference in general to the task. But because

*I'm less focused on management of things, or management of my thoughts a little bit, it's a lot easier just using that, like the versioning system."*

While W3 touched on the ease of *variation storage*, W11 discussed how the Popup Toolbar enhanced the ease with which they did *variation comparison*: *"the feature made it easier to do the comparisons, because then you can click the version that you're - you're comparing with, and then hover and look at the text [for the other variations]. Whereas, with the Baseline, you have to both keep track of where the version you're comparing with is and also simultaneously figure out which version you're comparing to...so that is trickier than dealing with the new approach."*

A trade-off some users commented on was that it was easier to compare large segments of text in-sequence within the main document as is done in the Baseline. W3 and W7 both noted the benefit for comparing variations in-sequence within the document. W3 said *"Yeah, I mean, one nice thing about Baseline is that I do get to see all of the versions together [viewed at the same time in the main text editing space]. So they're all listed for me"* and W7 stated *"I think in that particular scenario[taking a small part of variation 1 and merging it with most of variation 2 in order to make my best variation], having a linear approach [like Baseline] may be more helpful for me to like, see all the different variants and then compare?"* However, they both followed their statements by noting the *Variation Sidebar* could serve a similar purpose so the trade-off was perceived as advantageous for ABScribe. W3 commented: *"[with Baseline] it's much harder to have a number of sentences, which you're generating different versions for because even once you start to hit two different sentences, [and] we're trying to generate different variations, the document becomes very cluttered, and becomes difficult to manage. And you forget what the context is for each of those different variations. So for context, and for clutter, I think A is vastly superior."* and W7 said *"your idea bucket [Variation Sidebar] does afford us that feature as well. Because I found myself doing that a lot. I was checking when, when we were first testing out the tool and giving me comps, I was checking to see if it was different enough. And we weren't I wasn't constantly like toggling in the document text, Right? So I was just like scanning the idea buckets [Variation Sidebar], I don't think that you're losing out on that feature, necessarily, just because I'm seeing that the linear approach is more helpful"*

**F7: Variation Management–ABScribe reduces task workload by showing variations in context of the surrounding text during manual editing and when editing with AI.** The need for considering context during the revision process came up during several interviews. The benefits of *variation management* on editing variations with context are exemplified by W3-8, and 10's comments. They note that thanks to the in-place *variation comparison* via the hover interaction with the Popup Toolbar, they were able to see what a variation looks like within a paragraph. To this effect, W10's states: *"I like the nonlinear version, because when you hover on the different buttons, you can directly see the impact of different variations within the paragraph or within the context. So in that way, you know, whether the text fit into the original document or not. Whereas in Baseline, if you put all the variations linearly in the document, at some point, you just start to lose a sense of what's the context of this of this sentence, what am I writing there?"* W7 echoes the value of viewing variations with surrounding context: *"When you're*

*writing a paragraph, you're not looking at a sentence in isolation. So if you're changing a particular sentence, you want to see how it looks in comparison to the rest of your text. And so to have the nonlinear version allows you to kind of do that more seamlessly than with a linear version, where you'd have to reorganize a lot more in order to have that effect."* Whereas, W10 mirrored this, and noted that manually organizing variations while simultaneously figuring out context was challenging: *"I need to manually think of a way to organize all the variations so that I understand what they mean. Or like, what, how they're connected to the original text. That cost a lot of time. And it's like, very high physical demand."*

**F8: Mixed–ABScribe's variation storage and in-place LLM revision reduces subjective task workload.** Interactions that bring the user outside of the primary text editing interface and break their flow of writing or revising were perceived to be effortful and time-consuming. W7 points out how *variation management*—particularly *variation storage* - with the Baseline interface necessitates context switching: *"So if I were to go into version history, and I wanted to go back to a very particular change in one particular paragraph, but I made that like 50 changes ago, I would either have to revert back to something where all of the document would have been unchanged, or I'd have to do like a very inconvenient and kind of cumbersome process of like copying that particular change from that particular version history into my current doc and then proceeding, which is, like tedious"*

W4, 5, 7, and 9 commented that they all had various ways of handling *variation management* in their naturalistic writing tasks such as producing articles, research papers, and books, which were similar to how they handled variations with the Baseline interface. These methods, such as copy-pasting different versions from separate documents into the primary text editing interface for comparison in-sequence, or doing *variation comparison* side by side in separate windows required them to leave the primary text editing interface to perform *variation comparison*. W5, 7, and 9 all found *variation comparison* less cumbersome with ABScribe, especially when performing edits on several smaller text segments, due to the lack of context switching. W9 explains this here: *"Whereas here [ABScribe], I think because it takes less effort, like okay, I can do one sentence. I also want to do another one. So I'll do that. I don't need to copy the whole paragraph in [the chat-interface of Baseline] and try to get an answer from that. I can just do it to those sentences."*

Use of AI within the Baseline was also indicated as a source of context switching. W10 notes how the chat-like interface in Baseline pulls them out of context: *"Also, the chatbot in Baseline, I'd say it's pretty much [the] same as ChatGPT. So if I want something, I need to scroll back to try to look for it. So that's pretty much similar to the current AI writing system."* ABScribe's AI modifiers and AI Drafter enable the use of AI without requiring leaving the main text editing interface.

