Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

Using Large Language Models to Support Authorship of DIY Tutorials

Master's Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Shailesh Vaidheeswaran Iyer

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Ulrik Schroeder

Registration date: 24.11.2023
Submission date: 24.05.2024

Zentrales Prüfungsamt/Central Examination Office

RWTH AACHEN UNIVERSITY

Eidesstattliche Versicherung
Statutory Declaration in Lieu of an Oath

Name, Vorname/Last Name, First Name

Matrikelnummer (freiwillige Angabe)
Matriculation No. (optional)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel
I hereby declare in lieu of an oath that I have completed the present paper/Bachelor thesis/Master thesis* entitled

selbstständig und ohne unzulässige fremde Hilfe (insbes. akademisches Ghostwriting) erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
independently and without illegitimate assistance from third parties (such as academic ghostwriters). I have used no other than the specified sources and aids. In case that the thesis is additionally submitted in an electronic format, I declare that the written and electronic versions are fully identical. The thesis has not been submitted to any examination body in this, or similar, form.

Ort, Datum/City, Date

Unterschrift/Signature

*Nichtzutreffendes bitte streichen

*Please delete as appropriate

Belehrung:
Official Notification:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

Para. 156 StGB (German Criminal Code): False Statutory Declarations

Whoever before a public authority competent to administer statutory declarations falsely makes such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment not exceeding three years or a fine.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Para. 161 StGB (German Criminal Code): False Statutory Declarations Due to Negligence

(1) If a person commits one of the offences listed in sections 154 through 156 negligently the penalty shall be imprisonment not exceeding one year or a fine.

(2) The offender shall be exempt from liability if he or she corrects their false testimony in time. The provisions of section 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:
I have read and understood the above official notification:

Ort, Datum/City, Date

Unterschrift/Signature

Contents

List of Figures

List of Tables

Abstract

DIY tutorials are essential for knowledge exchange in the DIY community. Sometimes the poor quality of DIY tutorials leads to mistakes, misunderstandings, and safety risks. The latest generation of ”large language models” (LLMs), like GPT-4 and Gemini, showcase broad capabilities useful for supporting writing tasks. This thesis investigates how LLMs might aid the writing of high-quality DIY tutorials.

We built DIYmate, a text editor in which users collaborate with a generative model to write a DIY tutorial. We evaluated DIYmate with a user study (N=14) in which participants were tasked to complete a DIY project and then write a DIY tutorial using DIYmate. We then conducted a post-task interview about their experience using DIYmate. We collected quantitative data regarding their usage of DIYmate and qualitative data through semi-structured interviews. The results show that participants found the LLM helpful in planning and writing the DIY tutorial. However, reviews from the LLM are not as helpful. Based on these results, we recommend guidelines for prompt design and propose how the LLM-based controls can be better integrated to improve the quality DIY of tutorials.

Überblick

DIY-Tutorials sind unverzichtbar für den Wissensaustausch in der DIY-Community. Allerdings kann bei DIY-Tutorials von schlechter Qualität dies teilweise zu Fehlern, Missverständnissen und Sicherheitsrisiken führen. Die neueste Generation der "large language models" (LLMs)(deutsch: große Sprachmodelle) wie GPT-4 und Gemini verfügen über umfassende Fähigkeiten, die für die Unterstützung von Schreibaufgaben nützlich sein können. In dieser Masterarbeit untersuchen wir , wie LLMs den Prozess des Schreibens hochwertiger DIY-Tutorials unterstützen können.

Dafür bauten wir DIYmate, einen Texteditor, in dem Benutzer mit einem generativen Modell zusammenarbeiten, um ein DIY-Tutorial zu schreiben. Wir haben DIYmate durch eine Benutzerstudie (N=14) evaluiert, bei der die Teilnehmer ein DIY-Projekt abschließen sollten, um dann mit DIYMate ein DIY-Tutorial zu schreiben. Anschließend, führten wir nach Vollendung der Aufgabe ein Interview über ihre Erfahrungen mit DIYmate. Wir haben sowohl quantitative Daten zu ihrem Nutzungsverhalten gesammelt, sowie auch qualitative Daten in Form eines Interviews.Die Ergebnisse zeigen, dass die Teilnehmer das LLM bei der Planung und Erstellung des DIY-Tutorials hilfreich fanden. Allerdings sind die Bewertungen des LLM nicht so hilfreich. Basierend auf diesen Ergebnissen empfehlen wir Richtlinien für die Gestaltung von Prompts und schlagen vor, wie die LLM-basierten Kontrollen besser integriert werden können, um die Qualität der DIY-Tutorials zu verbessern.

Acknowledgements

First, I would like to thank Prof. Dr. Jan Borchers and Prof. Ulrik Schroeder for examining this thesis.

Special thanks to my supervisor, Marcel Lahaye, for his invaluable motivation, support, and guidance.

I would like to further take the opportunity to thank all of my colleagues working at Lehrstuhl für Informatik 10 for all the little things that helped me refine my thesis, as well as for feedback about my work.

I want to express gratitude to all my study participants for volunteering their time to participate in my study.

A big thank you to to Sijia Guo and Jonte Hünerbein for their assistance in proofreading this thesis. Also, a special thanks to Elisabeth Buttkus for her help with the German abstract.

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

EXCURSUS:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition:
Excursus

Source code and implementation symbols are written in typewriter-style text.

`myClass`

The whole thesis is written in American English. For the first-person, the pronoun ”we” is used and for the third-person, the pronoun ”they” is used.

Download links are set off in coloured boxes.

File: myFile[a]

[a] http://hci.rwth-aachen.de/public/folder/file_number.file

Chapter 1

Introduction

DIY culture and makers

The DIY (Do-It-Yourself) ethos has roots that extend beyond recorded history, closely linked to the fundamental human need to create, mend, and modify tools and materials for survival [Kuznetsov and Paulos, 2010]. In the 21st century, individuals passionate about crafting, creating objects, and sharing their know-how have come to be known as makers [Boeva and Troxler, 2020]. They engage in various pursuits ranging from artistic endeavors to digital fabrication. These enthusiasts often gather in DIY platforms and Makerspaces, exchanging techniques and insights for constructing via DIY Tutorials, altering, and tinkering with various artifacts [Buechley et al., 2009]. DIY communities flourish through a unique ethos of creativity, collaboration, and open sharing, standing in contrast to mass production, profit, and social capital [Kuznetsov and Paulos, 2010].

DIY TUTORIAL:
DIY tutorials explain the components, tools, and processes required to make DIY projects.[Wakkary et al. 2015]

Definition:
DIY Tutorial

Tutorials expand and sustain DIY culture

Wakkary et al. [2015] notes that the practice of writing and sharing DIY tutorials lies at the center of the distributed peer-production and creativity of DIY. Tutorials provide tutorship for particular projects and help develop the skills and competencies of those involved in DIY. In doing so, it helps expand and sustain the culture and practices of DIY.

Poor quality of tutorials and complexity of DIY tutorials

Online DIY tutorials frequently exhibit quality issues, including incomplete lists of material, poorly sequenced instructions, and bad images and text formatting [Wakkary et al., 2015]. Writing a DIY tutorial is a complex task. Writers need to balance concerns of accuracy, technical aspects of creation, assessing the skill levels of the audience, and portraying detailed and clear information.

DIY tutorial writing and practices are distinct from Technical writing and practices

While DIY tutorials are often compared to the professional discipline of technical writing and would benefit from taking professional input. DIY practices of everyday design are often in contradiction to technical writing and see themselves as distinct from professional practices of professional technical writing and thus do not seek professional input [Desjardins and Wakkary, 2013,Phillips et al., 2013].

LLMs offer an opportunity to address

Large language models (LLMs) offer the opportunity to address the issues of quality in DIY tutorials. The latest generation of LLMs such as GPT-4[1] and Gemini[2] demonstrate impressive capabilities without the need of extensive training for specific tasks. However, the initial focus of research has generally been on creative writing, such as fiction writing [Calderwood et al., 2020], theatre scripts, and screenplay [Mirowski et al., 2023]. They have not been explored as much in the realm of DIY tutorial writing.

Integrating LLMs into a cohesive system of interaction through design

The utility of these models is in producing human-usable outputs and they show great promise in generating context-relevant text and writing step-by-step instructions [Zhao et al., 2023]. However, it is crucial how these models are integrated into writing experiences to ensure a smooth flow for the writer in the writing process and in a collaborative manner with the model [Yuan et al., 2022 and Lee et al., 2022].

Motivation for this thesis

This thesis explores the intersection of LLMs with DIY tutorial writing. We investigate how various LLMs-based controls can assist a maker in writing a DIY tutorial by building a prototype system, DIYmate. We then evaluate this proto-

[1] https://openai.com/index/gpt-4-research/

[2] https://deepmind.google/technologies/gemini/#introduction

type by conducting a qualitative study in which we task makers to complete a DIY Project and write a DIY tutorial using DIYmate. This study aims to understand how LLMs can be used to support writing high-quality DIY tutorials.

RQ : How can we use LLMs to assist DIY tutorial authors in writing high-quality DIY tutorials?

1.1 Outline

In chapter 2 "Related work", we present the related work in three parts: a cognitive process model of writing, DIY tutorials, and the design space of .

In chapter 3 "Design and Implementation", we discuss the design and implementation of our LLM-based prototype system, DIYmate.

In chapter 4 "User Study and Results", we present the user study used to evaluate the prototype and its results.

In chapter 5 "Discussion", we discuss the results from our user study and propose some design guidelines based on it.

We conclude our thesis in chapter 6 "Summary and future work". We summarize our findings and their limitations and suggest directions for future research.

Chapter 2

Related work

In this Chapter, We have structured the related work into three parts. First, in section 2.1, we will look at the cognitive process model of writing and the writing goals design space based on it. Second, in section 2.2, we will look at the challenges of DIY tutorial authorship and recommendations to improve the quality of DIY tutorials. Third, in section 2.3, we will take a look at LLMs, how they can be considered in the task of writing, and different UX patterns for controllable text generation. We then conclude the chapter by presenting the gap this thesis explores in the literature.

2.1 Cognitive model of writing and design space

Cognitive process model of writing

Flower and Hayes [1981] proposed the cognitive process model of writing. They understood writing as an activity designed to create text for an audience on a specific topic. While Hayes [2012] later refined their model, we focus on the original model as shown in the figure 2.1. The cognitive process model deals with the individual writer. The task environment refers to the environment in which writing is taking place. It also highlights the importance of the writer's long-term memory, which the writer uses to

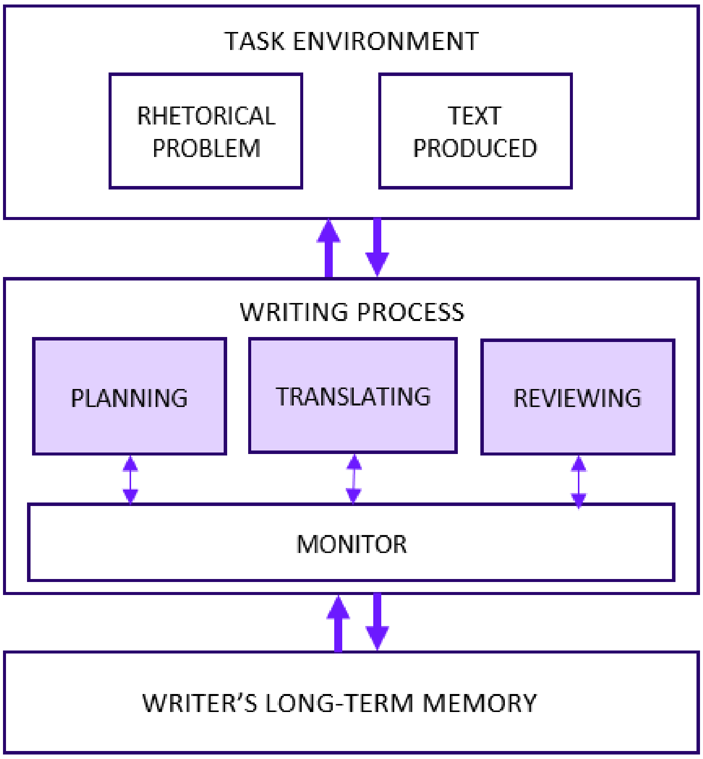


Figure 2.1: Cognitive process model of writing by Flower and Hayes [1981]

store facts about the world and their knowledge. Furthermore, writing is split into three distinct processes: planning, translating, and reviewing. Planning refers to the act of organizing ideas and setting goals. Translating refers to the act of converting ideas into words. Reviewing refers to the act of refining and revising the text. These processes don't have clear boundaries and can occur out of order. They are also hierarchical and recursive. For example, it could start with the goal of writing an academic text or an assignment, branching off into smaller and smaller goals, like writing a section of text or making a sentence sound more formal and clear.

Design space for writing support tools

Based on the cognitive process model of writing Gero et al. [2022] proposed a design space for writing support tools shown in the figure 2.2. The design space has two axes.

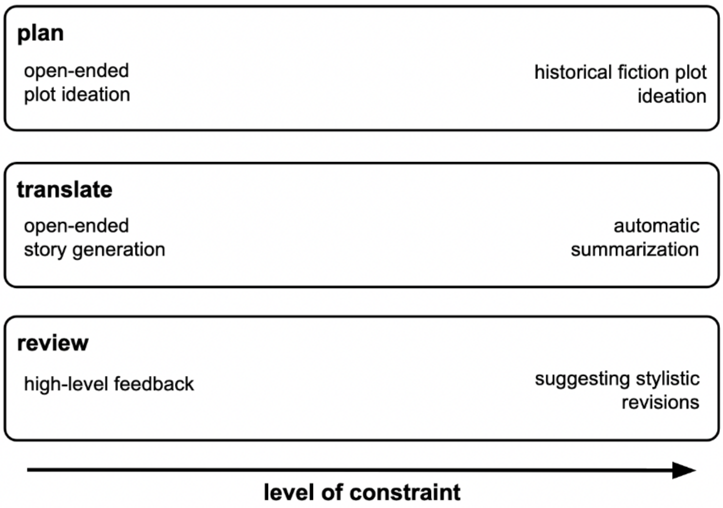


Figure 2.2: Writing goals design space. Figure taken from Gero et al. [2022]

On the horizontal axis, the level of constraint is the number of solutions a given problem has. The more constrained the goal of the task, the fewer solutions there are. On the vertical axis, the three processes from the cognitive process model of writing:Plan,Translate and Review. They found that highly constrained planning and reviewing are understudied. They recommended evaluating writing support tools by reporting more on actual writing done, using existing surveys, and reporting user interaction measures to make evaluations more comparable.

2.2 DIY Tutorial Authorship

Early research into DIY tutorial format

Dalton et al. [2014] investigated the qualities and drawbacks of the existing DIY tutorial format used for sharing DIY knowledge online. They translated an existing DIY tutorial into a minimal cookbook recipe-style format. By doing this, they realized that the recipe format can help reflect and rethink how to design tools to support DIY tutorials. The translation to a recipe format forces the DIY tutorial towards a minimalist, clear, and concise format. By constraining authors to the bare minimum, they proposed that it would lead to more uniform and accurate DIY tutorials.

Tseng and Resnick [2014] explored the design process documentation by DIY makers. They found that makers find the documentation process time-consuming and disruptive. The nature of the design process is often iterative, where makers try things, make mistakes, learn, and then fix things. They also surveyed makers and found the motivations for searching for DIY projects online, ranked in order of importance, to get ideas for a project, to learn a particular technique, and to look for projects to recreate. They concluded that makers might benefit more from sharing process documentation than finished tutorials. While also outlining some key requirements for process documentation in fab labs and maker spaces.

Issues in DIY tutorials - accurate information, competencies & tools, and tutorial format

Wakkary et al. [2015] analyzed the quality and effectiveness of ten DIY tutorials. They analyzed issues that lead to poor quality and poor effectiveness in DIY tutorials through three non-mutually exclusive categories. These categories include how well the interrelated complexity of competencies, components, and tools are represented in a DIY tutorial, how well instructions are sequenced in the DIY tutorial, and how well DIY instructions are communicated through images and text. They found that DIY tutorials are often inaccurate, often missing crucial information or incomplete lists of materials and tools. Furthermore, DIY tutorial authors have difficulty representing the complexities of competencies required for a specific DIY project and how that relates to the tools required to complete a DIY project. Lastly, DIY tutorials were often unclear and inconsistent in the quality of their communication and presentation of content, ranging from poor images to linking incompatible videos. These issues often lead to mistakes, misunderstandings, and safety risks for readers of a DIY Tutorial. Through their analysis, they provided practical guidance on how to improve the quality of DIY tutorials.

