ABScribe: Rapid Exploration of Multiple Writing Variations in Human-AI Co-Writing Tasks using Large Language Models

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

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

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

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

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

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

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

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

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

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

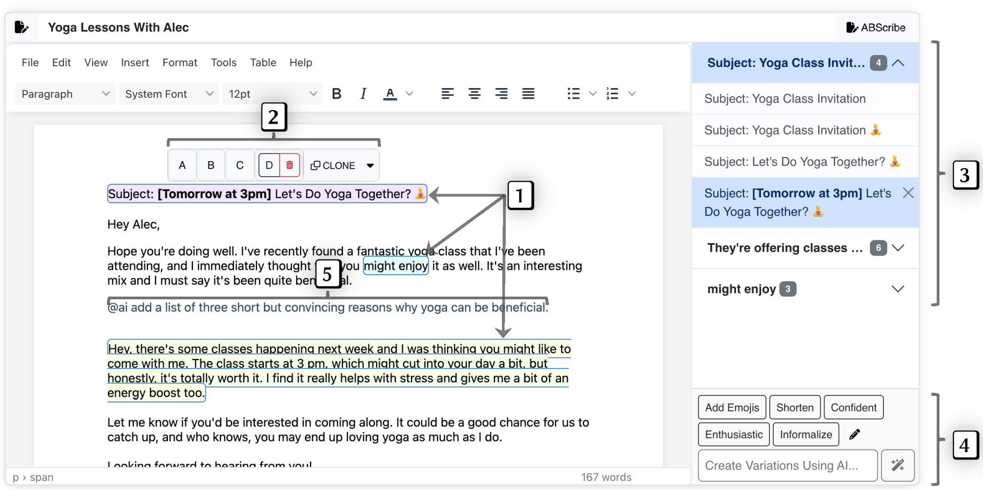

Figure 1: The ABScribe Interface: (1) Variation Components: Multiple variations are stored within text-segments that do not break the flow of the draft. (2) Hover Buttons: Users can swiftly compare multiple variations by hovering over buttons placed above the selected Variation Component, or clone and edit them in-place. (3) Variation Accordion: Users can view multiple variations and navigate through them using an organized accordion structure. (4) AI Buttons: Users can quickly create variations using AI by typing instructions auto-converted into reusable buttons that can be applied to other Variation Components. (5) AI Insert: Users can insert text from GPT-4 directly into the document by typing '@ai <prompt>' and pressing enter.

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 versions 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 of writing variations in human-AI co-writing tasks.

With ABScribe, users can swiftly produce multiple variations using LLM prompts, which are auto-converted into reusable buttons. Variations are stored adjacently within text segments for rapid in-place comparisons using mouse-over interactions on a context 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

1 INTRODUCTION

"The only kind of writing is rewriting" – Ernest Hemingway, *A Moveable Feast* [33]

Revision is an essential part of the writing process [23, 33, 48, 65]. Professional writers often write and rewrite text hundreds of times [12, 59] and recommend rewriting as a core strategy for writing well [12, 65]. Effective revision goes beyond minor editorial changes, and may help writers rework ideas, and powerfully affect their knowledge [19, 59] as they explore alternative variations to find a line of argument [19]. The revision process is *iterative* [17, 19]: happens in repeated cycles, throughout the writing process, *granular* [48, 56]: happens at the word, sentence, or paragraph-level, and *non-linear*[13, 59]: requires constant reconsideration of potential variations of existing text elements 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 [60] and avoid fixation [11, 26, 35, 60]. 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 [2], GPT-4 [50], PaLM 2 [3], and LLaMA 2 [61], can enable writers to generate multiple variations of text via prompting [10, 14], potentially reducing the workload of generating text variations. However, easier generation can exacerbate challenges surrounding the systematic storage, 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 or store them in their document editor 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 [8, 22, 64] and exploring the generative capabilities of LLMs [40, 45, 62, 63]. For example, Zamfirescu-Pereira et al. investigated ways to support non-AI experts with crafting effective prompts, and Yuan et al. [63] explored how users might use LLMs for creative writing. They 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 [63]. 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[1]–a novel writing interface that supports the rapid exploration of multiple writing variations in LLM-based human-AI co-writing tasks. We draw inspiration from Kim et al. design framework which shows the potential for object-oriented interactions with LLMs to encourage iteration and experimentation during writing [37], and propose a suite of five interface elements that support writers in swiftly exploring multiple writing variations by interacting with an ensemble of five interface elements: (i) **Variation Components** that store multiple human and AI-generated variations within flexible text segments in a non-linear manner, without overwriting text; (ii) **Hover Buttons** that reveal corresponding versions inside a Variation Component when users hover their mouse over them, allowing for rapid comparisons without breaking text flow; (iii) the **Variation Accordion** that organizes all variations in a navigable format; (iv) **AI Buttons** that automatically encapsulates LLM instructions into reusable buttons that can be applied across different Variation Components; and (v) **AI Insert** that allows writers to insert LLM-generated text directly into the document (see Figure 1).

To validate our design, we conducted a controlled evaluation study and interviews with 12 writers comparing ABScribe with a widely-used baseline worklow consisting of an AI integrated rich text editor based on GPT-4, with a chat-based 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 of multiple text variations in human-AI co-writing tasks.
(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 review literature in HCI and traditional design practices on weighing multiple alternatives, and discuss the relevance of this

[1]We name our system *ABScribe* to reference how we label multiple variations using the alphabet. Note that we support variations beyond just A and B. The Hover buttons can include multiple variations: A, B, C, D, E, etc.

design method when revising writing, guided by they theory on revision process in writing. We delve into difficulties that arise when trying to support multiple variation exploration using existing editing interfaces and contrast between *chat-based* and *in-place* interfaces to situate our design within a broader class of Human-AI writing interfaces.

2.1 Exploring Multiple Variations

HCI and traditional design practice encourages the parallel exploration of multiple variations to help avoid fixation on a singular idea [24, 35], to reduce the chances of eliminating rough but innovative ideas due to premature evaluation [11, 26], and to make us less prone to inflated subjective appraisals by giving us an opportunity to critically assessing ideas in relation to each other [11, 60]. 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 don't 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 the writing process, we see that revision goes beyond surface level edits [19], encompassing deeper writing subprocesses such as revising and evaluating ideas [20] and meaning discovery [59]. Experienced writers treat revision as a recursive, non-linear process [56, 59], and engage with the text in repeated cycles, with multiple objectives including finding the form or shape of an argument [59], experimenting with vocabulary and style [34], and going back and forth between multiple composing activities as writers revise text [18].