**F9: AI Integration– ABScribe reduces task workload by making prompts more reusable than Baseline.** Almost all users (W2, 3, 6-10, 12) noted that prompt reuse was much easier in the ABScribe interface. W3 commented on how the Baseline interface imposed a *"memory load or cognitive load issue to remember what prompts you have."* W2 notes how the AI Modifiers of the ABScribe interface eliminate the need to rewrite prompts stating *"I don't need to rewrite prompts every time. It was really very quick and efficient."*

*The usability of this one in terms of buttons, the reusing the prompts is very good.” While W8 called ABScribe’s AI Modifiers “much more streamlined.” W3 sums up the interaction concisely in the following quote: “A is vastly superior for reusability, there is no question. B, you basically have to work from memory, which can also be fine. But with version A, you click it, you don’t have the same memory load or cognitive load issue to remember what prompts you have before, cannot be more better facilitated.”*

## 6 DISCUSSION

Writing plays a crucial role in many people’s lives, and the interfaces used for editing text can deeply influence the writing process. As LLMs increasingly blur the boundaries between human and AI writing capabilities [33, 100], the HCI community needs to explore interfaces that improve writers’ interactions with AI. In our work, we contributed an ensemble of five interface elements that significantly enhance user perceptions on AI-assisted revision (RQ1) by supporting rapid exploration and organization of multiple writing variations without overwhelming users (RQ2).

### 6.1 Design Implications

To interrogate the broader applicability of our results toward other contexts, and to characterize how ABScribe may relate to creativity support tools/authoring applications in general, we consider the following design implications:

**D1: Emphasize Parallel Storage of Text Variations To Nudge Users Toward Working with a Larger Number of Increasingly Granular Ideas** In line with HCI design principles that highlight the importance of investigating multiple alternatives [10, 89], we have developed ABScribe to facilitate the rapid exploration and organization of many text variations, which is a problem exacerbated by AI. Our findings suggest that parallel storage is a useful approach for inspiring writers to generate numerous ideas for granular sections of text, without becoming overly attached to any single variation. Although our implementation deals with text-segments, the challenge of organizing proliferating ideas through parallel storage extends beyond textual content, to other forms of creative content. As companies advance foundational models for manipulating various types of inputs — such as Google’s recently announced ‘natively multimodal’ Gemini model [73], described as being adept at ‘reasoning seamlessly across text, images, video, audio, and code’<sup>6</sup> — designers can explore ways to extend our design to creativity support tools for other modalities. For example, an application for editing AI-generated images could implement an AI Modifier which accepts textual instructions and generates corresponding image variations, which are then organized within Variation Fields, contributing to HCI research on text-to-image AI editors [6, 54, 55, 70, 93]. The Popup Toolbar could be adapted to non-linear layer management in image editors [1, 11], or applied in presentation software, allowing users to experiment with different headings and bullet points on a slide.

**D2: De-Emphasize Parallel Storage to Support Editing Forms that Are Better Suited to Linear Representations of Content.** Our findings also suggest that while parallel storage nudges users towards exploring multiple text variations, it remains important

to cater to linear editing preferences. Many participants preferred to view previous variations side-by-side or in-sequence when producing a new variation. For example, W6 noted their habit of linearly organizing text variations for side-by-side comparison, which seemed less intuitive in ABScribe. Although our design does not prevent the sequential pasting of text blocks, it subtly discourages linear editing. This insight underscores the potential benefits of a Variation Sidebar as a supplementary tool for linear viewing of variations, complementing the Popup Toolbar’s parallel functionality. A more complex Variation Sidebar that allows for reordering and editing inside the accordion could help with linear revisions. The Variation Fields could also be changed to allow for both parallel and linear editing, such as through incorporating an ‘explode/contract’ button in the Popup Toolbar that allow users to switch between a linear, exploded display of variations for easier comparison and a more compact, collapsed view within the Variation Field. Together, D1 and D2 contribute to existing discussions surrounding linear versus non-linear representations of versions history across various domains [11, 41, 71, 74, 88, 99].