Build-Process oriented documentation of DIY projects

Tseng [2016] created build-in-progress, an online website where makers could share their stories of how their DIY projects develop. Makers could also look at and comment on other projects, offering advice and feedback. This

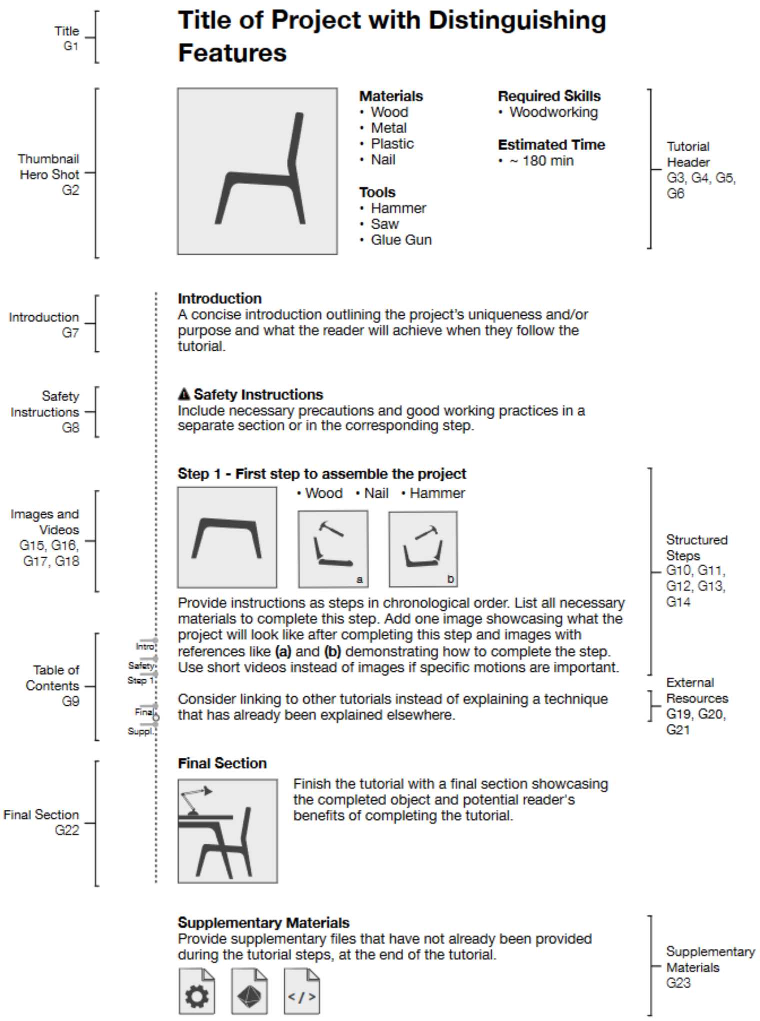

Title
G1
Title of Project with Distinguishing Features
Thumbnail
Hero Shot
G2
Materials
• Wood
• Metal
• Plastic
• Nail
Tools
• Hammer
• Saw
• Glue Gun
Required Skills
• Woodworking
Estimated Time
• ~ 180 min
Tutorial
Header
G3, G4, G5,
G6
Introduction
G7
Introduction
A concise introduction outlining the project's uniqueness and/or purpose and what the reader will achieve when they follow the tutorial.
Safety
Instructions
G8
⚠ Safety Instructions
Include necessary precautions and good working practices in a separate section or in the corresponding step.
Images and
Videos
G15, G16,
G17, G18
Step 1 - First step to assemble the project
• Wood • Nail • Hammer
a
b
Provide instructions as steps in chronological order. List all necessary materials to complete this step. Add one image showcasing what the project will look like after completing this step and images with references like (a) and (b) demonstrating how to complete the step. Use short videos instead of images if specific motions are important.
Structured
Steps
G10, G11,
G12, G13,
G14
Table of
Contents
G9
Intro
Safety
Step 1
Final
Suppl.
Consider linking to other tutorials instead of explaining a technique that has already been explained elsewhere.
External
Resources
G19, G20,
G21
Final Section
G22
Final Section
Finish the tutorial with a final section showcasing the completed object and potential reader's benefits of completing the tutorial.
Supplementary Materials
Provide supplementary files that have not already been provided during the tutorial steps, at the end of the tutorial.
</ >
Supplementary
Materials
G23

Figure 2.3: Tutorial template based on derived guidelines along with the applicable guideline highlighted along with the element they apply to by Lahaye et al. [2023]

helped the makers reflect on their DIY project and refine it. The website focused more on the process and less on the final product. However, in 2024, the website has been shut down. It still offers a glimpse into how DIY communities function and how makers learn from each other by providing feedback on each other's projects.

Tutorial authoring guidelines and ten high-level elements

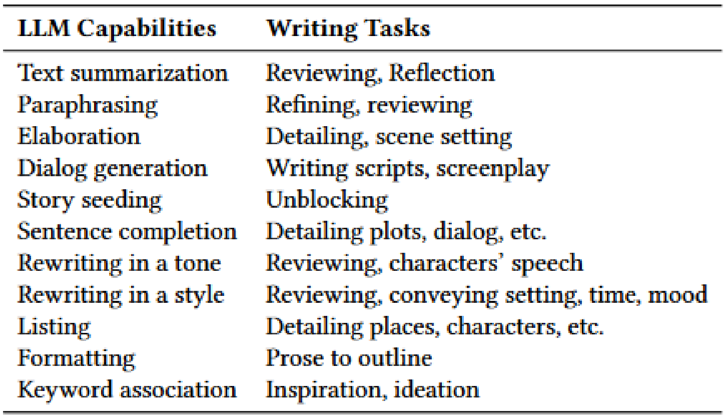
Lahaye et al. [2023] investigated the tutorial selection process of makers by interviewing them to understand what factors makers use to assess the quality of a tutorial. They found that makers look for ten high-level elements in the tutorial format: title, hero shot & thumbnail, tutorial header, introduction, safety instructions, table of contents, step-by-step structure, images & videos showing the crafting process, links to external tutorials, and final section. They also derived 23 guidelines for the DIY tutorial format that help understand the benefits of the ten elements. Through this, they laid the groundwork for research into DIY authoring systems and tools that can help DIY authors write DIY tutorials. Figure 2.3 shows an example tutorial format and the applicable guidelines for each element in the tutorial template.

2.3 Design space of Large Language Models

Large language models and prompt programming

Recent advancements in AI for natural language tasks were accelerated with the creation of the transformer architecture and focused attention mechanism by Vaswani et al. [2023]. As observed by Kaplan et al. [2020], the performance of language models improves when scaled up based on computation, model parameters, and training data size. Such scaled models are called large language models. Wei et al. [2022] observed a notable trait of LLMs is their emergent ability, wherein as models scale up, they show new and unique behavior not shown in smaller models. Zhao et al. [2023] identified three key emergent abilities of LLMs - in-context learning, instruction following, and step-by-step reasoning. Based on these emergent abilities, Reynolds and McDonell [2021] developed a theory of prompt programming; prompts are user inputs used to generate responses from the LLM. They further showed that zero-shot prompts can outperform few-shot prompts, where the model is given a few examples in the prompt to achieve a task.

Gmeiner and Yildirim [2023] explored the dimensions for designing LLM-based writing support. They identified



LLM Capabilities	Writing Tasks
Text summarization	Reviewing, Reflection
Paraphrasing	Refining, reviewing
Elaboration	Detailing, scene setting
Dialog generation	Writing scripts, screenplay
Story seeding	Unblocking
Sentence completion	Detailing plots, dialog, etc.
Rewriting in a tone	Reviewing, characters' speech
Rewriting in a style	Reviewing, conveying setting, time, mood
Listing	Detailing places, characters, etc.
Formatting	Prose to outline
Keyword association	Inspiration, ideation

Figure 2.4: Non-exhaustive list of LLM capabilities and writing tasks where they might be useful. Table taken from Gmeiner and Yildirim [2023]

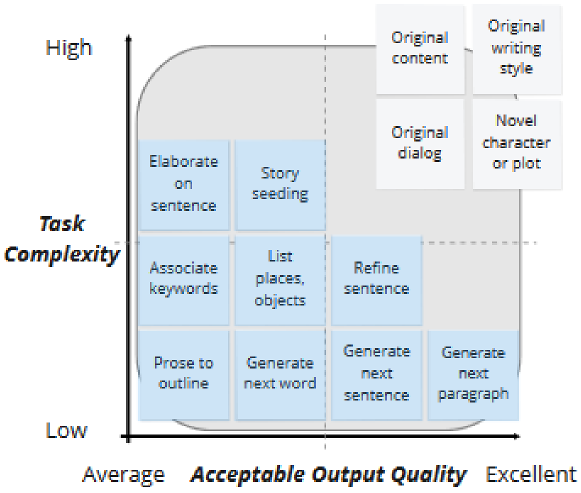


Figure 2.5: Task complexity - Output Quality matrix for LLM capabilities. Figure taken from Gmeiner and Yildirim [2023]

Dimensions for LLM-based writing support: LLM capabilities, task complexity & output quality

three key considerations when designing LLM-based writing support: LLM capabilities, writing task complexity, and output quality of generated text. They mapped different LLM capabilities to different writing tasks where they might be useful, as seen in figure 2.4. Their exploration revealed to them writing tasks where average quality outputs by LLMs are acceptable for a writing task as shown in figure 2.5.

Dataset for LLM-based collaborative writing

Lee et al. [2022] introduced Coauthor, a dataset, and an interface that can be used to replay the writing sessions. The dataset can be used to understand the generative capabilities of LLMs and improve interaction design for collaborative writing experiences.

User's should be able to edit system output

Calderwood et al. [2020] explored how novelists use generative language models in the fiction writing process. They found that writers found value in being able to edit the system's output with something they preferred. Thus, they advised that future systems should provide many suggestions, while understanding the intent of the writer, be editable, and regenerate with little to no mental overhead.

Suggestion lists are better than continuous text generation

Lehmann et al. [2022] conducted a study comparing suggestions lists from a generative AI to continuous generation of text on mobile devices. They found that the perceived ownership of text was lower, while effort for editing was higher when text was continuously generated compared to suggestions lists. Overall, AI generation increased the word length and user's perceived that it affected wording.

Wordcraft and UX patterns for controllable text generation

Yuan et al. [2022] presented Wordcraft, a text editor in which users collaborate with an LLM to write a story. They evaluated their prototype with a user study. Furthermore, they outlined different UX patterns for controllable text generation that they used in the Wordcraft interface. Since, we will be referring to these patterns in the next chapter. we will list and briefly explain them here:

- **Infilling** - users select a piece of text and ask the LLM to suggest alternatives.
- **Continuation** - users ask the LLM to continue the sen-

tence, paragraph, or story.

- **Elaboration** - users select a piece of text for the LLM to provide more details on.
- **Story-seeding** - users can ask the LLM to generate a plausible first sentence. This is also known as generation from scratch.
- **Free-form style transfer** - users can ask the LLM to rewrite a sentence in an arbitrary way.
- **Custom Prompting** - users can ask the LLM to perform arbitrary tasks on the fly.
- **Meta-Prompting** - users can ask the LLM for suggestions on what questions to ask the LLM itself.

Motivation for this thesis

The previous research on LLMs has mainly focused on creative writing. However, research into DIY tutorials is now moving towards providing recommendation guidelines for authoring tools. This thesis aims to explore the use of LLMs in creating high-quality tutorials. To achieve this, we will develop a prototype based on Wordcraft for authoring DIY tutorials and examine how different UX patterns can be applied to it. Subsequently, we will conduct a user study to evaluate how DIY tutorial authors interact with the prototype.

Chapter 3

Design and Implementation

In this chapter, we will detail the reasoning behind the design and implementation of the prototype, “DIYMate”. We will discuss which operations we created and implemented. Additionally, describe the user journey that users will go through in the prototype to write their DIY tutorial. The resulting prototype artifact ”DIYmate” is available under this link[1].

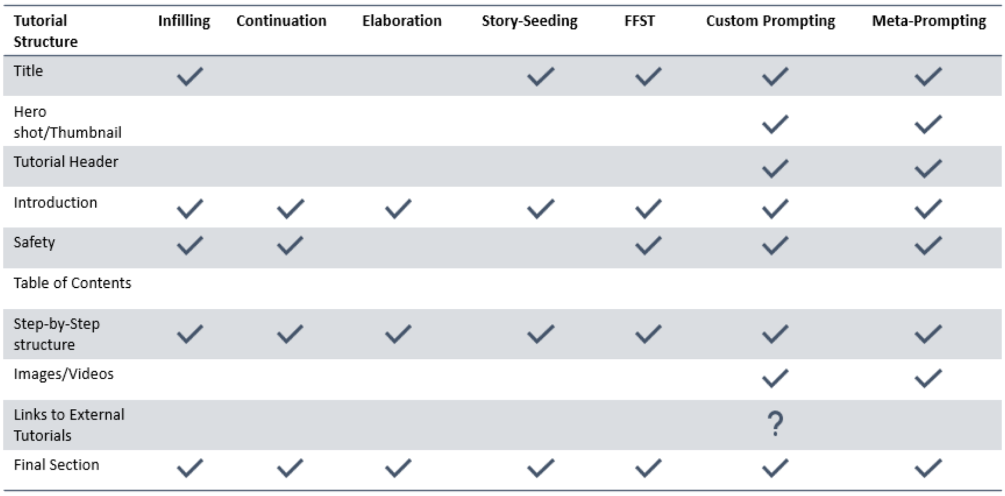
3.1 UX patterns for controllable text generation

We first began our design process by mapping the UX patterns for controllable text generation outlined by Yuan et al. [2022] to the ten high-level categories derived by Lahaye et al. [2023] for a DIY tutorial to figure out where we may apply those patterns as shown in the figure 3.1.

All UX patterns applicable for introduction, steps, and final section

We observed that for elements that contain a lot of text, namely the introduction, step-by-step structure, and the final section. All UX patterns for controllable text generation

[1] https://git.rwth-aachen.de/i10/thesis/thesis-shailesh-iyer-llm-diy-tutorial-authoring-support



Tutorial Structure	Infilling	Continuation	Elaboration	Story-Seeding	FFST	Custom Prompting	Meta-Prompting
Title	✓			✓	✓	✓	✓
Hero shot/Thumbnail						✓	✓
Tutorial Header						✓	✓
Introduction	✓	✓	✓	✓	✓	✓	✓
Safety	✓	✓			✓	✓	✓
Table of Contents							
Step-by-Step structure	✓	✓	✓	✓	✓	✓	✓
Images/Videos						✓	✓
Links to External Tutorials						?	
Final Section	✓	✓	✓	✓	✓	✓	✓

Figure 3.1: Table mapping the UX patterns of controllable text generation to High-level-categories of a DIY tutorial

Dedicated operations for introduction, steps, and final section

are applicable. These are the elements on which the user will spend most of their time revising and writing. Thus, they should be able to manipulate and generate text in these elements as they see fit. However, we see a benefit in providing specific dedicated operations for each element that fit their context. The introduction element could have an operation that could generate an introduction from the rest of the DIY tutorial. A Custom prompt operation for a given step using only the text from the step as the context. The final section could have an operation to summarize the DIY tutorial and generate a conclusion.

Title - all patterns except Continuation and Elaboration

We realized that all UX patterns except the Continuation and Elaboration patterns apply to the title element. Since a user would not want to generate additional text on the title, the other UX patterns allow the user to modify and change the title as they see fit without expanding the size of the title.

Safety - all patterns except Elaboration and Story-seeding

We noticed that all patterns except Elaboration and Story-seeding are applicable for safety instructions. Since they are mostly just sentences and text, the user should be able to manipulate them as they see fit. We reasoned that since safety instructions are supposed to be concise and precise, they do not benefit from elaborating and detailing why there is a need to follow the safety instructions. It is implicit that not following safety instructions would result in harm or risk to oneself or others. They also do not benefit from Story-seeding or generating from scratch since they depend on the rest of the DIY tutorial and will change and be modified based on the tutorial. The user could ideally use a Custom prompt operation in a given step to generate safety instructions for that step. Thus, we see no point in providing it as a dedicated operation as it depends on the DIY tutorial, the step, and the author if they want to provide safety instructions for a specific step.

No need for a dedicated operation for safety

We realized that for the hero-shot/thumbnail image, tutorial header, and images/video, only two patterns are applicable for Custom prompt and Meta-prompt. Since the image-based elements do not have text, none of the other patterns apply. Similarly, the tutorial header contains the list of materials, tools & skills and the estimated time required to complete the tutorial; these do not benefit from the other UX patterns either. However, we realize that it might help the user to have dedicated operations when dealing with these elements. The images could have a dedicated operation where you can directly generate DIY instructions from the image. Again, we can have a dedicated operation to generate these lists in the tutorial header based on the DIY tutorial.