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 [55] design thinking problem [42, 52]. There has also been some valuable work in HCI to support novel editing practices, such as supporting constraints and consistency in maintaining domain-specific terms across complex documents [28] using persistent, reified [6] text selections, and present the idea and implementation of a *varientlet*, that allows 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.

In this paper, we contribute to this line of work, and present a suite of interface elements that work together in supporting writers with the rapid exploration 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

As we work toward leveraging advanced LLMs such as ChatGPT [2], GPT-4 [50], PaLM 2 [3], and LLaMA 2 [61], which can enable writers to generate multiple variations of text based on different parts of their writing using prompts, HCI researchers looking to design writing interfaces are faced with several challenges. These include dealing with the non-deterministic nature of these models [36, 37], systematically exploring their capabilities [40, 63], and making prompt-writing easier for AI-novices [57, 64].

There has also been work on managing the output from generative AI to support the exploration of variations in different contexts such as exploring images [8, 39], and multi-modal interactions beyond text prompts to explore generative AI [43]. However, our understanding of how to best organize the prolific output in AI-augmented writing workflows is still limited.

We draw inspiration from Kim et al's work on the use of object-oriented interactions and reification [29] to encourage writers' experimentation of LLM output, as well as prior work on revision control in writing [27, 28] and other domains [32], and offer the design and implementation of a novel interface that tackles the problem of how to effectively *organize* multiple variations from an LLM in a way that minimizes task workload while supporting parallel exploration.

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

To help ground our interface design, we distinguish between two types of Human-AI Co-Writing interfaces into two types: conversational interfaces such as ChatGPT and Bard, and In-Place interfaces that directly inserts or modifies text in the document.

Chat-Based Interfaces: Currently the dominant mode, chat-based interfaces, like ChatGPT [2], Bing Chat [1], and Bard [51], have gained immense popularity. These conversational interfaces are highly intuitive, and mimic human-to-human chat interactions, but lack scaffolding for crafting prompts, which can be difficult for novice AI users [38, 64]. Another significant limitation is the linear chat-log structure. In contrast to the non-linear nature of how revision happens in writing [19, 59] 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) [7] approach where AI-generated text modifies the human text and vice versa. This offers increased flexibility over the edited content compared to chat and form-based interfaces by allowing users to edit individual sections of the text.

Recent research prototypes for LLM writing tools such as Wordcraft [63] and CoAuthor [40] allow for in-place editing. Wordcraft [63] melds an in-text interface with diffrent options for users to continue a narrative based on prior text and replacing text selections with AI-modified content. CoAuthor explores GPT-3's capabilities by capturing deep interactions between writers and GPT-3 via a similar in-place editing interface where users receive multiple edit suggestions from the model. Commercial tools like Grammarly[2]

[2]grammarly.com

and Wordtune[3] also allow users to enhance their writing by revising text using AI-driven suggestions. However, once generated by the AI and modified by the user, previous text can become obscured or lost as older text is superseded by new edits. Even if text is auto-saved, it is often preserved as linear version histories, as is the case in Google Docs (an online word processor), or undo/redo histories, making it difficult to work with multiple text variations concurrently.

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 carefully construct a baseline interface that represents current workflows, providing fresh empirical insights based on our interviews with writers. These insights help us understand user perceptions of the revision process and explore how differences between in-place editing and chat-based AI writing companions impact their workflow.

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 [19, 21, 30, 44, 58, 59]; HCI design philosophies that emphasize the consideration of multiple ideas before evaluation [16, 26, 60]; principles on reification and reuse for designing visual interfaces [6, 29]; and the latest research on utilizing LLMs in writing interfaces to foster experimentation and creativity [37, 40, 63]. Based on this, we formulated an initial set of design requirements, which are summarized below.

Requirement 1: Minimizing Task Workload while Exploring Multiple Variations of Text 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 disciplines encourage the iteration and evaluation of multiple solutions in parallel to ultimately "get the right design" [60]. We hypothesized that parallel exploration of text segments could aid writers during the revision process. We also considered that during the writing process writers frequently struggle with cognitive overload [47] and that even small demands on working memory can lead to decreased fluency [53]. 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 become quickly cluttered when trying to explore 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 manage variations. Our goal was to support writers in seamlessly integrating LLM-generated variations without cluttering the document or erasing existing content to retain the ability to simultaneously consider multiple variations. By presenting users with a range of variations, we give them the opportunity to select their favorites while simultaneously discarding less-favored alternatives [26].

Requirement 3: Support context-sensitive variation comparison and revision In a linear document editing interface, we found it difficult to maintain a sense of the surrounding text to situate new variations within existing context. This was particularly poignant when creating and comparing variations for smaller and embedded text segments - eg. 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 to surrounding text [46] This becomes increasingly challenging as the document holds more and more variations of text segments. Our objective was to design an interface that allows writers to systematically evaluate these variations within context, eliminate less-favored options, and generate new iterations based on existing ones.

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 [19, 59], we aimed to align our LLM integrations with this fluid, iterative nature of revision. Our goal is 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 draw inspiration from design principles for visual interfaces, focusing on reuse, polymorphism, and reification [6], and regard LLM prompts as reusable, polymorphic commands that can be applied to targeted text segments of varying lengths, transforming them into first-class objects. 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 [37]. We adopt aspects of this approach in our design, such as reifying [6] LLM prompts into reusable AI buttons, and turning text segments into interactive Variation Component objects or *cells* [37].

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 [60] the lead author iterated through multiple versions of each interface element, and tested them with a total of five pilot users during in-depth brainstorming and design-review sessions over six months.