**D3: Scaffold Prompts Within Discrete Interface Elements to Encourage Action-Oriented Prompt-Writing.** If the goal for designers is to guide users towards crafting more succinct and direct prompts, incorporating scaffolding around specific interface elements signifying actions can be effective. This approach is evident in the buttons created by AI Modifiers. In contrast, in free-form conversational user interfaces, we saw a tendency for users to adopt a more polite and verbose style. For example, participant W8 would say, “Can you please make this variation shorter?” accompanied by a pasted text snippet, rather than simply selecting a Variation Field and requesting, “Make it short.” Of course, there are contexts where encouraging a polite and detailed interaction with AI agent is beneficial, e.g., in training scenarios where a customer service representative practices dialogues with an AI. However, in situations focused on productivity tasks, such as composing an email, steering users towards a direct prompt-writing styles can help. Our findings illustrate how interface elements can noticeably influence prompt-writing style, akin to previous work on, such as Dang et al.’s investigation of diegetic and non-diegetic prompts [16]. It also adds to prior work using objects to represent AI prompts [47] and text segments [3, 36].

**D4: Reduce User Effort by Turning AI Prompts into Customizable Toolkit for Digital Tasks.** AI Modifiers in our interface highlights the potential of LLM prompts as tailored sets of reusable user actions, resembling a bespoke Swiss Army knife. For instance, users who frequently need to condense, translate, and proofread text can effortlessly create AI modifiers for these tasks, facilitating reuse and minimizing effort. Previous studies have shown that users may hesitate to write prompts when given multiple AI suggestions [16], underscoring the importance of reducing prompt-writing effort. Our design addresses aspects of this problem.

Unlike traditional menu buttons which have predictable outcomes, AI Modifiers can yield unexpected results due to the inherent unpredictability of AI models [40, 64, 95, 96]. Prior research on AI-assisted editing suggests that users may find imperfect AI outputs useful [97], given that they have the ability to edit. Thus, while using AI prompts as personalized toolkits is beneficial, designers will have to ensure that users can easily refine both the AI’s

<sup>6</sup><https://deepmind.google/technologies/gemini/>

imperfect output and the unpredictable instructions encapsulated by the AI modifier.

The current implementation of AI Modifiers as labeled buttons is only one of many possible approaches. Alternative user experiences could include AI-generated icons, options within a nested menu, or even simplified voice commands, for example, a user might initially issue a detailed voice command and then receive a suggestion for a condensed version for future use. Our design could be also be extended to give users more control over AI features, such as adjusting the 'temperature' to modulate predictability.

## 6.2 Limitations and Future Work

A noteworthy limitation of our work is that we constructed the baseline interface rather than opting to use an existing system in our evaluation. We chose to construct the baseline because early pilots of our evaluation study revealed that utilizing a real-world system as a baseline (in our case, ChatGPT and GoogleDocs) introduced several logistical challenges that made it difficult to control and run the study. For instance, to ensure ChatGPT used the same underlying LLM model, we had to provide participants with login details to a paid ChatGPT account, potentially exposing credit card details. When using Google Docs, participants would resort to using editing features outside the focus of our research, such as user comments. Our choice of constructing a baseline is in alignment with prior HCI research on AI writers [97]. However, this may have affected the ecological validity of our results. Future studies could compare ABScribe with real-world AI-assisted writing workflows.

There are also limitations to external validity common in lab-based evaluation studies. First, our evaluation study was restricted to English writers. Although the underlying LLM, GPT-4, supports multiple languages [69], our study focused on English writing tasks. Therefore, we would caution designers against directly applying our design without further evaluation. Second, our evaluation study was limited to a single 1.5-hour session with two guided writing tasks to ensure we could conduct our comparison in a controlled setting. Future work could investigate how writers select their own writing task and revise text over extended periods, possibly spanning several days.

A potential risk with any generative AI tool is that the AI may produce *hallucinated* or *confabulated* content [21, 45, 79, 85]. ABScribe's AI Drafter and AI Modifiers both present a risk of introducing hallucinations into the text, which becomes especially important to be cognizant of when writing in high-stakes domains or in educational settings. We carefully named our features to account for this. With the AI Drafter, the tool is specifically referred to as a *drafter*, not a *writing companion*, to nudge the user toward thorough revision of what it produces. The AI Modifier rephrases existing text written by the author, which provides the generative AI with a baseline grounded in the author's text, which is expected to be fact-checked.

Finally, it would be useful to implement logging features into ABScribe to concrete measures from the tool, such as: How many words did the subjects actually write; how many words were written by the LLM; what percentage of the resulting text was from the LLM vs. the human; how many times did the subjects interact with the various features in the ABScribe UI? We did not collect these metrics

as our focus was on addressing task workload and user perceptions. Future studies could investigate the relative contributions of human and AI in co-written documents.

## 7 CONCLUSION

In this work, we presented ABScribe, a human-AI co-writing interface built for swiftly exploring multiple writing variations using Large Language Models (LLMs). Our design only markedly decreases task workload ( $d = 1.20, p < 0.001$ ) but also bolsters user perceptions of the revision process ( $d = 2.41, p < 0.001$ ), in comparison to a baseline chat-based AI editing workflow. We evaluated our design with writers (N=12) to validate the efficacy of ABScribe and offer insight into how writers leverage LLMs to explore variations, revealing a preference for non-linear over linear revision strategies, especially when engaging with a multitude of variations at finer granularity levels. We also found that scaffolding LLM use with task-focused UI components, like buttons, encouraged writers to create more generalizable prompts and use more direct, imperative language in prompt design. Our work informs HCI research on the design of Human-AI writing interfaces for the rapid exploration of writing variations.