Only Custom and Meta-prompt for image-based elements and tutorial header

The table of contents can be auto-generated based on the chosen text format instead of relying on the LLM to generate the list since it would require the context of the entire DIY tutorial. Hence, none of the UX patterns are applicable here. Similarly, we are unsure if any patterns are appropriate for generating links to external tutorials since LLMs tend to generate deadlinks to resources or point to sources that do not exist on the internet that do not exist. We think they might create links to tutorials that do not exist without a mechanism to ensure that the URL is valid and if it is pointing to a resource the LLM says it points to.

No patterns apply to table of contents and links to external tutorials

We also went ahead and mapped the UX patterns to different parts of the writing goals design space by Gero et al.

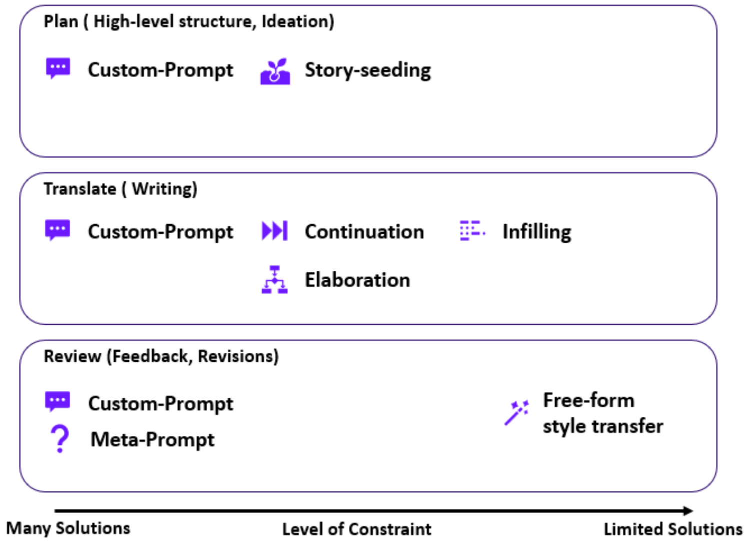


Figure 3.2: UX patterns of controllable text generation mapped to writer goals design space by Gero et al. [2022]

[2022]. We did this to understand where LLMs and these UX patterns may help aid the DIY tutorial author. As shown in the figure 3.2

Custom prompt and Story-seeding apply to the planning process

The planning process involves ideation and creating plans for writing, first a high-level plan like the structure of the tutorial, and later, lower-level plans like writing a paragraph in a specific manner. We think the UX patterns Custom prompt and Story-seeding are applicable. The Custom prompt is on the left because it has few constraints, and the Story-seeding is towards its right because the constraint here is the DIY tutorial the user wants to write. We also realized that creating a dedicated operation or step to generate a template or first draft of the DIY tutorial based on details supplied by the user might be helpful for our users. It might give users a structure for how a DIY tutorial should be structured and allow us control to have dedicated operations in specific parts of the DIY tutorial. The user can also use a custom prompt as they see fit to help plan the DIY tutorial.

Dedicated step or operation for outlining and planning

Custom prompt, Continuation, Elaboration, and Infilling apply to the translation process

The translation process involves converting ideas, thoughts, images, etc into words. The UX patterns applicable here are Custom prompt, Continuation, Elaboration, and Infilling. Custom prompts can be used to generate text arbitrarily based on user instructions and thus have very few constraints and many solutions. Thus, it sits on the left. Continuation and Elaboration also help with translation as they allow the user to generate the next phrase or add detail to a part of the text. They are constrained based on the existing DIY tutorial and thus sit in the middle as they have fewer solutions than Custom prompts. Infilling helps the user find suitable replacements and fill in the middle of piece of text, it is highly constrained due to fact that the solution must fit in with the text before and after the selection the user has made to find replacements. It sits on the right of all the other patterns. We don't see the need for a dedicated operation for this process, as the UX patterns can directly fulfill the goal of the translation process.

Custom prompt, Meta-prompt, and FFST apply to review process

The review process involves the writer getting feedback and revising the text according to their requirements. The UX patterns that apply are Custom prompt, Meta-prompt, and Free-form style transfer (FFST). Custom prompts and Meta-prompts allow the user to instruct or ask the LLM for suggestions for actions freely. Thus, they sit on the left as they have many solutions. FFST, on the other hand, involves transforming a piece of text stylistically based on how the user wants to modify the text. It is quite constrained and thus sits on the far right. We realize that the user might benefit from a dedicated operation for getting review and feedback. This allows them to quickly review the DIY tutorial they are writing and get feedback on it.

Dedicated operation for review/feedback

3.2 Requirements

The requirements for our prototype are based on the need to be able to write and edit a DIY tutorial, the goals of this thesis, and the design process seen in the previous section.

Full-fledged text editor with LLM-based operations

At the minimum, the prototype needs to have a rich text editor where authors can write and edit their DIY tutorials. It should be able to support images and multimedia

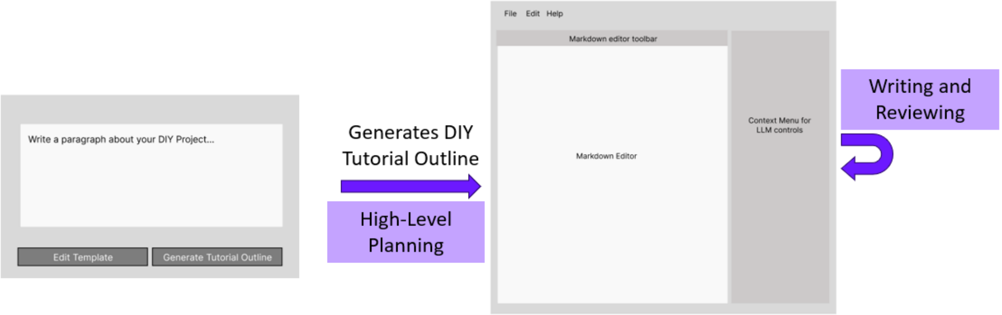

Write a paragraph about your DIY Project...
Edit Template
Generate Tutorial Outline
Generates DIY Tutorial Outline
High-Level Planning
File Edit Help
Markdown editor toolbar
Markdown Editor
Context Menu for LLM controls
Writing and Reviewing

Figure 3.3: Initial idea for the prototype shows the user describe the DIY project and generate an initial outline for a DIY tutorial

content in the future if required. Further, there must be a way for authors to apply LLM-powered operations to their DIY tutorial easily. They should be able to control whether to insert generated text. Otherwise, they might feel that they have no ownership of the DIY tutorial [Lehmann et al., 2022]

Additional features are outline,chat and review

Additional requirements from our design process include the following: The prototype also needs to have an outline step to generate a structure for the tutorial to allow site-specific operations, the ability for the users to chat with the LLM about their DIY tutorial and an operation that enables users to review the DIY tutorial and get feedback on it.

3.3 Initial Idea and paper prototype

Initial idea: outline generation by describing a DIY project

Our initial idea for the prototype involved a straightforward layout where, in the first step, it asks the user to describe their DIY project; if they want to, they can also edit the template of the DIY tutorial they want to generate. Then, they can click on the generate tutorial outline button, and the LLM will create a basic outline and structure of the tutorial. The outline step usually coincides with the high-level planning of the writing process. Then, the user can

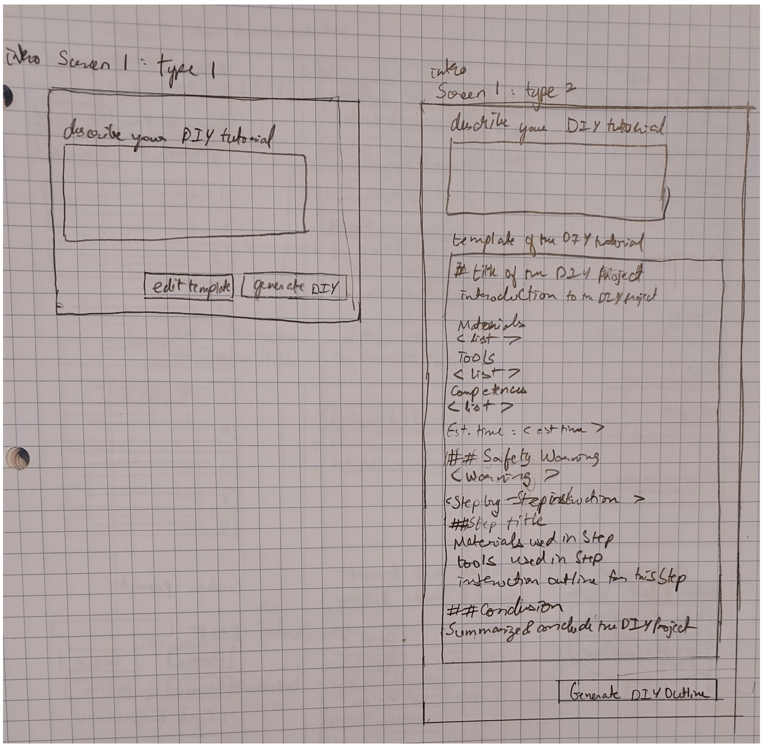


Figure 3.4: Paper prototype showing the outline step

write and edit the DIY tutorial in a text editor with LLM-powered controls on the side and a toolbar at the top for regular text editor options like undo, redo, etc in the text editor.

First screen - outline interface

We started by fleshing out the user flow on paper. The first screen looks very similar to the initial idea. The idea of the first screen is to help the user create an outline by describing their DIY project, which would then generate an outline. They can still edit the template of the generated DIY tutorial here if they want. Once they click the generate DIY, it goes to the editor screen.

Second screen - editor interface

This takes the user to the second screen, the editor screen, as shown in figure 3.5. The editor looks quite similar to Wordcraft[2] by Yuan et al. [2022] and how they designed their editor. The editor screen allows the user to edit and

[2] https://github.com/PAIR-code/wordcraft

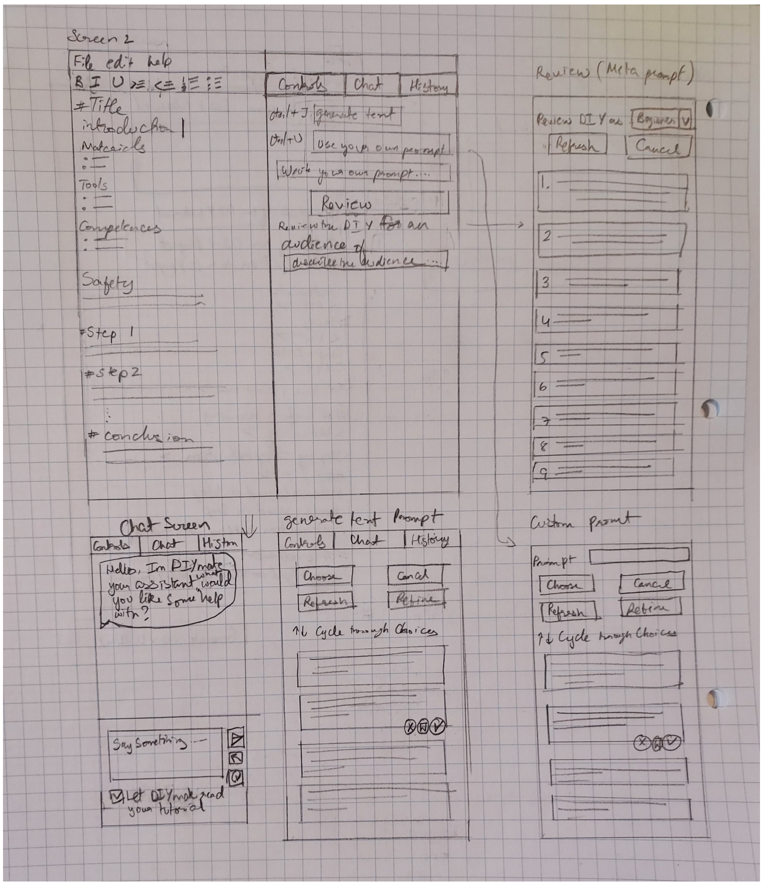


Figure 3.5: Paper prototype showing the editor step and the different updates to the sidebar when a choice or review operation is performed, a custom prompt operation started with user prompt and the chat tab

write their tutorial while simultaneously providing them with LLM-powered features on the right sidebar. They can use the various controls, and it would update the sidebar on the right while still showing the editor to show them multiple choices that the user can insert into the tutorial. To help the user review their tutorial, we also thought about adding a review operation where the user can get suggestions from the model to help them improve the tutorial or catch mistakes. In the sidebar, there is also the chat tab through which the user can chat with the LLM to discuss things about their tutorial or access an LLM while working on their DIY tutorial. We also ideated adding context-specific operations to show up on the sidebar.

3.4 Implementation

Modified the Wordcraft codebase for our implementation

Our prototype heavily builds on Wordcraft. We used the Wordcraft codebase as a base to build our prototype. Wordcraft was built using lit-html as the front-end and mobiledoc-kit as a text editor, it was built for using lambda, the LLM from google. Since Wordcraft was built for short story writing, it did not require a text editor with many features, like lists and headings; hence, the Wordcraft text editor does not really support them. Wordcraft has a really interesting architecture built on a service-based architecture where each feature is implemented as a service, for example, a service for saving documents, running operations, logging, etc. We also went with the architecture as modeled by Wordcraft.

We first decided to implement our prototype frontend using Typescript[3] and Lit[4]. We wanted to keep the base frameworks similar to Wordcraft to reuse parts of their codebase. Lit is a fast, frontend framework built around the Web components[5] standard as a light wrapper. Web components would allow this implementation to be used directly in future work as it is built for interoperability. It is also extremely fast and performant which was desired for our prototype.

We used the Tiptap[6] editor for our text editor framework after trying and testing various different text editors. It offered the right level of power and customization for the prototype. We implemented a markdown text editor where the user can type the DIY tutorial in a markdown format. This also allowed us to structure the tutorial into various sections and allow for creating site-specific operations.

For our backend, which would handle our queries and

[3] https://www.typescriptlang.org/
[4] https://lit.dev/
[5] https://www.webcomponents.org/introduction
[6] https://tiptap.dev/

store our images. we implemented our server in Flask. While heavily modifying it from the codebase of coauthor by Lee et al. [2022] for supporting queries and images.

We used chatGPT4.0-turbo for our LLM model since, at the time, it was the only model capable of supporting images and text together. All operations except the reviews and the chat used completions API. The reviews and chat instead used the assistant API, though at the time of implementation, the assistant API could not accept images. Every user starts with their own dedicated thread for the assistant API for chat and reviews. This thread keeps track of the user's context for the DIY tutorial they are writing. Since the chat and reviews utilize the same thread. Users can get a more customized review if they interact with the LLM using the chat feature.

3.4.1 Preliminary user study

Before starting the user study, we tested our prototype and user study protocol mentioned in the chapter 4 on 2 participants. Their feedback was invaluable to fix some issues in the prototype and also improve our study protocol. We will designate the test participants as TP01 and TP02.

TP01 suggested image-based operations and materials for crafting

TP01 had a lot of DIY experience under their belt. They had previously even written DIY tutorials and shared them online. They successfully completed the DIY project and wrote a tutorial about it. They made a cat stimulation toy shaped like a cat. They highlighted some formatting and punctuation errors. They also suggested that we add generating a DIY instruction based on the image. They recommended keeping paper clips, a ruler, and a protractor as materials for the DIY task. They pointed out that the error in the outline screen needs to catch attention with a red banner. Overall, the prototype worked well, and we got good feedback on improving our prototype before the study.

TP02 helped improve the visual hierarchy of the outline and the study protocol

TP02, on the other hand, had never done DIY before. They could not complete the DIY project they set out to do. The scope of the project exceeded their skill. They made a "Sand

Worm on a Desert". They were allowed to work on the tutorial while they were working on the DIY task. They highlighted that the outline screen was a bit hard to read. We then used this to improve the visual hierarchy and readability of the Outline step. Since they could not complete the project, they were distracted by simultaneously working on the tutorial and project. We realized that the participant would have to be given only one task at a time. They would be asked to do the DIY task and once they have completed the DIY project. Only then will they be asked to write the DIY tutorial.