Variation Components: Users can select any part of the text and create interactive Variation Components that can hold multiple writing variations (Figure 2, Part 1). Newer variations can be

[3]wordtune.com

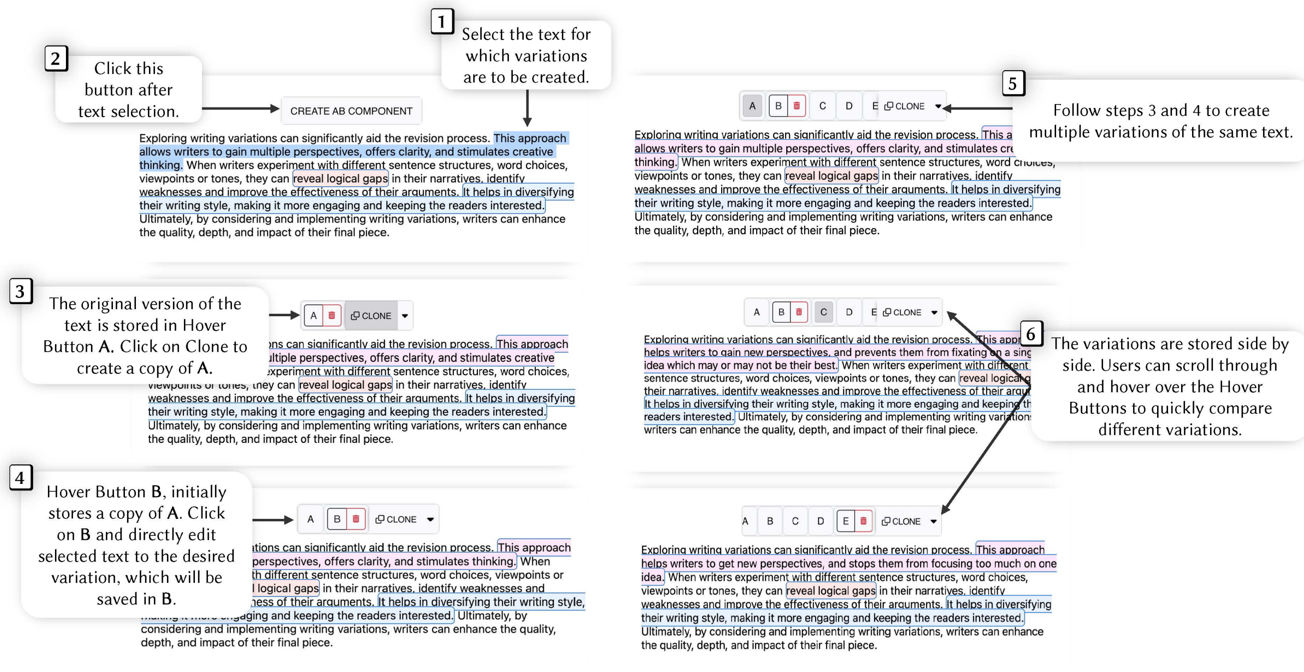

Figure 2: Hover Buttons & Variation Components: ABScribe supports the ability to store multiple writing variations in a variation component. These variations can be easily compared and swapped as shown above.

added to an AB Component without overwriting existing variations (Figure 2 Parts 2 and 3), and without breaking the flow of the passage.

Hover Buttons: Variations are represented using AB Hover Buttons that are dynamically placed above the active AB Component. Users can hover over each button to reveal the corresponding variation in the context of the surrounding passage (Figure 2, Parts 4 and 5). Moving the cursor away from the AB Hover Button reverts the AB Component back to the selected variation, allowing users to quickly compare between the selected and the hovered variation. Users can click on the Hover Button to select the variation and edit them in place, or discard it by clicking on the trash icon (Figure 2, Part 6).

Variation Accordion: To help writers view multiple variations together, and navigate through them more easily, we pair the Hover Buttons with an accordion structure where each Variation Component has its own header, and all corresponding variations are stored underneath. Clicking on the variations in the accordion dynamically re-positions the Hover Buttons above the corresponding Variation Component, and vice versa, allowing users to manage multiple variations in a visually structured manner (Figure 3)

The Variation Components, Hover Buttons, and the Variation Accordion work together to address R1, R2 and R3. To tackle R4, we developed two interface elements.

AI Buttons: Users can generate new variations by selecting an AB Component and typing instructions to the AI (Figure 3, Part 1). Instructions are automatically converted into labeled buttons (Figure 3, Part 2). The labels are generated using the LLM. As users experiment with newer variations using the AI, they create a set of custom AI Variation Buttons that they can reuse to apply to different parts of the passage, making these buttons reusable (Figure 3, Part 4). The prompts and labels for the buttons can be edited and improved over time. (Figure 4, Part 5). This allows writers to not only create a set of variations for a particular AB Component, but also design a set of 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.

AI Insert: Users can insert text from the AI model anywhere within the passage by writing instructions to the LLM in the following format: @ai <prompt>. 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 doesn't match what the user is looking for (Figure 5).

4 EVALUATING ABSCRIBE

To validate our design, we conducted an within-subjects evaluation study where we compared ABScribe to a carefully constructed baseline interface. The Baseline interface featured rich-text editing capabilities commonly found in word processing software such as Google Docs and Microsoft Word, as well as a conversational AI assistant similar to ChatGPT, and the ability to incorporate AI generated text into the document without the need to copy and paste to represent the tighter AI integration available in modern