## ACKNOWLEDGMENTS

We are grateful to the writers who participated in this work. We would also like to thank Siqu (Jessica) Zhang and Divya Gupta for their assistance with the transcription of interviews, and Steven Tohme for his help in open-sourcing the codebase. This work was partially supported by grants from the Office of Naval Research (N00014-21-1-2576), the Natural Sciences and Engineering Research Council of Canada (RGPIN-2019-06968), and the National Science Foundation (2209819).

## REFERENCES

- [1] William A Barrett and Alan S Cheney. 2002. Object-based image editing. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery, New York, NY, USA, 777–784.
- [2] Nancy Baym, Limor Shifman, Christopher Persaud, and Kelly Wagman. 2019. INTELLIGENT FAILURES: CLIPPY MEMES AND THE LIMITS OF DIGITAL ASSISTANTS. *AoIR Selected Papers of Internet Research 2019* (Oct. 2019), 1–5. <https://doi.org/10.5210/spir.v2019i0.10923>
- [3] Michel Beaudouin-Lafon and Wendy E. Mackay. 2000. Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (Palermo, Italy) (AVI '00)*. Association for Computing Machinery, New York, NY, USA, 102–109. <https://doi.org/10.1145/345513.345267>
- [4] Oloff C. Biermann, Ning F. Ma, and Dongwook Yoon. 2022. From Tool to Companion: Storywriters Want AI Writers to Respect Their Personal Values and Writing Strategies. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference (Virtual Event, Australia) (DIS '22)*. Association for Computing Machinery, New York, NY, USA, 1209–1227. <https://doi.org/10.1145/3532106.3533506>
- [5] Sara Bly. 2007. Fundamentals in HCI: Learning the Value of Consistency and User Models. In *HCI Remixed: Reflections on Works That Have Influenced the HCI Community*. The MIT Press, Cambridge, MA, USA. <https://doi.org/10.7551/mitpress/7455.003.0012> arXiv:[https://direct.mit.edu/book/chapter-pdf/2290056/9780262256070\\_cah.pdf](https://direct.mit.edu/book/chapter-pdf/2290056/9780262256070_cah.pdf)
- [6] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. 2023. Promptify: Text-to-Image Generation through Interactive Prompt Exploration with Large Language Models. arXiv:2304.09337 [cs.HC]
- [7] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.
- [8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prfulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