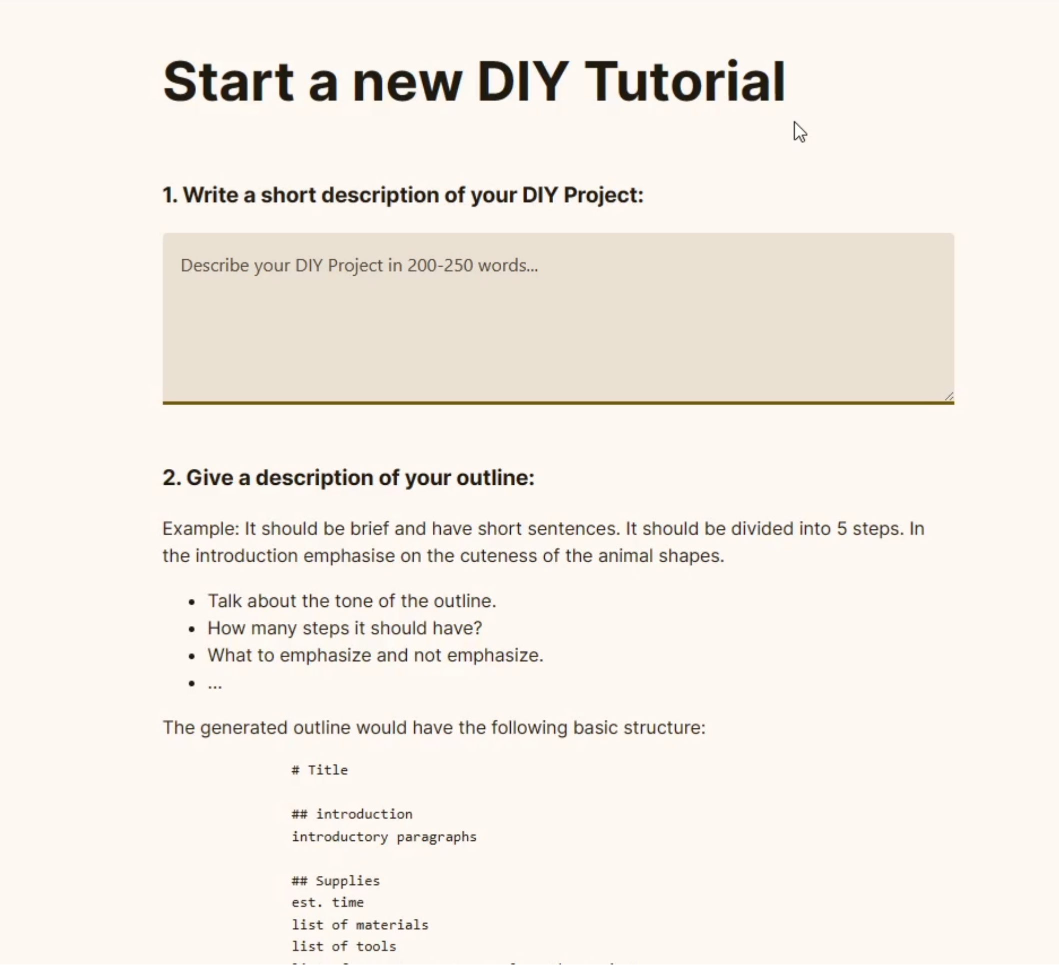
3.5 Outline step

The user starts a new DIY project at the outline step. Here, they are able to generate a quick outline for the DIY project that they completed. In the outline step, as shown in figure 3.6. The appendix A.1 shows other parts of the outline step. It has 4 key parts:

1. The user fills up the DIY project description.
2. The user fills up the DIY outline description
3. The user generates an outline
4. The user receives an outline, at this point they can go back edit the project description or outline description if they are not satisfied with the outline generated and regenerate an outline. The user can also edit the generated outline to fix minor issues.
5. If they are satisfied with the outline the user can click on confirm outline to go to the editor.

3.5.1 Template of generated DIY tutorial

We used the following structure for our generated DIY tutorial to control where certain operations could be available. It is inspired by the template from Lahaye et al. [2023].



Start a new DIY Tutorial

1. Write a short description of your DIY Project:

Describe your DIY Project in 200-250 words...

2. Give a description of your outline:

Example: It should be brief and have short sentences. It should be divided into 5 steps. In the introduction emphasise on the cuteness of the animal shapes.

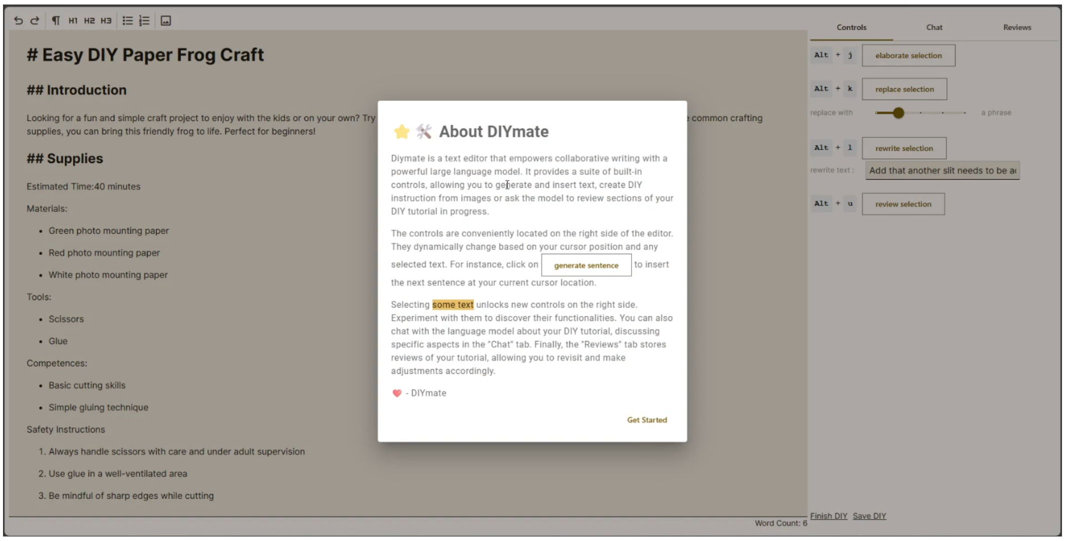
- Talk about the tone of the outline.
- How many steps it should have?
- What to emphasize and not emphasize.
- ...

The generated outline would have the following basic structure:

```
# Title

## introduction
introductory paragraphs

## Supplies
est. time
list of materials
list of tools
```

Figure 3.6: Outline step shown to the user, the user describes the DIY project and the outline. Then, they generate an outline

```
# Title of the DIY project
## Introduction
paragraph for introduction
## Supplies
list of materials
list of tools
list of skills
list of safety instructions

## Steps
\\ each step has the format below
### Step
```



Easy DIY Paper Frog Craft

Introduction

Looking for a fun and simple craft project to enjoy with the kids or on your own? Try common crafting supplies, you can bring this friendly frog to life. Perfect for beginners!

Supplies

Estimated Time:40 minutes

Materials:

- Green photo mounting paper
- Red photo mounting paper
- White photo mounting paper

Tools:

- Scissors
- Glue

Competences:

- Basic cutting skills
- Simple gluing technique

Safety Instructions

1. Always handle scissors with care and under adult supervision
2. Use glue in a well-ventilated area
3. Be mindful of sharp edges while cutting

Word Count: 6

⭐ 🛠 About DIYmate

Diymate is a text editor that empowers collaborative writing with a powerful large language model. It provides a suite of built-in controls, allowing you to generate and insert text, create DIY instruction from images or ask the model to review sections of your DIY tutorial in progress.

The controls are conveniently located on the right side of the editor. They dynamically change based on your cursor position and any selected text. For instance, click on generate sentence to insert the next sentence at your current cursor location.

Selecting some text unlocks new controls on the right side. Experiment with them to discover their functionalities. You can also chat with the language model about your DIY tutorial, discussing specific aspects in the "Chat" tab. Finally, the "Reviews" tab stores reviews of your tutorial, allowing you to revisit and make adjustments accordingly.

❤ - DIYmate

replace with: a phrase

rewrite text : Add that another slit needs to be a

Figure 3.7: Welcome banner shown to the user for onboarding into the editor

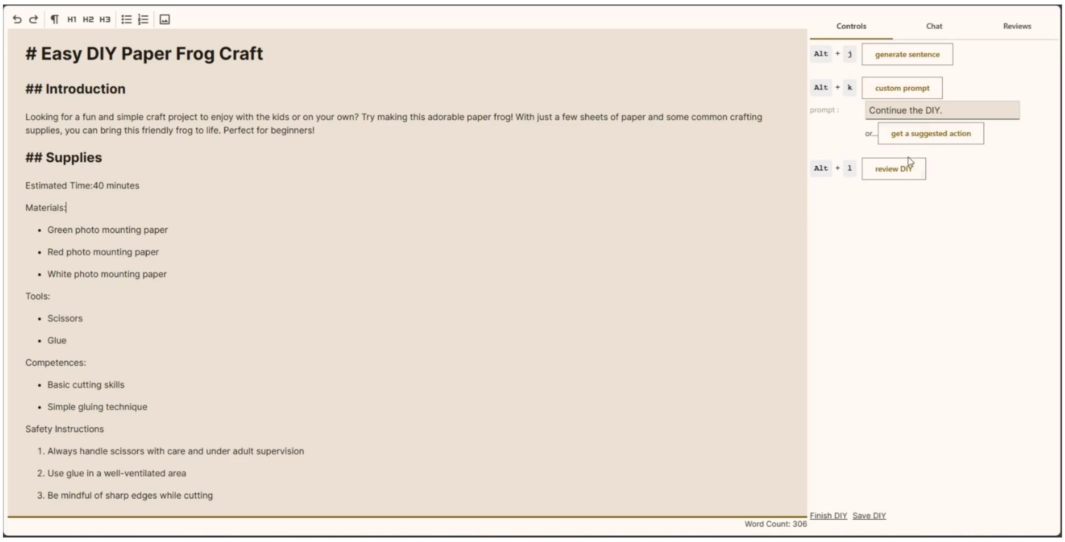
```
list of materials used in step
list of tools used in step
instructions for this step

## Conclusion
paragraph for conclusion
```

3.6 Editor Step

In the editor step, the first thing that the user sees is the welcome banner, which quickly introduces them to the DIYMate editor shown in figure 3.7; it explains the different features the editor has, like generating DIY instructions from images or reviewing the DIY tutorial. It also explains to the user that the operations shown in the sidebar change based on where the cursor of the user is. The appendix A.1 shows additional user interface images for the editor screen.

Editor step is the main screen of the prototype

Once the user clicks get started on the welcome banner, they can access and immediately interact with the text editor shown in figure 3.8, which takes up the left-hand portion of the screen. On the top is a menu bar for the text editor, allowing them to do some basic text editor opera-



Easy DIY Paper Frog Craft

Introduction

Looking for a fun and simple craft project to enjoy with the kids or on your own? Try making this adorable paper frog! With just a few sheets of paper and some common crafting supplies, you can bring this friendly frog to life. Perfect for beginners!

Supplies

Estimated Time:40 minutes

Materials:

- Green photo mounting paper
- Red photo mounting paper
- White photo mounting paper

Tools:

- Scissors
- Glue

Competences:

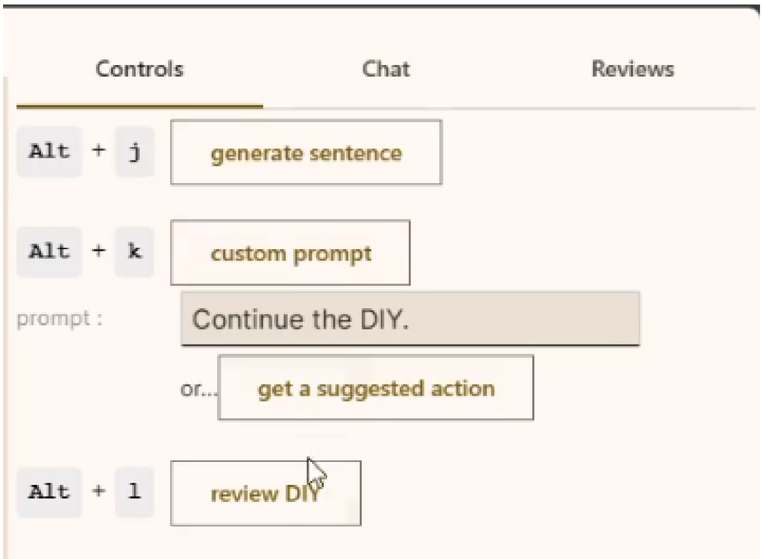
- Basic cutting skills
- Simple gluing technique

Safety Instructions

1. Always handle scissors with care and under adult supervision
2. Use glue in a well-ventilated area
3. Be mindful of sharp edges while cutting

Continue the DIY.

Figure 3.8: The figure shows the editor screen interface with a top bar for editor commands. A sidebar for LLM-powered controls and tabs for chat and review.



Continue the DIY.

Figure 3.9: editor sidebar with operations and the tabs

tions such as undo, redo, clear formatting, format to h1-3, add a bullet list, add a number list, and upload an image into a tutorial.

The figure 3.9 shows 3 tabs in the editor sidebar:

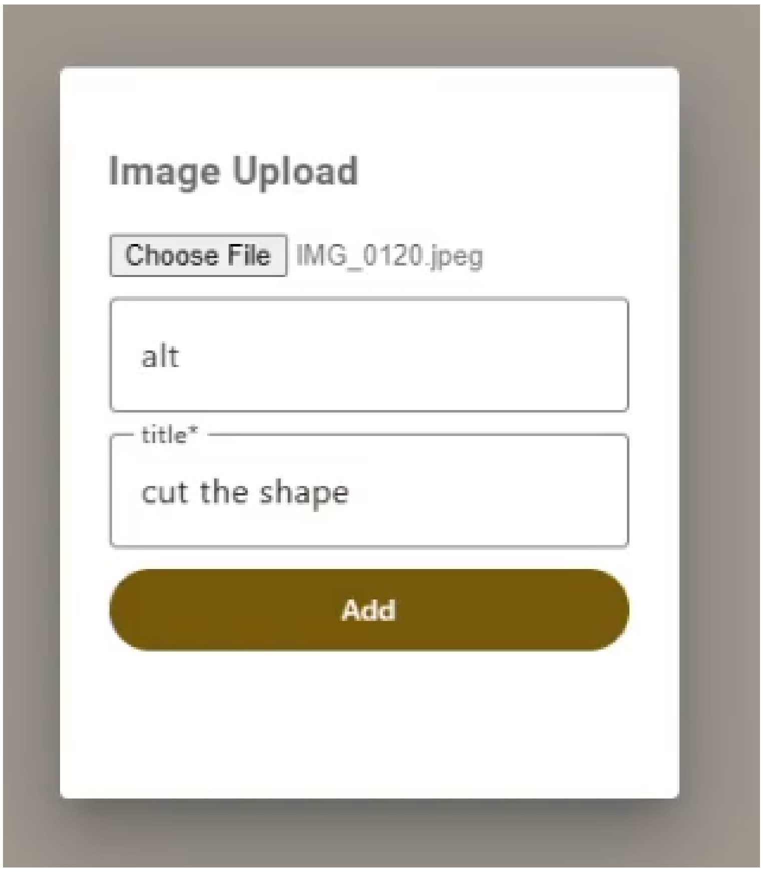


Image Upload

IMG_0120.jpeg

alt

title*: cut the shape

Figure 3.10: Image modal that shows up for image upload

1. Controls - which show the available operations at the current operation site. It also shows the currently active operation whether it be a choice operation or a review operation. Also, allowing the user to save and finish the DIY

2. Chat - A chat tab that allows the user to chat with the LLM

3. Review - this tab shows the different reviews saved by the user.

The figure 3.10 shows the image upload modal. The user has to choose a single image, enter the alt text for the image if they want to, add a title to the image. And, then click the add button in the bottom to add the image into the text editor.

3.7 Operations

LLM-powered operations implemented in the prototype

We implemented various operations to help our DIY authors write a DIY tutorial. The operations are distinct from the prompts used to query the model. Operations are what the user sees show up on the sidebar. They allow the system to start a prompt with different kinds of data and control where specific operations show up and what data should be given to the prompt to get the required output. The prompts for the various operations are listed in Appendix A.2. The operations are divided into two kinds: Choice operations, which allow the user to generate text and view different choices before selecting and inserting them in, and Review operations, which are a list of suggestions to help the user review their DIY tutorial.

Operation site	Cursor location/selection
Selection	text selection
Image node selection	an image node is selected
Empty document	when the document is empty
Empty section	when the current paragraph is empty
Start of section	when the cursor is at the start of a paragraph
End of section	when the cursor is at the end of a paragraph
Within sentence	when the cursor is within a sentence
Between sentences	when the cursor is between 2 sentences
DIY title	when the cursor is at the DIY title
DIY section title	when the cursor is at the DIY section title
DIY introduction	when the cursor is at the introduction
DIY conclusion	when the cursor is at the conclusion
DIY step title	when the cursor is at the step title
DIY Step	when the cursor is inside a step
none	no specific location

Table 3.1: An overview of operation sites where operations can be made available

We identified various operation sites shown in table 3.1.

They are used to make an operation available in a specific context based on the location and contents of the selection of the cursor.