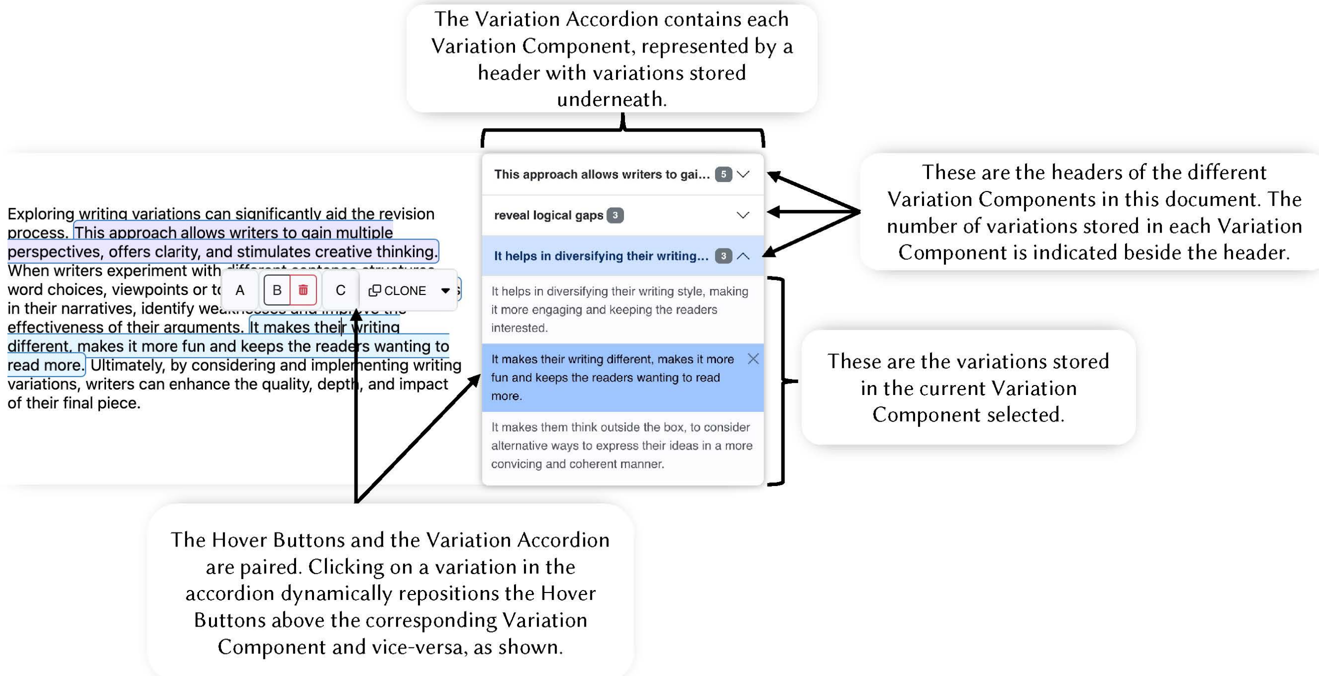

Figure 3: Variation Accordion: The Variation Accordion is an alternative method to viewing existing variations and is especially useful in viewing multiple variations side by side

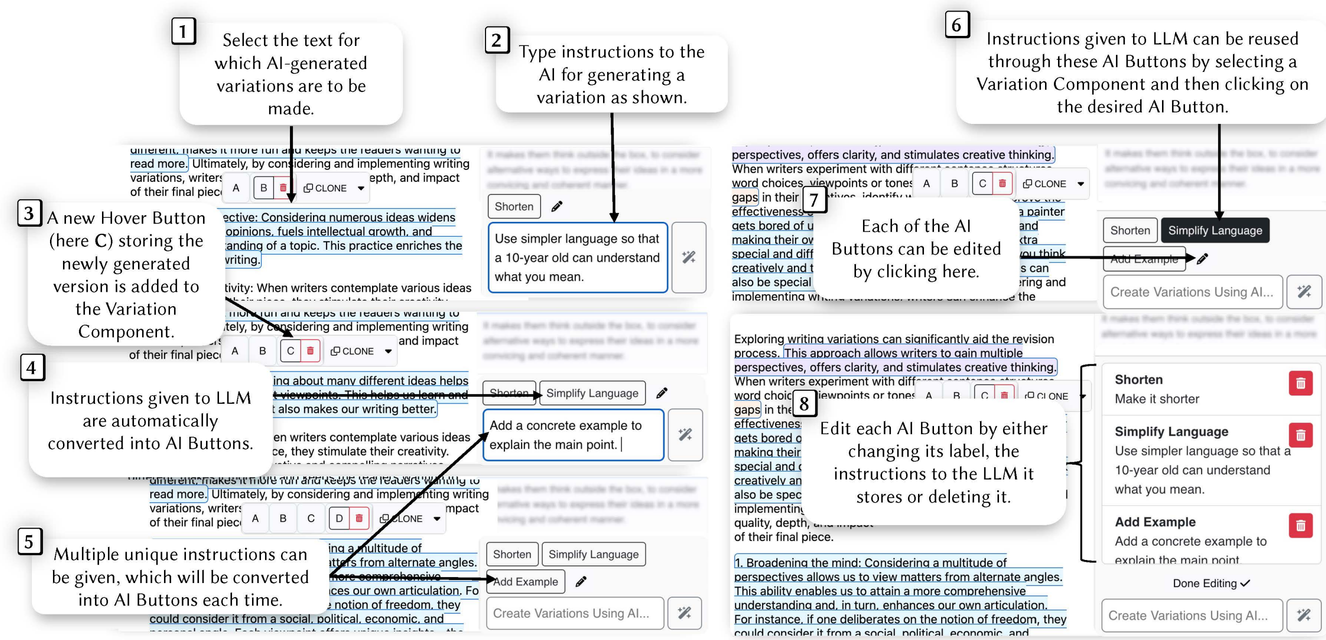

Figure 4: AI Buttons: Variation Components can also be edited using the AI buttons, which lets users specify alterations for an chunk. Descriptive labels are automatically generated for each AI button and each button can be reused and edited.

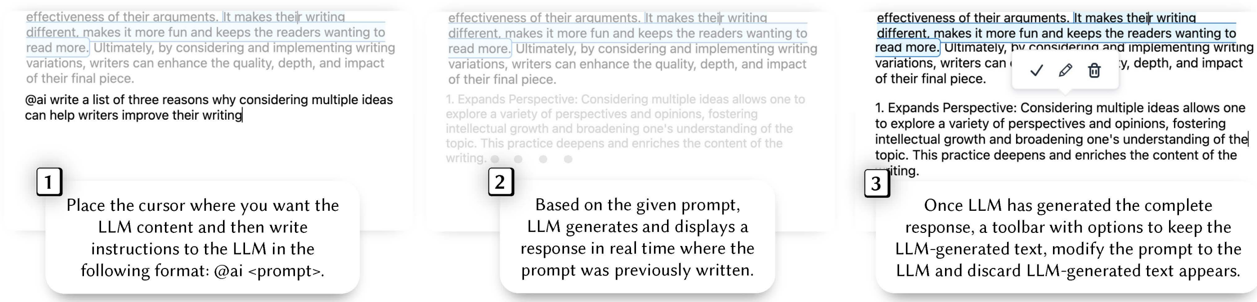

Figure 5: AI Insert: The AI Insert feature provides the ability insert LLM-generated text directly into the document, providing tighter integration between the Human and AI generated writing workflow. Users can see the AI generated content in real-time and choose insert or delete the output, or revise the prompt, giving users more control over what is included in their document.