- [9] Daniel Buschek, Martin Zürn, and Malin Eiband. 2021. The impact of multiple parallel phrase suggestions on email input and composition behaviour of native and non-native english writers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13.
- [10] Bill Buxton. 2010. *Sketching user experiences: getting the design right and the right design*. Morgan kaufmann, Boston.
- [11] Hsiang-Ting Chen, Li-Yi Wei, and Chun-Fa Chang. 2011. Nonlinear revision control for images. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–10.
- [12] N Ann Chenoweth. 1987. The need to teach rewriting. *ELT journal* 41, 1 (1987), 25–29.
- [13] Yin Ling Cheung. 2016. *Teaching Writing*. Springer International Publishing, Cham, 179–194. [https://doi.org/10.1007/978-3-319-38834-2\\_13](https://doi.org/10.1007/978-3-319-38834-2_13)
- [14] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–19.
- [15] Robert Dale. 2021. GPT-3: What's it good for? *Natural Language Engineering* 27, 1 (2021), 113–118.
- [16] Hai Dang, Sven Goller, Florian Lehmann, and Daniel Buschek. 2023. Choice over control: How users write with large language models using diegetic and non-diegetic prompting. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–17.
- [17] Joost FC de Winter and Dimitra Dodou. 2010. Five-point likert items: t test versus Mann-Whitney-Wilcoxon (Addendum added October 2012). *Practical Assessment, Research, and Evaluation* 15, 1 (2010), 11.
- [18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 <http://arxiv.org/abs/1810.04805>
- [19] Steven P Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L Schwartz, and Scott R Klemmer. 2010. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 4 (2010), 1–24.
- [20] Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022. Understanding Iterative Revision from Human-Written Text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 3573–3590. <https://doi.org/10.18653/v1/2022.acl-long.250>
- [21] Robin Emsley. 2023. ChatGPT: these are not hallucinations—they're fabrications and falsifications. *Schizophrenia* 9, 1 (2023), 52.
- [22] Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College composition and communication* 32, 4 (1981), 400–414.
- [23] Jill Fitzgerald. 1987. Research on revision in writing. *Review of educational research* 57, 4 (1987), 481–506.
- [24] Linda Flower and John R Hayes. 1981. A cognitive process theory of writing. *College composition and communication* 32, 4 (1981), 365–387.
- [25] Linda Flower and John R. Hayes. 1981. A Cognitive Process Theory of Writing. *College Composition and Communication* 32, 4 (1981), 365–387. <http://www.jstor.org/stable/356600>
- [26] Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for Science Writing using Language Models. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference (Australia) (DIS '22)*. Association for Computing Machinery, New York, NY, USA, 1002–1019. <https://doi.org/10.1145/3532106.3533533>
- [27] Katy Ilonka Gero, Tao Long, and Lydia B Chilton. 2023. Social dynamics of AI support in creative writing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–15.
- [28] Louie Giray. 2023. Prompt Engineering with ChatGPT: A Guide for Academic Writers. *Annals of Biomedical Engineering* 51 (06 2023), 3. <https://doi.org/10.1007/s10439-023-03272-4>
- [29] Natalie Goldberg. 2016. *Writing down the bones: Freeing the writer within*. Shambhala Publications, 2129 13th St, Boulder, Colorado.
- [30] Gabriela Goldschmidt. 2011. Avoiding design fixation: transformation and abstraction in mapping from source to target. *The Journal of creative behavior* 45, 2 (2011), 92–100.
- [31] Steven M Goodman, Erin Buehler, Patrick Clary, Andy Coenen, Aaron Donsbach, Tiffanie N Horne, Michal Lahav, Robert MacDonald, Rain Breaw Michaels, Ajit Narayanan, et al. 2022. Lampost: Design and evaluation of an ai-assisted email writing prototype for adults with dyslexia. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*. Association for Computing Machinery, New York, NY, USA, 1–18.
- [32] Google. 2023. PaLM 2 Technical Report. <https://ai.google/static/documents/palm2techreport.pdf>.
- [33] Katja Grace, John Salvatiere, Allan Dafoe, Baobao Zhang, and Owain Evans. 2018. When will AI exceed human performance? Evidence from AI experts. *Journal of Artificial Intelligence Research* 62 (2018), 729–754.
- [34] Saul Greenberg and Bill Buxton. 2008. Usability Evaluation Considered Harmful (Some of the Time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) (CHI '08). Association for Computing Machinery, New York, NY, USA, 111–120. <https://doi.org/10.1145/1357054.1357074>
- [35] Han L Han. 2020. Designing Representations for Digital Documents. In *Adjunct Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 174–178.
- [36] Han L Han, Miguel A Renom, Wendy E Mackay, and Michel Beaudouin-Lafon. 2020. Textlets: Supporting constraints and consistency in text documents. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13.
- [37] Han L Han, Miguel A. Renom, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2020. Textlets: Supporting Constraints and Consistency in Text Documents. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376804>
- [38] Joseph Harris. 2017. *Rewriting: How to do things with texts*. University Press of Colorado, Boulder, CO, USA.
- [39] Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (2006), 904–908. <https://doi.org/10.1177/154193120605000909> arXiv:<https://doi.org/10.1177/154193120605000909>
- [40] Iikka Hauhio, Anna Kantosalo, Simo Linkola, and Hannu Toivonen. 2023. The Spectrum of Unpredictability and its Relation to Creative Autonomy. In *Proceedings of the 14th International Conference on Computational Creativity*. Association for Computational Creativity (ACC), Spain, 148–152. <https://computationalcreativity.net/iccc23/> International Conference on Computational Creativity, ICC ; Conference date: 19-06-2023 Through 23-06-2023.
- [41] Andrew Head, Fred Hohman, Titus Barik, Steven M. Drucker, and Robert DeLine. 2019. Managing Messes in Computational Notebooks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300500>
- [42] Ernest Hemingway. 2014. *Moveable feast: the restored edition*. Simon and Schuster, New York, NY, USA.
- [43] Roland K Huff. 1983. Teaching revision: A model of the drafting process. *College English* 45, 8 (1983), 800–816.
- [44] David G Jansson and Steven M Smith. 1991. Design fixation. *Design studies* 12, 1 (1991), 3–11.
- [45] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.
- [46] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. 2022. PromptMaker: Prompt-based Prototyping with Large Language Models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI EA '22). Association for Computing Machinery, New York, NY, USA, Article 35, 8 pages. <https://doi.org/10.1145/3491101.3503564>
- [47] Tae Soo Kim, Yoonjoo Lee, Minsuk Chang, and Juho Kim. 2023. Cells, Generators, and Lenses: Design Framework for Object-Oriented Interaction with Large Language Models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 4, 18 pages. <https://doi.org/10.1145/3586183.3606833>
- [48] Yewon Kim, Mina Lee, Donghwi Kim, and Sung-Ju Lee. 2023. Towards Explainable AI Writing Assistants for Non-native English Speakers.
- [49] Barbara Kroll and Joy Reid. 1994. Guidelines for designing writing prompts: Clarifications, caveats, and cautions. *Journal of Second Language Writing* 3, 3 (1994), 231–255.
- [50] Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. Association for Computing Machinery, New York, NY, USA, 1–19.
- [51] Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, et al. 2022. Evaluating human-language model interaction.
- [52] Carrie S Leverenz. 2014. Design thinking and the wicked problem of teaching writing. *Computers and Composition* 33 (2014), 1–12.
- [53] Clayton Lewis and Cathleen Wharton. 1997. Chapter 30 - Cognitive Walkthroughs. In *Handbook of Human-Computer Interaction (Second Edition)* (second edition ed.), Marting G. Helander, Thomas K. Landauer, and Prasad V. Prabh (Eds.). North-Holland, Amsterdam, 717–732. <https://doi.org/10.1016/B978-0-44481862-1.50096-0>
- [54] Vivian Liu. 2023. Beyond Text-to-Image: Multimodal Prompts to Explore Generative AI. In *Extended Abstracts of the 2023 CHI Conference on Human*