3.7.1 Choice operations

Figure 3.11 shows the choice operation, where the different options are shown on the sidebar. The user can select them to insert them into the text editor. They can also use the arrow keys to see how the text would appear in the DIY tutorial on the left. During an operation, the user is not allowed to modify/edit text in the text editor. We will now list and briefly describe the various choice operations available in the prototype.

Outline: The outline operation allows the user to generate a DIY tutorial outline based on the description of the DIY tutorial and the outline description. It currently outputs only a single choice, since the length of the generated text is quite long. The outline operation uses the outline prompt listed in A.2.10. In the editor screen it is available only when the document is empty.

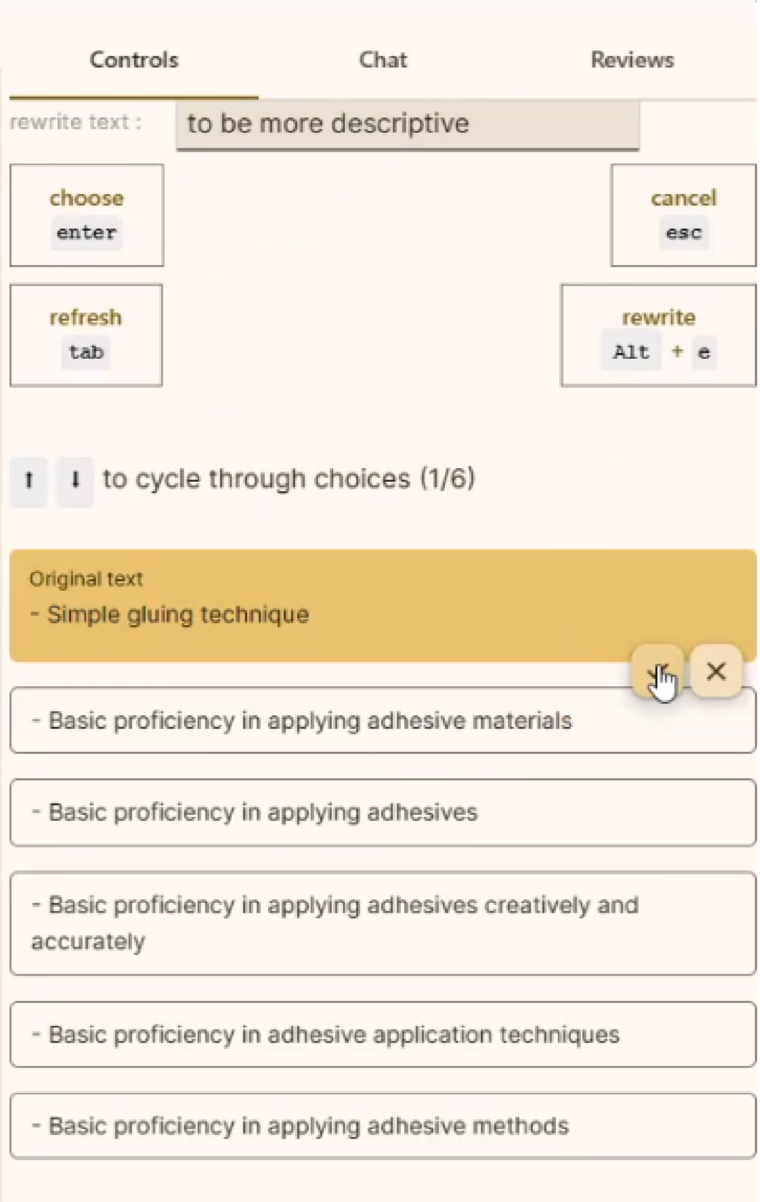
Continue: The continue operation allows the user to continue the DIY tutorial from where their cursor currently is. It generates 5 choices. It uses the text before the cursor to generate the text after.

Custom image prompt: Allows the user to run a custom prompt on an image selection. It generates 5 choices.

Elaboration: Generates text to elaborate on a piece of text the user selects. it generates 5 choices.

Freeform: Allows the user to run a custom prompt where they get to input a custom instruction for the model and it generates 5 choices. It sends the entire DIY tutorial as context to generate the text.

Freeform Step: Allows the user to run a custom prompt where they get to input a custom instruction for the model



rewrite text : to be more descriptive

↑ ↓ to cycle through choices (1/6)

Original text
- Simple gluing technique

- Basic proficiency in applying adhesive materials

- Basic proficiency in applying adhesives

- Basic proficiency in applying adhesives creatively and accurately

- Basic proficiency in adhesive application techniques

- Basic proficiency in applying adhesive methods

Figure 3.11: A choice operation shown in the sidebar of the editor screen

and it generates 5 choices. It sends the current step the user is on as context to generate the text.

Generate Introduction: Generates a new introduction when the user is in the introduction. It generates 5 choices. It sends a DIY tutorial but without the introduction to generate a new one.

Generate Conclusion: Generates a new conclusion when the user is in the conclusion. It generates 5 choices. It sends a DIY tutorial without the conclusion to generate a new one.

Image DIY instruction: Generates a DIY instruction based on the image in selection. It generates 5 text choices for the DIY instruction to be inserted into the editor.

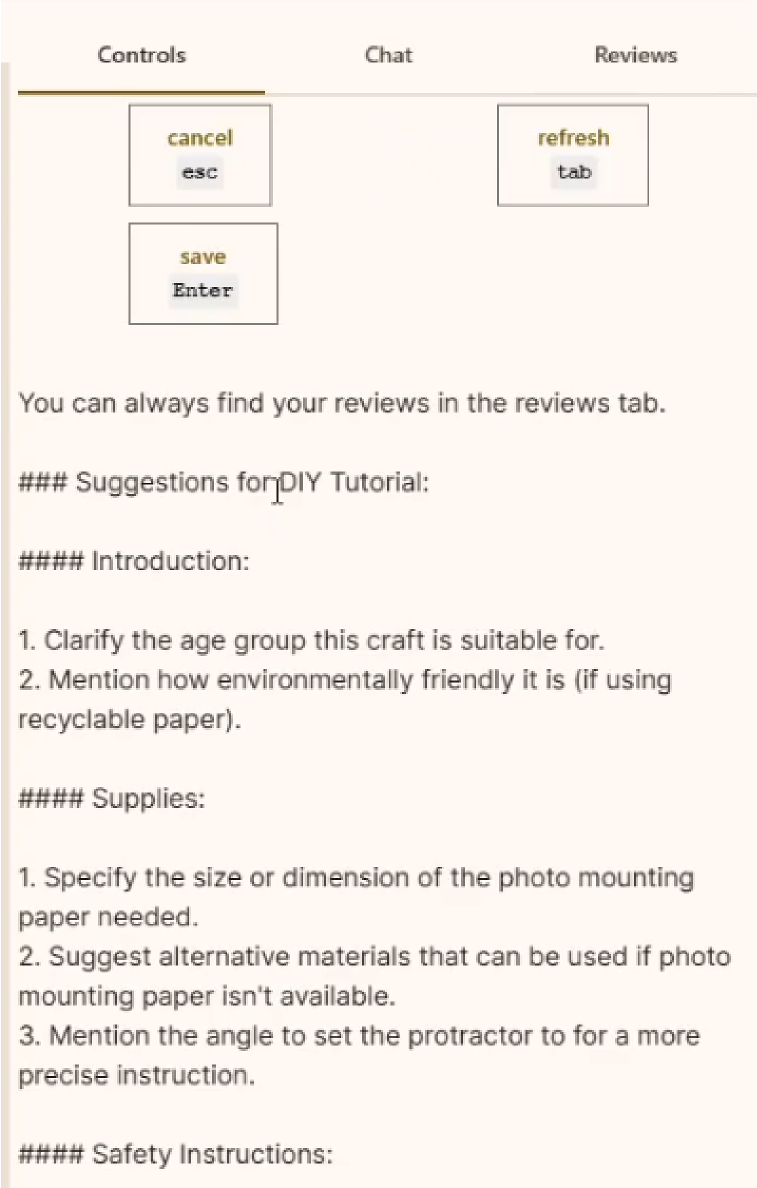
Meta prompt: Suggests actions for the user to use in a freeform or freeform step prompt. It is a helper operation that cannot exist independently of other operations.

Next sentence: Generates the next sentence if the user is at the current sentence or the end of a current sentence. It generates 5 choices.

Replace: Suggests replacements for the currently selected piece of text. It sends the text before the cursor and the text after the cursor to generate a replacement. It generates 5 choices.

Rewrite Sentence: Allows the user to rewrite the current sentence based on custom instructions. It is available when the user is within the sentence. It generates 5 choices. It sends the text before the sentence, the text after the sentence, the current sentence, and the user's directions on how to rewrite.

Rewrite Selection: Allows the user to rewrite the current selection based on custom instructions. It is available when the user has selected some text. It generates 5 choices. It sends the text before the sentence, the text after the sentence, the current selection, and the user's direction on how to rewrite.



You can always find your reviews in the reviews tab.

Suggestions for DIY Tutorial:

Introduction:

1. Clarify the age group this craft is suitable for.
2. Mention how environmentally friendly it is (if using recyclable paper).

Supplies:

1. Specify the size or dimension of the photo mounting paper needed.
2. Suggest alternative materials that can be used if photo mounting paper isn't available.
3. Mention the angle to set the protractor to for a more precise instruction.

Safety Instructions:

Figure 3.12: A review operation shown in the sidebar of the editor screen

Rewrite Choice: Allows the user to rewrite the current choice option based on custom instructions. It is available as a helper operation in any choice operation. It generates 5 choices. It only sends the current choice and direction to rewrite.

3.7.2 Review Operations

The figure 3.12 shows the review operation when the user starts a review DIY operation. It shows a list of suggestions in the text on the right sidebar. The user can save the review to then look at it in the reviews tab.

Review DIY: It generates a list of suggestions as feedback to review for the DIY tutorial. available anywhere the cursor is. It creates suggestions for each section and step of the DIY tutorial. The suggestions are listed in the text for the user to read and review.

Review selection: It generates a list of suggestions as feedback to review for the current selection. It is available when the user has selected some text. The suggestions are listed in the text for the user to read and review.

3.8 Document storage

Local storage of DIY tutorial

We also store the DIY tutorial created by the user during a session in the local storage. It is accessible via an admin interface where you just need to add a query parameter to the home page to access the different DIY tutorials created by the users. The documents are stored in JSON schema format from Tiptap but can easily be converted to HTML, markdown, or plain text if required.

Autosave, undo and redo

The DIY tutorial is automatically saved whenever a change is detected in the contents of the editor. We have also implemented undo and redo as a feature to allow the user to undo or redo any changes. This can be started by using the shortcuts `CTRL+Z` or `CTRL+Y` or by clicking on the undo/redo button in the top menu bar in the editor.

Chat and review stored with the document

The saved document contains the reviews and chat messages sent and received by the user, as these are session-specific data specific to a given DIY tutorial. We store them alongside the rest of the document to be retrieved when reopening it to view or edit.

3.9 Logging

Logs store usage data

We implemented a robust logging mechanism in the prototype to allow us to track which operations are used and with what instructions and data were used to start them. We also store the number of times a specific operation is used. Thus, giving us good usage data for our prototype.

Logs are stored locally

Currently, the logs are stored in local storage, along with all the other data stored for a session for the user. We have also implemented a schema for the logs so it will be easy to transfer it to a database and store it in the backend.

Data can be easily exported

At the end of the session, we can export the user data into a JSON file. It contains the demographic data of the user, the number of times different operations and actions were performed, chat history

Chapter 4

User Study and Results

In this chapter, we will discuss our user study design and results from evaluating our prototype. First, we describe the study design and the methodology of the study. Later, we describe the results of the user study.

4.1 Methodology

4.1.1 Procedure

DIY task, tutorial writing task, and a semi-structured interview

The study was split into three parts: a DIY task, a tutorial writing task, and a semi-structured interview. The study was held in person and lasted approximately 120 minutes. First, the participants were asked to fill out the informed consent and demographics form. Next, they were given a sheet with instructions about the DIY task. Participants were asked to complete a craft DIY ″Animal″ with the provided materials. They were asked to write down their steps on paper and take pictures of their project as they crafted it. Once they finished the DIY project. We helped them transfer the images to the laptop with the prototype for the DIY tutorial. Afterward, participants were asked to write down a DIY tutorial about the DIY project they just completed using the prototype. Participants were allowed to take 120 minutes for these tasks if time permitted. Lastly, we asked the participants about their experience using the prototype in a semi-structured interview.

4.1.2 DIY task

DIY craft task chosen for simplicity, open-ended nature, and creativity

Our participants were given a task to craft a DIY "Animal" using the materials listed in the appendix B.2. The task is open-ended so that participants can utilize their creativity and decide the kind of DIY project they want to do. We also wanted to test our prototype to see how well it handled different kinds of DIY projects. We also chose a craft task so that our participants would not need specific technical skills to complete a project. And they would not get stuck on a project or task they could not complete.

Project to be chosen based on participant skills and to fit within study time

Participants were to ideate and decide what they wanted to do for the DIY project as long as they fit the instructions. They could decide on any kind of craft project that fits their skills and experience. Also, they had to choose a DIY project that they could comfortably finish within the time constraints of the study or a maximum of 120 minutes. While leaving 30 minutes to 1 hour to write the DIY tutorial. We also suggested actions and techniques that could reduce the time to complete the DIY project.

Participants can search on the internet

We wanted our participants to write down their tutorials without referring to other external tutorials. However, We understood that in DIY, as observed by Kuznetsov and Paulos [2010], participants learn and take inspiration from another DIY project online. Thus, they could use the internet to search for inspiration, specific techniques, and other DIY tutorials to complete their DIY project.

Taking notes and documenting the DIY project

Participants were asked to write down the steps on a sheet of paper and take images of the DIY project as they performed each step. This was to help them in the DIY tutorial task by allowing them to have a written sequence of actions they performed in the DIY project.

4.1.3 Tutorial Writing task

The particpants were asked to write a DIY tutorial using the prototype we built, they could refer to the notes they made during the DIY project task and also the images and the artifact that they made for the DIY task.

Forbidden from using external sources while writing a tutorial

During the tutorial writing task, the participants could not refer to the external sources they used while building the DIY project. This was to make sure that the tutorial written was original and not just a copy of an existing tutorial while also keeping the focus on the usage of the prototype to assist in the task of writing the tutorial.

Completing the tutorial was not necessary

While participants were compelled to complete the DIY project, they could leave the tutorial writing task incomplete. This was done to make the study fit within the time constraints. Also, our main goal with the tutorial writing task was to evaluate how the participants used the different LLM-powered operations to write a DIY tutorial. Thus, the completion of the tutorial was not necessary.

Encouraged to use LLM-operations

Participants were encouraged to try different LLM-powered operations during the writing process. This was done since participants would be using the system for the first time, and they might default to using it as a normal text editor. This was to promote the exploration and usage of different LLM operations as the participants saw fit.

4.1.4 Participant Demographics

We selected participants who had experience with DIY projects and were familiar with online DIY platforms. Additionally, we included participants who had no prior experience with DIY in order to gain insight into how well the prototype works for non-makers. Our participants were recruited from the university campus.

4.1.5 Setup

We performed a within-groups qualitative user study, where the participants completed a craft DIY project and used the prototype to write down a DIY tutorial. Each participant was given the same instructions for the DIY project. The instructions for the DIY task are linked in the appendix B.1 The study protocol is linked in the appendix B.2

We would collect data about the usage of the prototype and then, in the end, conduct a semi-structured interview to get qualitative data regarding the experience of the participant in using the prototype to write a DIY tutorial. The entire study and interview was performed in English. The recording from the Interview was transcribed for qualitative data analysis.

4.1.6 Research Question

The goal of our study was to answer the following research question:

RQ1: How can we use LLMs to assist DIY Tutorial Authors in authoring/writing high-quality DIY tutorials?

Since the question above is broad, we further split our question down into three focused questions based on the cognitive process model of writing by Flower and Hayes [1981].

RQ1.1: Can we use LLMs to improve planning the DIY tutorial?

RQ1.2: Can we use LLMs to improve translating the DIY tutorial?

RQ1.3: Can we use LLMs to improve reviewing the DIY tutorial?

4.1.7 Data Collected

We collected various pieces of data to evaluate LLMs as a writing support tool based on evaluation data guidelines by Gero et al. [2022]. While our study is qualitative, they recommend also collecting quantitative data about their usage of the writing support tool:

1. Demographics data - occupation, experience in DIY, how frequently they perform a DIY project, and number of DIY tutorials written & published.
2. Screen/Video data about their usage of the prototype.
3. log data based on system usage - number of operations used, type, data used to start an operation, number of inserted words, final word count of the tutorial.
4. Chat history - If they chatted with the LLM during the study. We would store their messages and responses.
5. Review history - the review suggestions suggested by the LLM to the participants.
6. Audio data from the semi-structured Interview.
7. Time spent in different stages of the study - working on the DIY project, On the Outline step, and writing the tutorial in the editor.
8. The confirmed outline from the outline step and the DIY tutorial finished at the end.