AI editors such as Notion.AI[4]. To minimize potential confounding factors due to tangential differences between the study conditions, we maintained the same overall layout for the common UI elements such as the width of the sidebar, placement and dimensions of buttons and text, font size, and color, and used the same underlying LLM model, GPT-4, to implement the generative AI features. We sought to answer the following research questions:

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?

4.1 Participants

We recruited 12 writers (5 women, 7 men), aged 18 to 34, all of whom reported proficiency in reading and writing in English. They were screened for prior experience in a broad variety of both fiction and non-fiction writing genres. We included a range of prior experience levels of AI tool usage. This was done to ensure that our findings were not tied to specific genres or specific AI usage traits. Moreover, all participants had sufficient writing experience to be able to comment on the revision process and its applicability to different kinds of writing. See Table 1 for writer profiles.

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 drafting a social media post—sought to provide an ecologically valid writing experience that fits timing constraints of the study and serves as a realistic use-case for LLMs. Such scenarios align with recent HCI studies on human-AI co-writing [25, 37]. Additionally, commercial AI-assisted writing tools, such as Grammarly Go, Respondable, and Copy.ai, leverage AI for similar purposes.

Scenario Descriptions:

- **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.

Then, using either the ABScribe or the Baseline interface, participants were asked to explore 8 variations (increasing or decreasing length, formality, word diversity, adding emojis, and two variations of their choosing) of 3 distinct text segments (title/subjectline, third sentence of the second paragraph, entire third paragraph),

[4] notion.ai

ID	Writing Experience	AI-Assisted Tools Usage
W1	***Moderate:*** Worked as a staff writer for two political science publications. Writes fiction as a hobby.	***Moderate:*** 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	***Moderate:*** Taught ESL courses to non-native English speakers, specializing in IELTS, TOEFL, and business English instruction.	***Experienced:*** Uses ChatGPT by feeding it the main points of the article to generate a draft, and then editing the responses provided by ChatGPT.
W3	***Highly Experienced:*** 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.	***Moderate:*** 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	***Highly Experienced:*** 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.	***Limited:*** 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	***Experienced:*** 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.	***Experienced:*** 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	***Experienced:*** 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.	***Limited:*** Used ChaptGPT in a very limited capacity, mostly to brainstorm assignment structures and topic sentences when writing.
W7	***Experienced:*** 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.	***Limited***: Briefly experimented with ChatGPT.
W8	***Moderate:*** Written mostly technical papers, but also wrote some short stories as a hobby.	***Moderate:*** Uses ChatGPT every other day mostly to proofread, create templates for texts, and find the right creative direction when writing.
W9	***Experienced:*** Usually writes research papers in computing education and blogs. Blogs usually cover personal experiences at work as well as hobbies.	***Limited:*** Used Grammarly and ChatGPT to edit writing.
W10	***Moderate:*** Wrote some column articles for personal social media accounts, and several research papers over the past five years.	***Experienced:*** 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	***Highly Experienced:*** Focuses on academic writing such as papers, scholarship applications, reviewing, etc.	***Advanced:*** 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	***Experienced:*** Engages in hobby novel writing, academic writing, blog writing	***None:*** 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.

summing up to 24 variations. This approach ensured exploration of variations 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.

4.3 Measures

To assess subjective task-workload, we used the widely used NASA-TLX [31] 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 [37, 40, 41]:

(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. Conducted via videoconference, the entire study lasted approximately 1.5 hours. 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 various writing variations using LLM-integrated editing tools—participants undertook two 15-minute 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 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 revision process that we seek to improve through our design. The evaluation concluded with a recorded 30-minute semi-structured user interview on their experience with each interface. As a token of our appreciation for their participation, each participant received 30 Canadian dollars.

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 [9] 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-subject design, we use pairwise one-sided t-tests to compare *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. [15]. Additionally, we aim to check for normality. We aimed to determine if ABScribe presented significant improvements over the baseline, prompting us to select a one-tailed

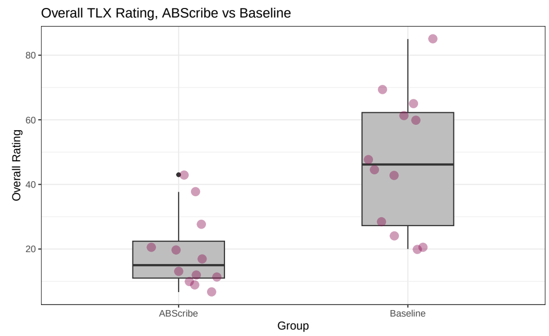

Figure 6: Overall results on NASA TLX subjective task workload

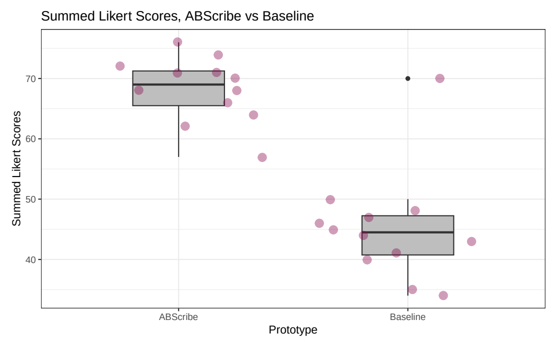

Figure 7: Summed Likert ratings for users' perceptions of the revision process

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 apriori power analysis for a pairwise one-sided t-test, showing that we can detect with 80% power at least $d = 0.8$ effect size with sample size $N = 12$ participants for a significance level $\alpha = 0.05$.

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." – W1*

The overall response to ABScribe, as exemplfied by 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 (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* 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* groups findings relating to the ease of specific tasks during the writing and revision process as well the ease of specific interactions with the interface to accomplish those tasks.