- Factors in Computing Systems* (Hamburg, Germany) (CHI EA '23). Association for Computing Machinery, New York, NY, USA, Article 482, 6 pages. <https://doi.org/10.1145/3544549.3577043>
- [55] Vivian Liu and Lydia B Chilton. 2022. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 483, 23 pages. <https://doi.org/10.1145/3491102.3501825>
- [56] Tao Long, Dorothy Zhang, Grace Li, Batool Taraiif, Samia Menon, Kymnedy Simone Smith, Sitong Wang, Katy Ilonka Gero, and Lydia B Chilton. 2023. Tweeporial Hooks: Generative AI Tools to Motivate Science on Social Media.
- [57] Charles A MacArthur. 2018. Evaluation and revision. *Best practices in writing instruction* 287 (2018), 23 pages.
- [58] Stephen MacNeil, Andrew Tran, Juho Leinonen, Paul Denny, Joanne Kim, Arto Hellas, Seth Bernstein, and Sami Sarsa. 2023. Automatically Generating CS Learning Materials with Large Language Models. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2* (Toronto ON, Canada) (SIGCSE 2023). Association for Computing Machinery, New York, NY, USA, 1176. <https://doi.org/10.1145/3545947.3569630>
- [59] Thomas Mahatody, Mouldi Sagar, and Christophe Kolski. 2010. State of the art on the cognitive walkthrough method, its variants and evolutions. *Intl. Journal of Human-Computer Interaction* 26, 8 (2010), 741–785.
- [60] George A. McCulley. 1985. Writing Quality, Coherence, and Cohesion. *Research in the Teaching of English* 19, 3 (1985), 269–282. <https://www.jstor.org/stable/40171050> Publisher: National Council of Teachers of English.
- [61] Deborah McCutchen, Paul Teske, and Catherine Bankston. 2009. Writing and cognition: Implications of the cognitive architecture for learning to write and writing to learn. In *Handbook of research on writing*. Routledge, Milton Park, Abingdon, Oxfordshire, United Kingdom, 554–578.
- [62] John McPhee. 2017. *Draft No. 4: On the writing process*. Farrar, Straus and Giroux, 18 West 18th Street New York, NY US.
- [63] Selina Meyer, David Elsweller, Bernd Ludwig, Marcos Fernandez-Pichel, and David E Losada. 2022. Do we still need human assessors? prompt-based gpt-3 user simulation in conversational ai. In *Proceedings of the 4th Conference on Conversational User Interfaces*. Association for Computing Machinery, New York, NY, USA, 1–6.
- [64] Holland Michel. 2020. A.: The black box, unlocked: predictability and understandability in military AI. Ginebra, United Nations Institute for Disarmament Research.
- [65] Microsoft. 2013. Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web. <https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/>.
- [66] Bonan Min, Hayley Ross, Elinor Sulem, Amir Pouran Ben Veyshe, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *Comput. Surveys* 56, 2 (2023), 1–40.
- [67] Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. 2023. Co-Writing Screenplays and Theatre Scripts with Language Models: Evaluation by Industry Professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 355, 34 pages. <https://doi.org/10.1145/3544548.3581225>
- [68] OpenAI. 2022. Introducing ChatGPT. <https://openai.com/blog/chatgpt>.
- [69] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]
- [70] Jonas Oppenlaender. 2022. The Creativity of Text-to-Image Generation. In *Proceedings of the 25th International Academic Mindtrek Conference* (Tampere, Finland) (*Academic Mindtrek '22*). Association for Computing Machinery, New York, NY, USA, 192–202. <https://doi.org/10.1145/3569219.3569352>
- [71] Bryan O'Sullivan. 2009. Making sense of revision-control systems. *Commun. ACM* 52, 9 (2009), 56–62.
- [72] Sundar Pichai. 2023. An important next step on our AI journey. <https://blog.google/technology/ai/bard-google-ai-search-updates/>.
- [73] Sundar Pichai and Demis Hassabis. 2023. Introducing Gemini: Our largest and most capable AI model. <https://blog.google/technology/ai/google-gemini-ai/#sundar-note>
- [74] John Plaice and William W Wadge. 1993. A new approach to version control. *IEEE transactions on Software Engineering* 19, 3 (1993), 268–276.
- [75] James P. Purdy. 2014. What Can Design Thinking Offer Writing Studies? *College Composition and Communication* 65, 4 (2014), 612–641. <http://www.jstor.org/stable/43490875>
- [76] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- [77] Sarah Ransdell, C. Michael Levy, and Ronald T. Kellogg. 2002. The structure of writing processes as revealed by secondary task demands. *L1-Educational Studies in Language and Literature* 2, 2 (May 2002), 141–163. <https://doi.org/10.1023/A:1020851300668>