4.2 Results

4.2.1 Demographics

The table 4.1 displays an overview of the participant's demographics data. 12 out of 14 participants had some experience with DIY, and two participants had no experience with DIY before the study. However, only three of them had written DIY tutorials before. A majority of our participants were computer science students from the university, except two. In terms of frequency of starting or completing a DIY project, one participant started a DIY project every week, five participants started a DIY project once a month, four participants started a DIY project every couple of months, one participant started a DIY project once a year, and three participants almost never started a DIY project.

ID	Study Field	DIY experience	DIY Frequency	DIY tut frequency
1	Computer Science	8	Once a Month	Atleast 25
2	Computer Science	4	Once a Year	None
3	Computer Science	0	Almost Never	None
4	Computer Science	3	Every Couple of Months	None
5	Master Data Science	1	Once a Month	None
6	Media Informatics	18	Once a week	None
7	Computer Science	0	Almost Never	None
8	Mechanical Engineering	10	Every Couple of Months	Atleast five
9	Computer Science	6	Almost Never	None
10	computer science	10	Every Couple of Months	Atleast one
11	Computer Science	3	Every Couple of Months	None
12	Bsc. Mathematics	2	Once a Month	None
13	Computer Science	14	Once a Month	None
14	Computer Science	0.5	Once a Month	None

Table 4.1: The table showing an overview of participant demographic data

4.2.2 DIY Projects and tutorials

Almost all participants completed the DIY project and tutorial

The tables 4.2 and 4.3 display an overview of the different projects completed by the participants, what they referenced during the DIY task phase, and their tutorial completion status. All participants were able to complete the DIY project that they set out to do. While most participants, 12 out of 14, were able to complete writing their DIY tutorial to their satisfaction. One participant was unable to finish writing their tutorial. While having completed writing the tutorial, one participant admitted to cutting steps in their tutorial to fit within the time constraints of the study.

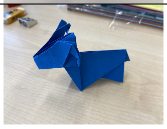
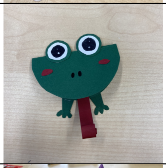
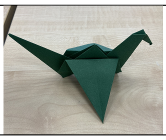
ID	DIY Project	Reference	Tutorial Completion Status	Image
1	DIY Cartoon-Style Layered Paper Frog	Image inspiration	Complete	
2	Fold an Origami Goat in 23 Easy Steps	Video tutorial	Complete	
3	DIY 3D Paper Bunny	Image inspiration	Complete	
4	DIY Paper Goat Project	Image inspiration	Complete	
5	Easy DIY Paper Frog Craft	Image inspiration	Complete	
6	Peacock Origami Tutorial	No reference, Video tutorial to check step	Almost Complete(cut steps for time)	
7	How to Fold an Origami Crane	No reference	Complete	

Table 4.2: An overview of DIY projects performed by participants P01-P07

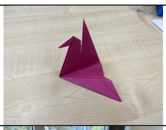
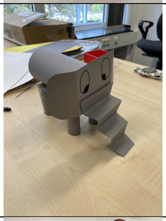
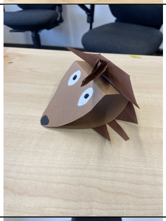
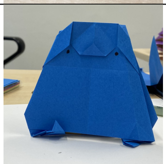
ID	DIY Project	Reference	Tutorial Completion Status	Image
8	DIY Origami Swan Tutorial	No reference	Incomplete	
9	DIY Paper Elephant	Image inspiration	Complete	
10	DIY Adorable Paper Sloth	Written tutorial	Complete	
11	DIY Paper Box Owl	Image inspiration	Complete	
12	Punk Hedgehog	Image inspiration	Complete	
13	DIY Boxy Paper Lion	Image inspiration	Complete	
14	DIY Blue Paper-folded Origami Penguin	Video tutorial	Complete	

Table 4.3: An overview of DIY projects performed by participants P08-P14

Overall, the participants all had varied projects with different kinds of skills involved in crafting: five origami projects, two flat craft projects, and seven 3-D papercraft projects. To complete their DIY projects, participants could search the internet. Participants searched for an image to get some inspiration for their DIY project from, a video tutorial or a written tutorial.

5 origami projects, 2 flat craft projects, and 7 papercraft projects

Of the five participants who did an origami project. two chose to watch a video tutorial; they said it aided them in folding. The remaining three participants completed their projects without almost any reference. Only 1 out of those 3 participants watched a video tutorial to check a step in the folding process after completing most of their DIY tutorial without any reference.

Video tutorial for reference for origami DIY project

The nine remaining participants completed their papercraft projects by searching for images for reference and inspiration. 8 out of 9 participants made the DIY projects independently from an original idea. One remaining participant chose to use an online written DIY tutorial because they said they were not good at drawing, sketching, and crafting.

Images for inspiration as reference for remaining DIY projects

Participant	word count	gen. words inserted
1	477	425
2	670	535
3	878	707
4	728	644
5	458	269
6	539	525
7	405	229
8	309	242
9	384	19
10	349	28
11	532	333
12	411	210
13	766	555
14	534	567

Table 4.4: The table displaying the final word count of a DIY tutorial and the number of generated words inserted

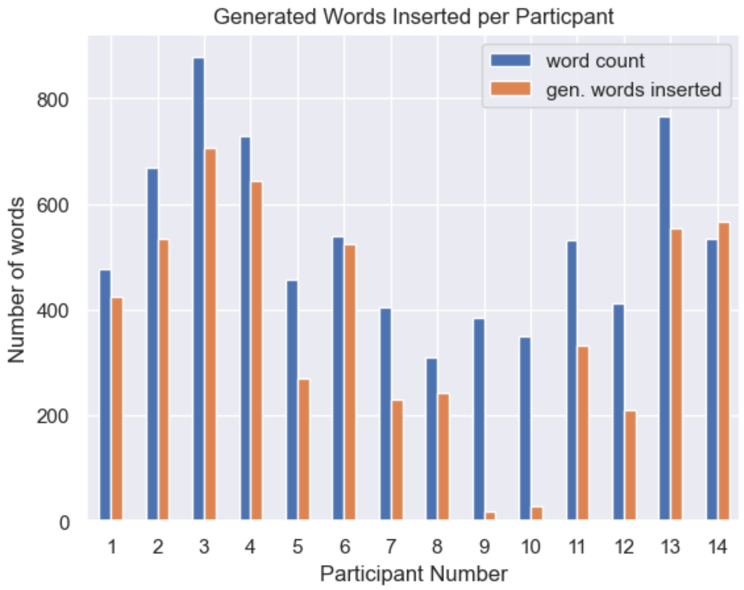


Figure 4.1: The figure shows the final word count in the tutorial document mapped against the words generated and inserted into the document by the particpant

The table 4.4 shows the number of words in the final tutorial and the number of words inserted into the DIY tutorial in the editor screen. It should be clarified that most of the tutorial was generated as an outline on the outline page. We consider them part of the original document, and the prototype cannot distinguish which characters were entered/modified by a person.

The graph 4.1 shows the amount of generated words inserted against the final word count of the document. We see that the final word counts and generated counts for certain participants are very close to each other. The closer the inserted word count is to the final, the higher the likelihood that the participants have used an operation to insert words into the DIY tutorial. They will likely have edited the inserted words to make it fit their DIY tutorial.

For participants P09 and P10, the outline generated in the first step of the prototype matched their DIY project due to writing a precise outline describing their steps in great de-

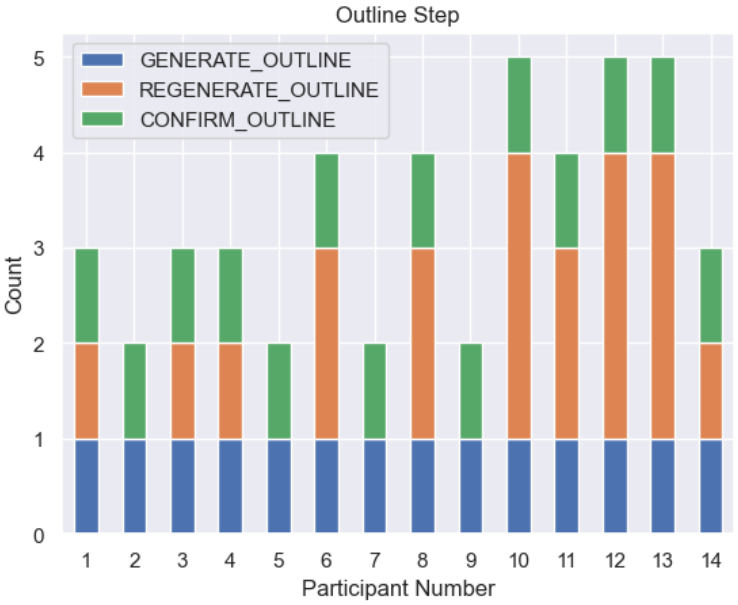


Figure 4.2: The figure shows the different operations carried out by the participant in the outline step. The first generation of the outline, the number of times the outline was regenerated, and the final confirmation of the outline

tail. They also manually edited the rest of the DIY tutorial in the editor. They only had to add images and fix specific details of their DIY tutorial.

4.2.3 Outline Operations

The graph 4.2 displays the different operations performed in the Outline step of the prototype, where the users generated the initial outline for the DIY tutorial of their DIY project.

Nine participants were initially confused about the outline step. After some help explaining what to do in the outline step, they were able to create an outline to their satisfaction. Notably, 4 out of 14 participants got an outline that matched what they wanted pretty well without any iterations. For P02 and P07, they made a standard origami DIY.

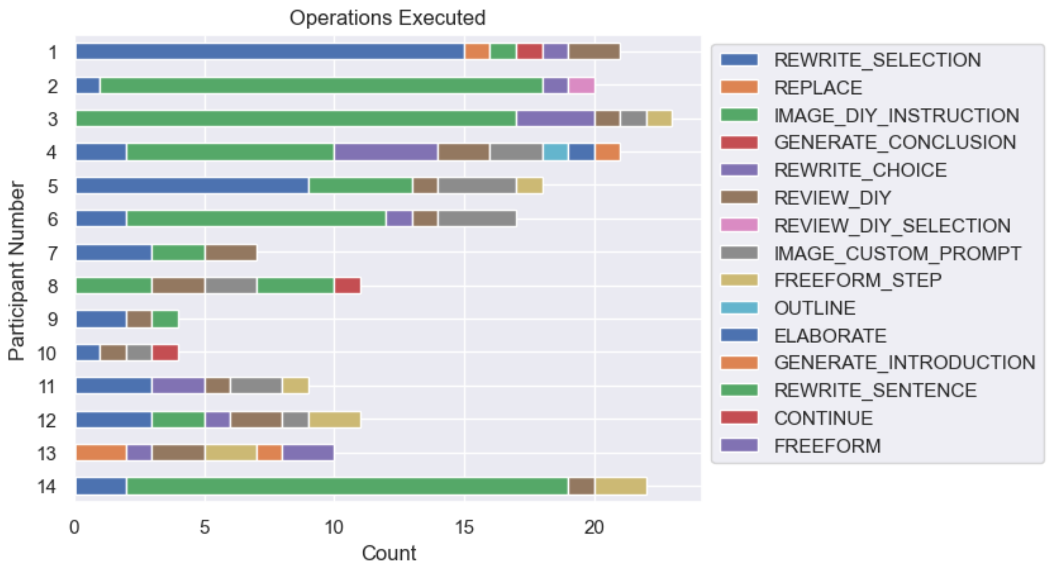


Figure 4.3: The figure shows the different operations executed by each participant.

The LLM provided broke down the project into a few high-level steps and could output the details pretty well, as the folding steps for those origami forms are standard. For P05 and P09, They gave an exact description/sequence of their tutorial, and hence, they got an outline that matched their DIY project.

The remaining 10 participants had to generate the outline at least once to understand how to create an outline they could be satisfied with. 4 out of these 10 participants, namely - P01, P03, P04, and P14, only had to regenerate the outline once they understood what the generated outline looked like and contained. The other 6 participants worked on the outline step in an iterative manner to get an outline they were satisfied with. 9 participants liked the outline generation process. Some lamented the non-intuitiveness of the outline generation and would have wanted to progressively refine their outline instead of regenerating one from scratch.

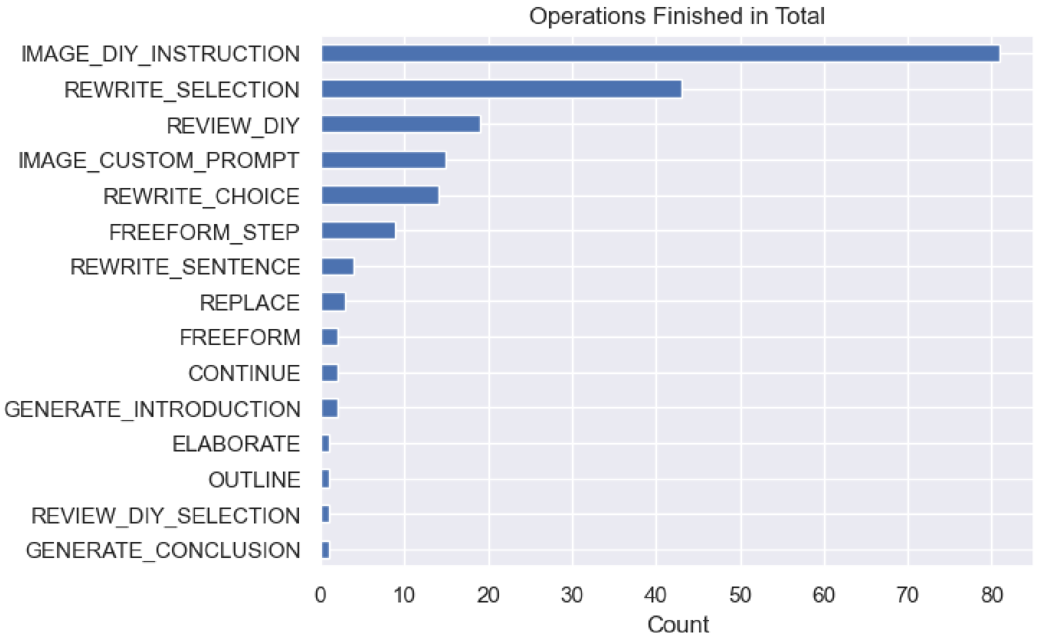


Figure 4.4: The figure shows the total number of operations sorted in decreasing order performed by the participants in the study.

4.2.4 Editor Operations

Figure 4.3 shows which operations were executed by which participants in the editor window. The operations shown here include operations that the participants canceled because the choices presented to them did not conform to them as they wanted them to be. Participants were generally encouraged to try different operations.

Figure 4.4 shows the total number of operations finished during the user study across 14 participants sorted in a descending manner. The most frequent operation used by the participants was the image DIY instruction, which generated DIY instruction based on the image. Participants found it to be convenient and reported that they found describing the photos to be the hardest part. However, they often found the length of text generated to be an issue. They would either rewrite it manually after removing the irrelevant bits or use the rewrite operation.

Instructions used in rewrite operation
to be more descriptive ⧫ ″ ⧫ add adult supervision ⧫ Make it fit to the text before with the frog drawing ⧫ add more flavor text ⧫ The slit is for the head instead of the legs ⧫ Add that another slit needs to be added for the tail ⧫ to add information that the paper should be still fold in a half ⧫ so it sounds more clear ⧫ so it sounds more clear and encouraging that you finished ⧫ to be more straightforwars⧫ to be shorter⧫ delete materials and tools used in steps⧫ delete materials and tools used⧫ but keep everything else⧫ rename steps, start with step 4⧫ to be concise⧫ replace feet with legs⧫ Clarify what a ″half DIN A4″ paper sheet means for an international audience who might use different paper size standards.⧫ mention that the tape was folded in a way such that it sticks from both sides to secure the body on the plaform⧫ to reference the introduction⧫ to be longer and describe the skill level⧫ also encourage the user to share their owls with others⧫ change this, i didnt use small black cycles but drew it with a balck pen⧫ first i had to make the white cycles for the eyes, than draw the black pupillen⧫ reduce it to the relevant minimum⧫ Add that one can draw eyes with a felt-tip pen

Table 4.5: The table shows the prompts used by users in the rewrite operation

The second most prominent operation was the rewrite selection, which most participants found extremely useful in enhancing existing written text. Nine participants expressed that they liked using the operation. Also, participants found it convenient that they could direct the text to be rewritten themselves. Participants used it in various ways, as shown in 4.5, from targeted rewrites to add specific details, to enhance the writing, to make the text more concise. The participants often used the rewrite choice operation to shorten the text generated before inserting it into the DIY tutorial.