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

F1: 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 storing multiple variations within AB Text Components without cluttering the document, and the ability to switch between variations using the Hover Buttons and the Variation Accordion made some participants feel less pressured to commit to an initial variation before considering multiple options. For example, W3 noted: *"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 [Baseline] 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 AB Components store variations and the ability to reuse prompts using the automatically generated Variation Buttons meant that they could create, revise and evaluate a larger number of variations with ABScribe. W12 noted that storing variations 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: 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 when using ABScribe more easily than Baseline. 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."*

Being able to work with smaller text-segments, however, made some participants worry about the overall coherence of the passage. 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 in-place, 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 of the text, unless you want to perform a single edit and immediate make that change."*

F3: 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 Variation Buttons 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."* 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 p*lease make this variation shorter?"* 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 [5].

F4: 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."*

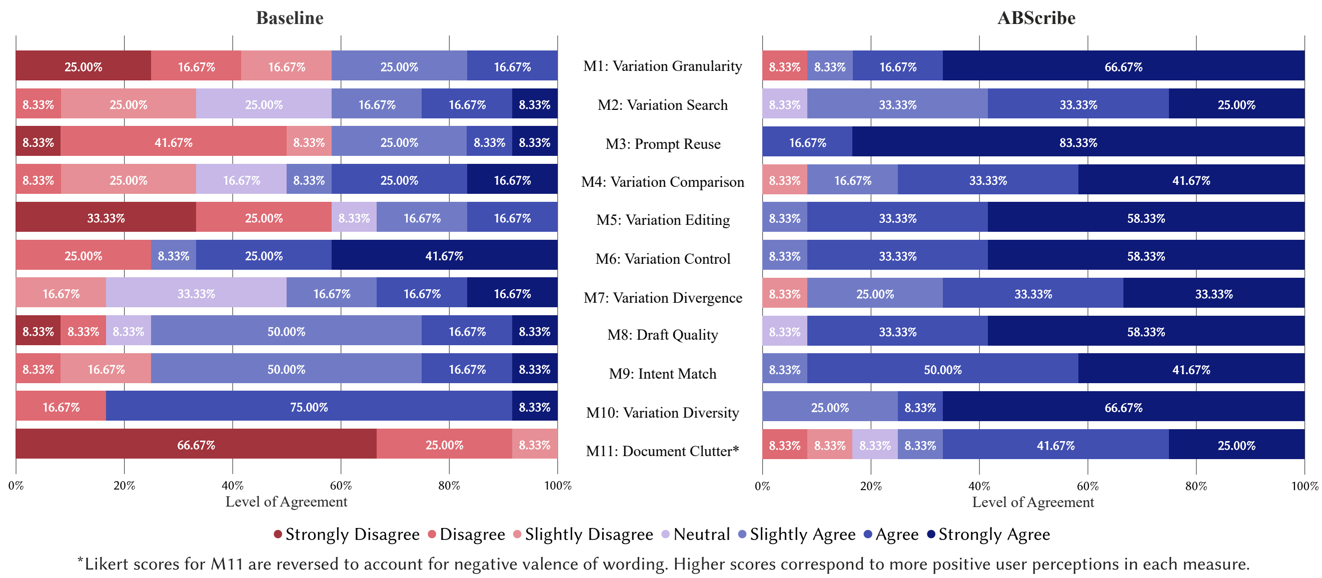

Figure 8: Responses to Likert-Scale Measure on the Revision Process for Exploring Multiple Writing Variations. Higher the agreement level, the more positive the user perception.

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 [ABScribe] 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 ABScribe 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."* Decomposing custom prompts that only apply to a specific text segment into more general, more atomic prompts that apply across variations encouraged participants to *combine* multiple prompts by clicking multiple Variation Buttons. 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 [ABScribe]."*

5.2 RQ2: Subjective Task Workload

Participants pointed to four aspects of the revision workload in ABScribe that led to the significant reduction in subjective task workload: reduction in clutter, ease of variation management, access to surrounding text context, and reduced context-switching during LLM use.

F5: ABScribe reduces task workload by reducing document clutter. W4, 5, 7, 10, 11, commented that the general lack of clutter in the ABScribe interface when dealing with multiple versions affected various aspects of their writing and revision process. Specifically, the way ABScribe stores variations in-place with the ABToolbar as opposed to in-sequence in the doucument 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 ABScribe to the linear storage of variations 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."*

Although all participants agreed that the ABScribe interface produced a less cluttered, in contrast to what we expected, two of them perceived clutter as being not necessarily bad. W2 and W6, who were both very comfortable with the worklow afforded by Baseline as it was similar to what they were used to doing, 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]."* 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 because 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 ABScribe. 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: ABScribe reduces task workload by enhancing variation management. Two major recurring activities in the exploration of multiple variations were variation storage, or tracking variation history, as W7 referred to it, and the comparison of multiple variations. W3, 6, 7, and 11 noted that the non-linear storage of the ABScribe interface necessitated less overhead to manage and revise multiple variations. W3 commented that by having the ABScribe interface manage storing variations for them, 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 Hover Buttons enhanced the ease with which they compared variations: *"the feature made it easier to 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."*

Some users noted that there were some instances where comparing larger text-segments was easier to do in-sequence, and suggested that the Variation Accordion interface could also serve the purpose of in-sequence comparison. W3 notes this: *"Yeah, I mean, one nice thing about Baseline is that I do get to see all of the versions together. So they're all listed for me...but 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 - for context, and for clutter, I think A is vastly superior."*

F7: ABScribe reduces task workload by showing variations in context of the surrounding text during manual editing and LLM use. The need for considering context during the revision process came up during several interviews. W3-8, and 10 commented that thanks to the in-place comparison of variations, they were able to see what a variation looks like in a paragraphs, as exemplified by W10's comment: *"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? 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."*

W7 echoes this preference for the non-linear ABScribe interface: *"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: ABScribe's variation storage and in-place LLM revision reduces subjective task workload . Interactions that brings 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 the baseline interface leads to these sorts of interactions: *"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 managing different variations of text when using the baseline or outside of the study that required them to leave the primary text editing interface to perform comparison of variations. This was either to copy paste different versions from separate documents into the primary text editing interface for comparison in sequence, or simply to compare variations side by side in separate documents. W5, 7, and 9 all found the ABScribe interface less cumbersome, 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."*

F9: 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 buttons 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 buttons *"much more streamlined."* W3 sums up the interaction concisely in the following quote: *"A [ABScribe] is vastly superior for reusability, there is no question. B [Baseline], 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

In this work, we present the design and implementation of ABScribe which is composed of five key design elements: Variation Components, Hover Buttons, the Variation Accordion, AI Buttons, and AI Insert. Our comparative evaluation study shows that these elements reduce task workload and significantly enhance user perceptions of the revision process when managing multiple variations of text segments in human-AI co-writing tasks as compared to a familiar baseline editing system (Section 4).