- [78] Horst Rittel. 1967. Wicked problems. *Management Science*,(December 1967) 4, 14 (1967), B141–B142.
- [79] Michele Salvagno, Fabio Silvio Taccone, and Alberto Giovanni Gerli. 2023. Artificial intelligence hallucinations. *Critical Care* 27, 1 (2023), 1–2.
- [80] Reinhard Sefelin, Manfred Tscheligi, and Verena Giller. 2003. Paper prototyping - what is it good for? a comparison of paper- and computer-based low-fidelity prototyping. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) (CHI EA '03). Association for Computing Machinery, New York, NY, USA, 778–779. <https://doi.org/10.1145/765891.765986>
- [81] Anthony Seow. 2002. The writing process and process writing. *Methodology in language teaching: An anthology of current practice* 315 (2002), 320.
- [82] Hanieh Shakeri, Carman Neustaedter, and Steve DiPaola. 2021. Saga: Collaborative storytelling with gpt-3. In *Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing*. Association for Computing Machinery, New York, NY, USA, 163–166.
- [83] Mike Sharples. 2002. *How we write: Writing as creative design*. Routledge, Milton Park, Abingdon, Oxfordshire, United Kingdom.
- [84] Nikhil Singh, Guillermo Bernal, Daria Savchenko, and Elena L Glassman. 2023. Where to hide a stolen elephant: Leaps in creative writing with multimodal machine intelligence. *ACM Transactions on Computer-Human Interaction* 30, 5 (2023), 1–57.
- [85] Andrew L Smith, Felix Greaves, and Trishan Panch. 2023. Hallucination or Confabulation? Neuroanatomy as metaphor in Large Language Models. *PLOS Digital Health* 2, 11 (2023), e0000388.
- [86] Carolyn Snyder. 2003. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [87] Nancy Sommers. 1980. Revision strategies of student writers and experienced adult writers. *College composition and communication* 31, 4 (1980), 378–388.
- [88] Walter F. Tichy. 1982. Design, implementation, and evaluation of a Revision Control System. In *Proceedings of the 6th International Conference on Software Engineering* (Tokyo, Japan) (ICSE '82). IEEE Computer Society Press, Washington, DC, USA, 58–67.
- [89] Maryam Tohidi, William Buxton, Ronald Baecker, and Abigail Sellen. 2006. Getting the Right Design and the Design Right. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) (CHI '06). Association for Computing Machinery, New York, NY, USA, 1243–1252. <https://doi.org/10.1145/1124772.1124960>
- [90] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shrubti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models.
- [91] 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*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., Long Beach, CA. [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fb0d53c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fb0d53c1c4a845aa-Paper.pdf)
- [92] Miriam Walker, Leila Takayama, and James A. Landay. 2002. High-Fidelity or Low-Fidelity, Paper or Computer? Choosing Attributes when Testing Web Prototypes. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 46, 5 (2002), 661–665. <https://doi.org/10.1177/154193120204600513> arXiv:https://doi.org/10.1177/154193120204600513
- [93] Yunlong Wang, Shuyuan Shen, and Brian Y Lim. 2023. RePrompt: Automatic Prompt Editing to Refine AI-Generative Art Towards Precise Expressions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (CHI '23). ACM, Singapore, 1–29. <https://doi.org/10.1145/3544548.3581402>
- [94] Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. 2022. Promptchainer: Chaining large language model prompts through visual programming. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. Association for Computing Machinery, New York, NY, USA, 1–10.
- [95] Roman V Yampolskiy. 2019. Unpredictability of AI.
- [96] Roman V Yampolskiy. 2020. Unpredictability of AI: On the impossibility of accurately predicting all actions of a smarter agent. *Journal of Artificial Intelligence and Consciousness* 7, 01 (2020), 109–118.
- [97] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story Writing With Large Language Models. In *27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) (IUI '22). Association for Computing Machinery, New York, NY, USA, 841–852. <https://doi.org/10.1145/3490099.3511105>
- [98] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 437, 21 pages. <https://doi.org/10.1145/3544548.3581388>
- [99] Lei Zhang, Ashutosh Agrawal, Steve Oney, and Anhong Guo. 2023. VRGit: A Version Control System for Collaborative Content Creation in Virtual Reality. In

*Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 36, 14 pages. <https://doi.org/10.1145/3544548.3581136>

- [100] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers.
- [101] William Knowlton Zinsser. 2001. *On writing well: The classic guide to writing nonfiction*. Quill/A Harper Collins Books, 195 Broadway New York, NY, USA.

## A BASELINE INTERFACE

The Baseline interface featured a conversational AI assistant similar to ChatGPT and the ability to insert AI generated text directly into the document. Refer to Figure 9 for a screenshot.

## B SCENARIO DESCRIPTIONS AND PROMPTS

The two task scenarios were described to the participants as follows:

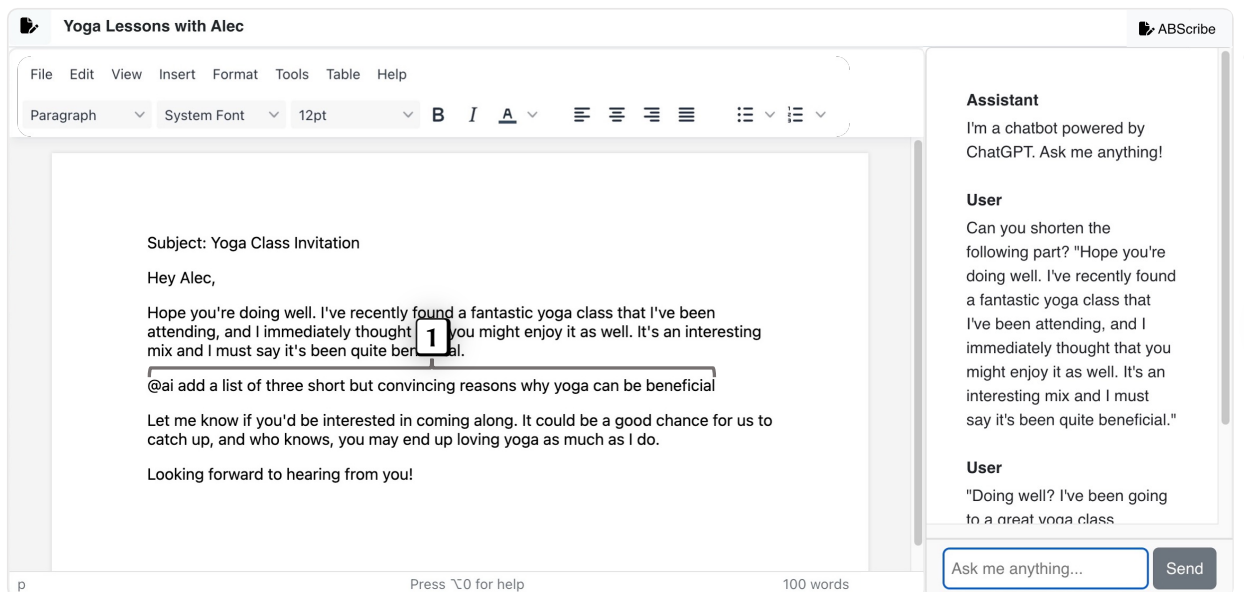
- **LinkedIn Post:** Imagine you're crafting a LinkedIn post to secure a copywriting job. Copywriters produce captivating, clear-cut text tailored for various advertising mediums like websites and print ads. You want to convince your network to point to relevant opportunities and form new connections.
- **Email to a Professor:** Imagine you're writing an email to introduce yourself to Professor Bardley, with whom you've never communicated before. Aiming to leave a positive first impression, you're exploring multiple ways to best introduce yourself.

For each scenario, participants generated an initial draft of roughly the same length using the following prompts:

- **LinkedIn Prompt:** Help me write a LinkedIn post to find a job as a copywriter. I have some experience writing posts for a university club to ensure members stay engaged. I also

took a course on copywriting last fall and want to highlight that. I am excited about writing and want to convince my connections to direct me to roles that might be a good fit or introduce me to people. Keep it within three paragraphs.

- **Email Prompt:** Compose an email to Professor Bardley. I've never had the opportunity to meet them, but I'm eager to make a favorable first impression. I'm enrolled in their Computational Social Science course for the upcoming fall and aspire to join their lab as a research assistant next summer. I want to convey my familiarity with their significant work on detecting misinformation on social media and developing tools to counteract it. Keep it within three paragraphs.



**Figure 9: The Baseline Interface: (1) We retained the ability to insert text directly into the document using AI Drafter as some modern AI editors have similar capabilities. (2) The chat-based interface on the left used the same underlying model (GPT-4) as ABScribe. All tangential differences such as font size and rich-text editing capabilities were consistent with ABScribe.**