After completing their DIY tutorial, participants were encouraged to use the review DIY operation. Participants expressed that many of the model's review suggestions were not useful. Some participants did find one or two useful suggestions. However, the operations executed graph does not record whether a suggestion was utilized. Some participants applied a suggestion from a review using a rewrite prompt or a freeform prompt as shown in tables 4.5 and 4.6. Some participants noted that it did not catch obvious mis-

Instructions used in Free-form & Free-form step operations
Add the two body parts together with glue♦ write a step that you use the glue stick as a shape to create an outline of the eye♦ Explain that when creating the symmetrical wings, I golded the paper, drew a wing on one side, cut it out, and had two identical wings.♦ write something about selecting a good origame paper e.g size♦ add something regarding how to fold it and that you should leave some space in the buttom unfolded where you will later gloue it to the the body♦ Add that you have to connect leg with body♦ ”Add that you have to connect leg with body. just add the last step. Don’t repeat the steps before”♦ ”Do this: General Suggestions: - Include a brief glossary of terms for beginners who might not be familiar with crafting terms. - Add tips on what to do if steps don’t go as planned, e.g., if glue isn’t holding well. - Provide easier alternatives for steps that require more dexterity, offering simpler options for younger crafters. - Suggest ways to make the activity more interactive, like involving personal customization ideas right after certain steps.”♦ Create next step about bending the diagonal edges of the paper in like for a paper plane♦ Draw eyes with felt-tip pen

Table 4.6: The table shows the prompts used by users in the Free-form and Free-form step operations

takes. Participant P04 noted that it did not catch the issue in the numbering of the steps.

The table 4.7 shows the various custom instructions used by the participants on their images. They used it to enhance or describe specific details of an image or the describe the image itself.

The graph 4.5 shows the percentage of operations that were executed, restarted, or canceled as a whole. 89.01% of all the image DIY instruction operations were successfully included in the document. 9.89% of all image DIY instruction operations were canceled. 1.09% of all the image DIY instruction operations were restarted. The rewrite selection operation was successfully incorporated into the document around 75.43%, restarted at 7.01%, and canceled 17.54% of the time.

The continue, generate next sentence, outline, and review selection operations were only executed once in the editor and were canceled. The generate conclusion operation was

Instructions used in custom image prompt operation
Cut along the lines made beforehand to create the two halves.⧫ Fold the cutout body 90 degrees⧫ Add a cut in the back of the body to fit the head in⧫ describe the image⧫ describe the folding step⧫ generate short DIY instruction⧫ tell me how great it looks⧫ describe the shapes on the image

Table 4.7: The table shows the prompts used by users in the custom image prompt operation

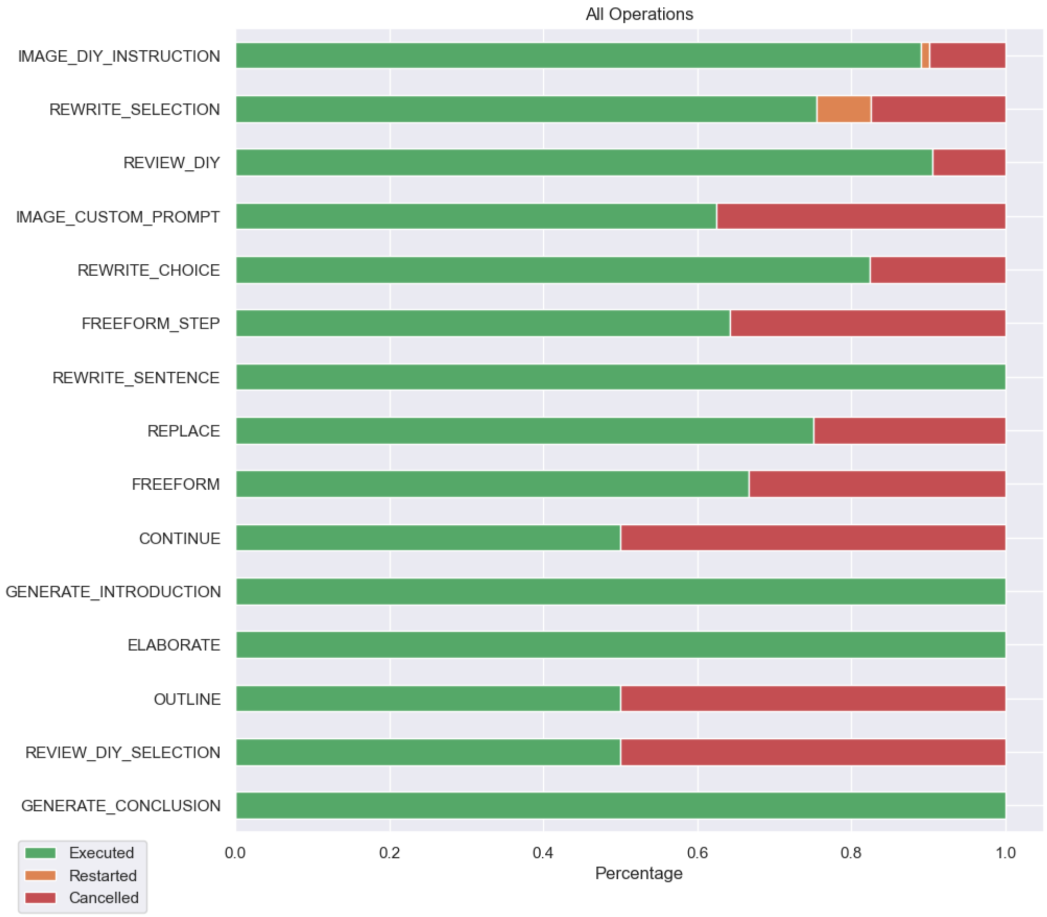


Figure 4.5: figure shows operations with the percentage of how many were executed, restarted, and canceled

only used once by participant P01. P04 and P13 used the generate introduction operation once each. The generated introduction and conclusion operations were successfully inserted into the document each time they were used. Also, the rewrite sentence was used twice by P08 and once by P09. They were used to make the sentence more concise and add detail to the instruction respectively.

4.3 Time spent on tasks

The graph 4.6 shows time spent by the participants on various tasks in the study: Making the DIY project, the Outline step, and the editor. The 14 participants spent an average time of 52.20 minutes on the DIY Project, 13.62 minutes on the outline step 34.16 minutes on the editor working on their tutorial. The maximum time spent on each task is 70.23 minutes for the DIY project. P13 had an extremely complex papercraft project that took most of their time crafting little parts for the paper lion. P14, on the other hand, had to make an origami penguin, but midway through the process, they realized the messed up step and video tutorial they were referring to was vague on a specific folding step. So they restarted their DIY Origami penguin from scratch. P07 spent the least amount of time, 27.81 minutes, in their DIY project.

4.3.1 Semi-structured Interviews

Interview regarding participant's experience using the prototype

After completing the DIY project and using the prototype to write a tutorial about it, the participants took part in a semi-structured interview. This interview aimed to gather information about the participants' experience with the writing process using the prototype. The participants were asked about how the prototype supported them with their writing process and their workflow in terms of planning, translating, and getting feedback on their DIY tutorial. The interviews were in English, and the audio data was recorded and transcribed.

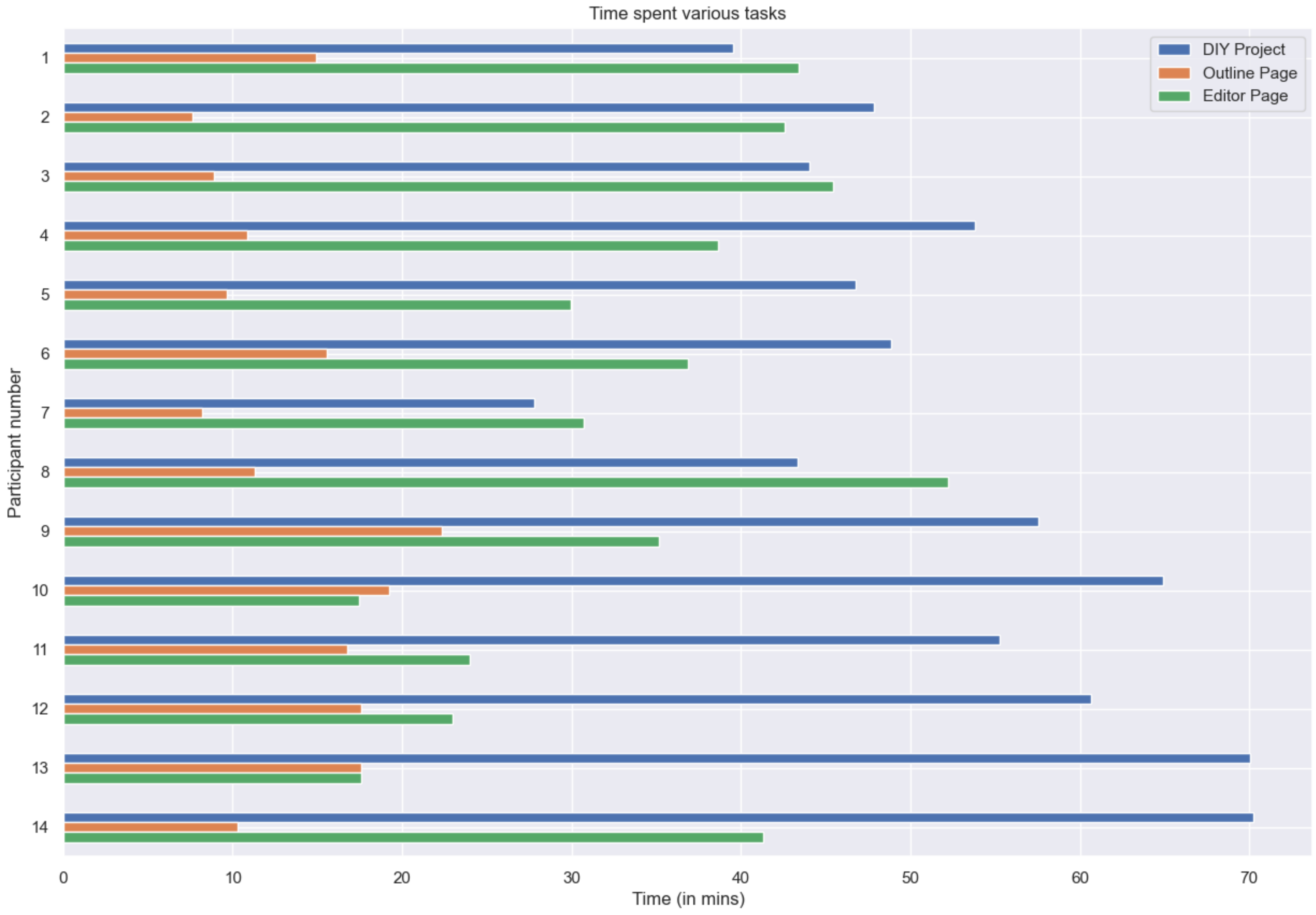


Figure 4.6: figure shows the time spent by the participants in completing the DIY project, time spent on the outline page and on the editor page.

We did qualitative coding of our interview data since we wanted to understand the writing experience of the participants from their own perspectives. According to Saldaña [2013], a code is a word or a short phrase that symbolically assigns a summative, salient, essence-capturing, and evocative attribute for a portion of language-based data.

Qualitative coding

From the different qualitative coding techniques, we used evaluation coding since our goal was to evaluate how participants used the prototype. If the participant expressed a positive sentiment regarding an aspect of the prototype, we assigned a positive code (+). We assigned a negative code (-) if the participant expressed a negative sentiment. We also assigned recommendation codes (REC) to capture user recommendations and derived recommendations from the negative codes.

Evaluation coding with positive (+), negative (-), and recommendation (REC) codes

In order to evaluate the data from our interview transcripts, a total of 561 segments were coded. Out of them, 303 were assigned positive codes, 200 were assigned negative codes, and 58 were assigned recommendation codes. The codes were then refined and grouped together into high-level codes to better organize and gain insights into the experience of participants. These high-level codes were then organized into categories. These categories were `PLAN` for the act of planning in the writing process, `TRANSLATE` for the act of translating thoughts into words,`REVIEW` for reviewing the writing and making changes based on the feedback, `GEN TEXT` for participant impressions of the generated output text and `TOOL` for participant impressions of the prototype itself. We will present the codes according to their evaluation.

Bottoms-up coding with high-level categories: PLAN, TRANSLATE, REVIEW, GEN TEXT, and TOOL

4.3.2 Positive codes

In this section, we list and describe the different high-level positive codes grouped by categories.

- **PLAN:**
  - `+PLAN:outline useful` (assigned to 21.45%

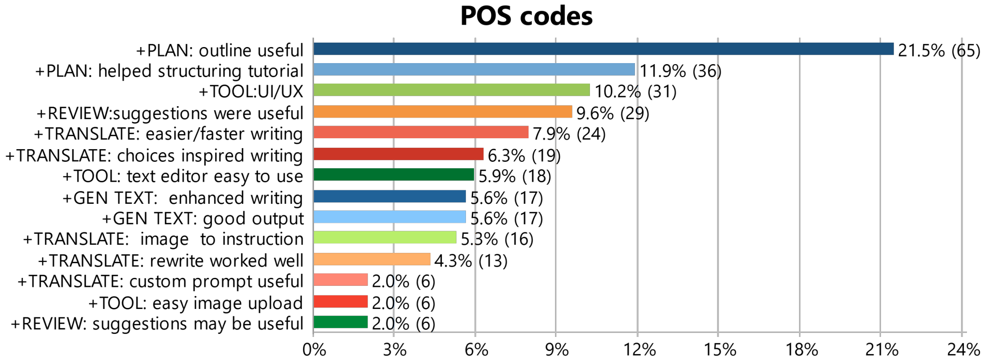


Figure 4.7: Positive Codes

of segments, 65 segments, by 100.00% of participants, 14 participants) assigned when the participants expressed that the outline was useful.

- `+PLAN:helped structuring tutorial` (assigned to 11.88% of segments, 36 segments, by 78.57% of participants, 11 participants) assigned when the participants expressed that the outline helped them structure the tutorial.

- **TOOL:**
  - `+TOOL:UI/UX` (assigned to 10.23% of segments, 31 segments, by 85.71% of participants, 12 participants) assigned when the participant expressed a positive sentiment regarding the UI/UX of the prototype.
  - `+TOOL:text editor easy to use` (assigned to 5.94% of segments, 18 segments, by 71.43% of participants, 10 participants) assigned when the participant expressed that the text editor was easy to use.
  - `+TOOL:easy image upload` (assigned to 1.98% of segments, 6 segments, by 35.71% of participants, 5 participants) assigned when the participants expressed that uploading the images was easy.

- **REVIEW:**

  - `+REVIEW:suggestions were useful` (assigned to 9.57% of segments, 29 segments, by 71.43% of participants, 10 participants) assigned when the participants found a review suggestion to be useful.
  - `+REVIEW:suggestions may be useful` (assigned to 1.98% of segments, 6 segments, by 28.57% of participants, 4 participants) assigned when participants expressed that the review suggestions may be useful.

- **TRANSLATE:**
  - `+TRANSLATE:easier/faster writing` (assigned to 7.92% of segments, 24 segments, by 78.57% of participants, 11 participants) assigned when the participants expressed that operations helped to make writing easier or faster.
  - `+TRANSLATE:choices-inspired writing` (assigned to 6.27% of segments, 19 segments, by 50.00% of participants, 7 participants) assigned when the participants expressed that the choices inspired them to enhance their writing while not directly inserting the choice into the tutorial.
  - `+TRANSLATE:image to instruction` (assigned to 5.28% of segments, 16 segments, by 57.14% of participants, 8 participants) assigned when participants liked the image DIY instruction operation.
  - `+TRANSLATE:rewrite worked well` (assigned to 4.29% of segments, 13 segments, by 64.29% of participants, 9 participants) assigned when the participants expressed that the rewrite operation worked well.
  - `+TRANSLATE:custom prompt useful` (assigned to 1.98% of segments, 6 segments, by 35.71% of participants, 5 participants) assigned when the participants found the custom prompt useful.