We present six findings (F1-6, Sections 5.1 and 5.2) that provide evidence for the efficacy of our design elements. These findings offer insights into how writers utilize both chat-based and in-place Human-AI co-writing interfaces (Sections 2.3, 5.1). They also illustrate the affordances and limitations of these interface types for exploring multiple writing variations, their influence on the size, granularity, quantity, and diversity of variations, as well as the prompt style users take on.

6.1 Non-Linear Text Revision Control

In ABScribe, the Variation Component, and dynamically placed Hover Buttons provide an effective approach to managing text variations non-linearly. This offers an alternative to the linear text revision control features found in current editing interfaces and chat-based AI-assistants.

We find distinct influences on task workload (Section 5.2) and the style of prompts created by users between these two methods (F3-4, Section 5.1). Our interface is grounded in a non-linear nature of the writing and editing process [56, 59], offering a way for designers to support multiple fine-grained variations without overwhelming users (F2, Section 5.1). As our design for the Hover Buttons and Variation Components builds upon a familiar rich-text editing interface, they can potentially be integrated into existing document editors without major layout changes. This would enable writers to work more closely with LLM-based generative AI content, within the context of the surrounding text (F7, Section 5.2).

We also observed that the non linear approach affords advantages such as viewing, editing, and combining multiple variations together. Notably, participants W1 and W4 highlighted the usefulness of the non-linear text revision control even without AI (F6, Section 5.2), indicating the broader relevance of the Hover Button and Variation Component interface overlay for the wider range of text editors.

6.2 Scaffolding Prompts Focused Around Specific Writing Tasks

Our design for AI Buttons demonstrates how scaffolding LLM prompts around interface elements can influence user prompt-writing behavior. Specifically, the AI buttons, auto-generated after writers create a prompt for a text-segment variation, encourages writers to craft more direct, imperative prompts (F3, Section 5.1). It also made them reflect more on the revision process, shifting the focus from conversing with a chatbot to designing a button that represents their writing style for reuse across variations (F9, Section 5.2). We noticed a trade-off: while making prompts more reusable, it nudged users away from conversational prompts tied to a specific variation's nuances (F4, Section 5.1). For example, they were more polite in the chat-based interface and direct when using the button scaffold. This highlights the importance of considering how different UI scaffolds influence the kinds of prompts users write.

These findings contribute to the body of research on encouraging writers' experimentation of LLM output using modular LLM interfaces [29], and expands the scope of valuable design paradigms in text-editing, such as the use of reified text selections [6, 27, 28], to supporting the exploration of multiple variations in LLM-based revision-focused writing.

6.3 Moving Beyond Systematic Exploration to Systematic Evaluation of Variations

Our interface elements could be further developed to support the systematic evaluation of variations. While we want to avoid premature elimination of rough ideas [11, 26], we also want to afford critical assessment of variations in relation to each other [11, 60]. For instance, if we develop an extension that lets writers test different versions of the ABScribe draft containing a randomized subset of variations, they could link our interface to an open-source A/B testing framework like MOOClets [54], UpGrade [49] or Planout [4]. This would allow writers to quantitatively evaluate which versions best meet their objectives based on specific metrics. A copywriter, for example, wanting to rapidly explore and assess different advertisement versions, can use ABScribe to explore draft variations, compare variations in the draft's context using Hover Buttons, select a subset for further evaluation, and run randomized online experiments via an A/B testing framework. Moving beyond systematic exploration, which our current study covers, towards systematic evaluation is a logical next step. This opens a diverse design space for A/B authoring tools that simplify designing variations for evaluation, making it a promising avenue for future research.

Our design seeks to enhance the natural flow of writing and editing texts by supporting close collaboration between humans and AI, grounded in established theories on the revision process. As highlighted in Section 2.1, revising text is not merely about superficial changes [19], it delves into deeper subprocesses like idea formulation [20] and meaning discovery [59]. Skilled writers view revision as a cyclical and recursive process [56, 59], diving into the content repeatedly with different aims such as shaping a persuasive narrative [59], playing with word choices and tone [34], and switching between various writing tasks during text revision [18]. Our interface provides a concrete example of affordances that editing interface designers can leverage to make the Human-AI collaboration in revisions more congruent with established revision theory.

In Section 2.1, we hypothesized that improving ease of use for handling multiple variations could help us effectively apply HCI design principles, affording the user greater freedom in experimenting with variations, thereby avoiding premature fixation on an initial idea [35, 60, 60]. As we found in F1 (5.1), the design of ABScribe helps writers to not commit to an initial idea too early, and instead, explore a greater quantity of variations before evaluating them, providing evidence for the applicability of parallel exploration in text revision.

6.4 Limitations

Our work has two limitations to external validity, which are common in lab-based evaluation studies. First, our evaluation study was restricted to English writers. Although the underlying LLM, GPT-4, supports multiple languages [50], suggesting that our interface can extend beyond English, our study focused on English writing tasks. Without specifically studying how different languages influence task workload and user perceptions of the revision process, we cannot comment on the wider applicability of our tools, and 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. We crafted realistic writing tasks that could be completed within the study's timeframe, representing use cases from prior studies and commercial AI apps. Ideally, writers would select their own writing task and spend an extended period, possibly spanning several days or even a week, to revise and explore variations.

To address some of these concerns, we complemented the writing tasks with in-depth user interviews, allowing writers to reflect on and discuss the implications of our design beyond the study's writing tasks. Several participants, such as W1, 3, 4, 11, and 12, expressed interest in using our tool for creative and academic writing, in settings beyond the use cases we explored.

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 interface is composed of an ensemble of five distinct elements: Variation Components, Variation Accordion, Hover Buttons, AI Buttons, and AI Insert. Collectively, these elements not only markedly decrease task workload ($d = 1.20, p < 0.001$) but also bolster user perceptions of the revision process ($d = 2.41, p < 0.001$), in comparison to a popular AI-integrated editing workflow consisting of a rich text editor augmented with a chat-based AI assistant and the ability to insert AI generated content.

Our evaluation with writers (N=12) validate the efficacy of our design 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.

REFERENCES

[1] 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/.

[2] 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt.

[3] 2023. PaLM 2 Technical Report. https://ai.google/static/documents/palm2techreport.pdf.

[4] Eytan Bakshy, Dean Eckles, and Michael S Bernstein. 2014. Designing and deploying online field experiments. In *Proceedings of the 23rd international conference on World wide web*. 283–292.

[5] 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). https://doi.org/10.5210/spir.v2019i0.10923

[6] 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

[7] Sara Bly. 2007. Fundamentals in HCI: Learning the Value of Consistency and User Models. (2007).

[8] 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]

[9] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla 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.

[11] Bill Buxton. 2010. *Sketching user experiences: getting the design right and the right design*. Morgan kaufmann.

[12] N Ann Chenoweth. 1987. The need to teach rewriting. *ELT journal* 41, 1 (1987), 25–29.

[13] Yin Ling Cheung. 2016. Teaching writing. *English language teaching today: Linking theory and practice* (2016), 179–194.

[14] Robert Dale. 2021. GPT-3: What's it good for? *Natural Language Engineering* 27, 1 (2021), 113–118.

[15] 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.

[16] 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.

[17] 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

[18] Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College composition and communication* 32, 4 (1981), 400–414.

[19] Jill Fitzgerald. 1987. Research on revision in writing. *Review of educational research* 57, 4 (1987), 481–506.

[20] Linda Flower and John R Hayes. 1981. A cognitive process theory of writing. *College composition and communication* 32, 4 (1981), 365–387.

[21] 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

[22] Louie Giray. 2023. Prompt Engineering with ChatGPT: A Guide for Academic Writers. *Annals of Biomedical Engineering* (2023), 1–5.

[23] Natalie Goldberg. 2016. *Writing down the bones: Freeing the writer within*. Shambhala Publications.

[24] 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.

[25] 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*. 1–18.

[26] 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

[27] Han L Han. 2020. Designing Representations for Digital Documents. In *Adjunct Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 174–178.

[28] 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*. 1–13.

[29] 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

[30] Joseph Harris. 2017. *Rewriting: How to do things with texts*. University Press of Colorado.

[31] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.

[32] 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*. 1–12.

[33] Ernest Hemingway. 2014. *Moveable feast: the restored edition*. Simon and Schuster.

[34] Roland K Huff. 1983. Teaching revision: A model of the drafting process. *College English* 45, 8 (1983), 800–816.

[35] David G Jansson and Steven M Smith. 1991. Design fixation. *Design studies* 12, 1 (1991), 3–11.

[36] 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 *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–8.

[37] 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. (2023).

[38] 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.

[39] Jaewook Lee, Jaylin Herskovitz, Yi-Hao Peng, and Anhong Guo. 2022. Image-Explorer: Multi-Layered Touch Exploration to Encourage Skepticism Towards Imperfect AI-Generated Image Captions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–15.

[40] 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*. 1–19.

[41] 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. *arXiv preprint arXiv:2212.09746* (2022).

[42] Carrie S Leverenz. 2014. Design thinking and the wicked problem of teaching writing. *Computers and Composition* 33 (2014), 1–12.

[43] 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

[44] Charles A MacArthur. 2018. Evaluation and revision. *Best practices in writing instruction* 287 (2018).

[45] 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

[46] 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.

[47] 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, 554–578.

[48] John McPhee. 2017. *Draft No. 4: On the writing process*. Farrar, Straus and Giroux.

[49] April Murphy, Stephen E Fancsali, and Steven Ritter. [n. d.]. UpGrade: Scalable Digital Experimentation in Authentic Learning Settings. ([n. d.]).

[50] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[51] Sundar Pichai. 2023. An important next step on our AI journey. https://blog.google/technology/ai/bard-google-ai-search-updates/.

[52] James P Purdy. 2014. What can design thinking offer writing studies? *College Composition and Communication* (2014), 612–641.

[53] 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

[54] Mohi Reza, Juho Kim, Ananya Bhattacharjee, Anna N. Rafferty, and Joseph Jay Williams. 2021. The MOOClet Framework: Unifying Experimentation, Dynamic Improvement, and Personalization in Online Courses. In *Proceedings of the Eighth ACM Conference on Learning @ Scale* (Virtual Event, Germany) *(L@S '21)*. Association for Computing Machinery, New York, NY, USA, 15–26. https://doi.org/10.1145/3430895.3460128

[55] Horst Rittel. 1967. Wicked problems. *Management Science,(December 1967)* 4, 14 (1967).

[56] Anthony Seow. 2002. The writing process and process writing. *Methodology in language teaching: An anthology of current practice* 315 (2002), 320.

[57] 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*. 163–166.

[58] Mike Sharples. 2002. *How we write: Writing as creative design*. Routledge.

[59] Nancy Sommers. 1980. Revision strategies of student writers and experienced adult writers. *College composition and communication* 31, 4 (1980), 378–388.

[60] 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

[61] Hugo Touvron, Louis Martin, and Kevin Stone. [n. d.]. Llama 2: Open Foundation and Fine-Tuned Chat Models. ([n. d.]).

[62] Yi Wu, Nan Jiang, Hung Viet Pham, Thibaud Lutellier, Jordan Davis, Lin Tan, Petr Babkin, and Sameena Shah. 2023. How Effective Are Neural Networks for Fixing Security Vulnerabilities. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Seattle, WA, USA) *(ISSTA 2023)*. Association for Computing Machinery, New York, NY, USA, 1282–1294. https://doi.org/10.1145/3597926.3598135

[63] 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

[64] 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

[65] William Zinsser. 2006. On writing well: The classic guide to writing nonfiction. *New York, NY* (2006).