- **GEN TEXT:**

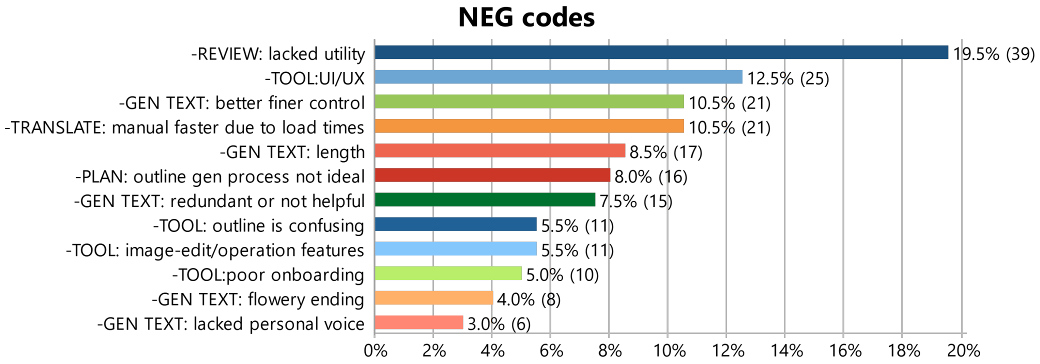


Figure 4.8: Negative codes

  - `+GEN TEXT:enhanced writing` (assigned to 5.61% of segments, 17 segments, by 71.43% of participants, 10 participants) assigned when the participants expressed that the generated text enhanced their writing.
  - `+GEN TEXT:good output` (assigned to 5.61% of segments, 17 segments, by 35.71% of participants, 5 participants) assigned when the participants expressed that the generated text was good.

4.3.3 Negative Codes

In this section, we list and describe the different high-level negative codes grouped by the categories.

- **REVIEW:**
  - `-REVIEW:lacked utility` (assigned to 19.50% of segments, 39 segments, by 85.71% of participants, 12 participants) assigned when the participants expressed that the review suggestions lacked utility for them.
- **TOOL:**

  - `-TOOL:UI/UX` (assigned to 12.50% of segments, 25 segments, by 71.43% of participants, 10 participants) assigned when the participants expressed a negative sentiment about the UI/UX of the prototype.
  - `-TOOL:image-edit/operation features` (assigned to 5.50% of segments, 11 segments, by 50.00% of participants, 7 participants) assigned when the participants had a negative reaction to the lack of basic image editing tools and features.
  - `-TOOL:outline is confusing` (assigned to 5.50% of segments, 11 segments, by 64.29% of participants, 9 participants) assigned when the participants found the outline step confusing.
  - `-TOOL:poor onboarding` (assigned to 5.00% of segments, 10 segments, by 50.00% of participants, 7 participants) assigned when the participants expressed that the onboarding was poor.

- **GEN TEXT:**
  - `-GEN TEXT:better finer control` (assigned to 10.50% of segments, 21 segments, by 57.14% of participants, 8 participants) assigned when the participants expressed that the generated text was difficult to control.
  - `-GEN TEXT:length` (assigned to 8.50% of segments, 17 segments, by 50.00% of participants, 7 participants) assigned when the participants expressed that the length of the generated text was problematic.
  - `-GEN TEXT:redundant or not helpful` (assigned to 7.50% of segments, 15 segments, by 57.14% of participants, 8 participants) assigned when the participants found the generated text to be redundant and not helpful.
  - `-GEN TEXT:flowery ending` (assigned to 4.00% of segments, 8 segments, by 35.71% of participants, 5 participants) assigned when the participants found that the generated text to have had an unnecessary and flowery ending sentence.

  - `-GEN TEXT:lacked personal voice` (assigned to 3.00% of segments, 6 segments, by 35.71% of participants, 5 participants) assigned when the participants expressed that they thought the final tutorial lacked their personal voice.

- **PLAN:**
  - `-PLAN:outline gen process not ideal` (assigned to 8.00% of segments, 16 segments, by 64.29% of participants, 9 participants) assigned when the participants expressed that the routine generation process was not ideal for them.

- **TRANSLATE:**
  - `-TRANSLATE:Manual faster due to load times` (assigned to 10.50% of segments, 21 segments, by 57.14% of participants, 8 participants) assigned when the participants expressed that it would have been faster to type manually rather than wait for the choices to load.

4.3.4 Recommendation codes

In this section, we list the different high-level recommendation codes that were assigned. We will discuss them in the next chapter.

- **REC: improve image-related features** (assigned to 17.24% of segments, 10 segments, by 50.00% of participants, 7 participants)
- **REC: improve outline flow** (assigned to 13.79% of segments, 8 segments, by 28.57% of participants, 4 participants)
- **REC: better control over generated text** (assigned to 12.07% of segments, 7 segments, by 35.71% of participants, 5 participants)

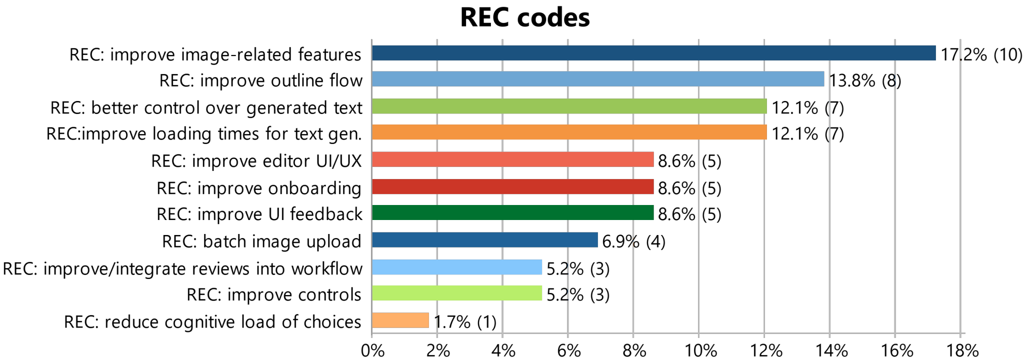


Figure 4.9: Recommendation codes

- **REC: improve loading times for text gen.** (assigned to 12.07% of segments, 7 segments, by 35.71% of participants, 5 participants)

- **REC: improve UI feedback** (assigned to 8.62% of segments, 5 segments, by 28.57% of participants, 4 participants)

- **REC: improve editor UI/UX** (assigned to 8.62% of segments, 5 segments, by 28.57% of participants, 4 participants)

- **REC: improve onboarding** (assigned to 8.62% of segments, 5 segments, by 21.43% of participants, 3 participants)

- **REC: batch image upload** (assigned to 6.90% of segments, 4 segments, by 28.57% of participants, 4 participants)

- **REC: improve controls** (assigned to 5.17% of segments, 3 segments, by 14.29% of participants, 2 participants)

- **REC: improve/integrate reviews into workflow** (assigned to 5.17% of segments, 3 segments, by 21.43% of participants, 3 participants)

- **REC: reduce cognitive load of choices** (assigned to 1.72% of segments, 1 segment, by 7.14% of participants, 1 participant)

Chapter 5

Discussion

In this section, we will discuss the results from the previous section, make some brief recommendations based on the study, and discuss limitations.

5.1 Discussion of Codes

5.1.1 Plan

Participants were asked if the prototype helped them in the process of outlining and planning the DIY tutorial. The following section discusses the evaluation codes assigned to participants in the category of `PLAN`.

Outline step was useful and helped structure the tutorial

The code `+PLAN:outline useful` was assigned to 21.45% of all segments and All 14 participants found the outline generation step useful for planning the DIY tutorial. The outline step made it faster to generate a first draft of the tutorial and made creating an outline for a DIY tutorial easier. Also, the code `+PLAN:helped structuring tutorial` was assigned to 11.88% of the segments, and 78.57% or 11 participants expressed it. The generated outline enabled the participants to help structure the DIY tutorial.

> *"I didn't have to think how I wanted the structure to be. I didn't have to think about the original ordering of the steps or a list of the materials required. Also Yeah, it helped me through my thinking process. What should I put, one by one"*
> *— Participant 5*

reduced the burden of the planning process, generated outline helped split and sequence steps

It relieved some of the burdens of the planning process of writing a DIY tutorial. The generated outline helped with structuring the DIY tutorial into different sections. First, the use of a template to split it into an introduction, supplies, steps, and conclusion. Second, the generated outline was even able to split the different tasks of the DIY project into different steps in the DIY tutorial which many participants found helpful.

> *"it already generated this really stable like step by step or not step by step, but like bigger scope outline for me and Yeah [...]" — Participant 2*

Precise description or standard steps for the DIY generated a good outline

Often, the more precise the outline description the participant gave, the closer the generated outline matched their DIY project and contained fewer errors in terms of sequencing or incompleteness. In five cases of the origami DIY project, the outline generated matched the DIY origami project accurately. Since the steps for folding an origami project are standard, in this case, it even broke the origami process into logically consistent steps like folding the body, then the legs, then the head.

Participants had a hard time describing their DIY projects

Though the code `-PLAN:outline gen process not ideal` was assigned to 8% of the segments, and 64.29% or 9 participants expressed it. Participants had a hard time trying to describe their DIY project or describe an outline for their DIY tutorial. Thus, faced difficulties with the planning process of writing a DIY tutorial. It was also assigned when the template generated by the outline differed from what the participants wanted in their DIY tutorial.

> *"It's just that like at the end, it felt like I had to delete a lot of it, like for example, it tried to struc-*

> *ture the origami process into multiple sets, steps like the base of the Peacock and then the actual unfolding of the Peacock. Well, for me it was just like, yeah, now we're going to fold. That's like a bunch of little steps." — Participant 6*

Though for certain participants like P06, the template generated was not ideal due to the nature of their DIY project. They removed tools, skills, and safety instructions from the supplies part of the template. They reduced their tutorial to just 1 major step and then inserted the images and instructions for folding the origami peacock. This was due to a lack of the option to edit the template of the generated DIY tutorial.

Participants wanted to edit the template of generated outline

This generation process not being ideal compounded with the code `-TOOL:outline is confusing` assigned to 5.5% of negative segments and expressed by 9 participants. It was assigned when participants expressed confusion in the outline step. So, participants had difficulty describing their DIY project and were confused by how the outline step worked. This was most likely due to the poor design and lack of instructions, as well as a lack of examples of what to do in the outline step. Thus, the participants' experience suffered due to the poor onboarding of the outline step. Hence the recommendation code `REC:improve outline flow` was derived from it, assigned to 8 segments.

Outline step was confusing due poor onboarding

However, once they had generated an outline, the confusion quickly dissipated. Overall, the outline step of the prototype helped users structure and plan for their DIY tutorial. but it would be ideal to improve the outline step to make it less confusing and easier for the users to understand what they need to do.

Make the outline step less confusing to improve it

5.1.2 Translate

Participants were asked if the prototype supported them with the act of translation in the writing process. The act of converting ideas, thoughts, and images into words. In this section, we will look at how the participants responded to that question.

Operations helped make writing easier and faster

The code +TRANSLATE:easier/faster writing occurred 24 times, expressed by 11 participants. The participants said the operations helped them find the right words and structure sentences. The code +TRANSLATE:choices inspired writing was assigned to 19 segments. 5 participants said that while they did not directly utilize the choices being shown by inserting them into the DIY tutorial. The choices inspired them to write better sentences.

Participants really liked the image DIY instruction operation but wanted more image related features

The code +TRANSLATE:image to instruction occurred in 16 segments; it was assigned when the participants had a positive impression of the image-based operations. Though participants expressed frustration that the prototype did not have basic image editing features and if they could generate the instructions using multiple images, the code -TOOL:image-edit/operation features was assigned to 11 segments. The recommendation code REC:improve image-related features was derived and assigned to 10 segments.

Rewrite operation worked well but it could be improved

The code +TRANSLATE:rewrite worked well was assigned to 13 segments. This was assigned to segments where the participants expressed that the rewrite operation worked well for them. 2 participants, expressed that they would want to use this to manipulate larger sections of the DIY tutorial than was possible in the prototype. Thus, the recommendation code REC:improve controls was derived from it and was assigned to 3 segments.

> *"I found it very helpful is that you can generate the steps like you just write, 'write the step about something' and then it generates the whole step for you."*
> *—Participant 5*

Participants found the custom prompt useful for their goals

The code +TRANSLATE:custom prompt was useful occurred 6 times in the coded segments. This was assigned to the 5 participants who used it and found it helpful for their intended goals. Participant P05 used the freeform/custom prompt to generate a new step after the current one. Another, participant P13 used it to generate a glossary of terms used in the DIY tutorial.

The code `-TRANSLATE:manual faster due to load times` was assigned to 21 segments. 8 participants expressed the sentiment that it would be faster at times to write the text manually just because of the amount of time they had to wait to generate the text. The recommendation code `REC:improve loading times for text gen.` was assigned to 7 segments and derived from this.

Sometimes typing manually was faster

Overall, the participants found the operations provided by the prototype helpful for the act of translation. We could add basic image editing and related features such as rotating and cropping the image and DIY instruction generation from multiple images. Certain operations like rewrite could also be improved to generate more tokens. The issue of loading times is definitely a problem that cannot be as easily addressed. Since the part adding latency is from the API response from the OpenAI servers. We could potentially look at other LLMs or run an LLM locally to improve the load times.

Overall prototype helped with translation, but load times are a significant issue

5.1.3 Review

> *"It reminded you of some stuff that you may be missed. So it's actually that is a good point for example. giving hints." — Participant 1*

The code `+REVIEW:suggestions were useful` was assigned to 29 segments. 10 participants found the review suggestions useful for improving their tutorial. They said that it helped them check their process and catch small mistakes. It also offered suggestions that were useful to improve the DIY tutorial.

Review caught missed details and reminded you to check process

> *"The review it was very nice. I just clicked it once, but yeah, it gave like very stupid suggestions, but I think that for a larger project it would actually work very well. " —Participant 5*

Suggestion may be useful on larger projects

The code `+REVIEW:suggestions may be useful` occurred 6 times and was assigned to 4 participants who did not use the review to improve their tutorial but still expressed that the suggestions may be useful in larger projects. The participants reasoned that the review suggestion would allow them to focus on specific parts of the tutorial, which would be useful in a large project.

> *"Why do you want to have it a trapezoid shape? So why I don't want to reason it, its a DIY tutorial. I don't want to go into detail about well."*
> *— Participant 9*

Reviews lacked utility, were pointless

On the other hand, the code `-REVIEW:lacked utility` was assigned to 39 segments and was expressed by 10 participants. The participants pointed out that a lot of the review suggestions seemed pointless, generic, and excessive. One review suggestion suggested to the participant P09 to give a reason for choosing a trapezoid shape for their elephant. The participant did not want to explain a reason for their vision of what their DIY elephant should look like. They thought including such a reason would not be helpful in a DIY tutorial.

Reviews were cumbersome to apply

While some review suggestions were useful, participants often found it cumbersome to integrate the review into their DIY tutorial. They had to manually copy a review suggestion into a rewrite or freeform custom prompt operation. Hence, The recommendation code `REC:improve/integrate reviews into workflow` was derived. Ideally, the review suggestions would only point out glaring issues or things that interest the tutorial author.

Overall sentiment on reviews was mixed

Overall, participants had a more mixed response to the reviews, with every participant who expressed they were useful also expressing that they lacked utility. It often generated too many suggestions and some were really pointless. While the review can definitely be better integrated into the workflow. Suggesting that reviews were not as helpful in the prototype as planning and translating. Further research needs to be performed to understand how to improve the feedback system for the user for a DIY tutorial.

5.1.4 Generated text

Generated text enhanced writing and was good

This section discusses the sentiments expressed by the participants of the generated output created by the LLM. The codes `GEN TEXT:enhanced writing` and `GEN TEXT:good output` were assigned to 17 segments each. This was assigned when the participants expressed that the generated text from the different operations enhanced their writing or gave them a good output that they could use in their DIY tutorial.

> *"Sometimes the prompt was like great, but like. But then there's one spot you have to correct again" —Participant 4*

Generated text output was difficult to control

The code `-GEN TEXT:better finer control` which occurred 21 times, was used to express the desire by the participants to control the output of the generated text better. Participants often had to rewrite or delete parts of the generated text to use in their tutorial. Participants expressed that they wished to write precise and terse instructions for their DIY tutorial. However, the amount and type of text generated did not allow them to do that.

> *"It's kind of like when I just expected like 1 precise sentence that describes what I want, and then it gives me, like, this whole paragraph." — Participant 4*

Generated text length was either too long or too short

The generated text was either often too long or too short compared to what the participant wanted. The code `-GEN TEXT:length` was assigned to 17 segments for when the participant expressed issues with the length of the generated output. Participants wanted finer control over the length of the generated text. This also lead to an issue where participants had to read long pieces of generated text to figure out if it was useful to them or not. Thus, the recommendation code `REC: reduce cognitive load of choices` was derived from this and assigned to 1 segment.

> *"It ended sentences like it would give your bunny, a more beautiful look to your, your more cohesive look or something like that with all the same stuff. And I was like, that's a little bit repetitive and I would wouldn't like to if I would have read it and be reminded of it in every sentence." — Participant 3*

Generated text was redundant

The code `-GEN TEXT:redundant or not helpful` was assigned to 15 segments when the participants expressed that they found the generated text to be redundant and not helpful. Often, all the choices would have the same single issue, or the choices would be repetitive. Participants also said that the generated text had a flowery or unnecessary ending, the code `-GEN TEXT:flowery ending` was assigned to 8 segments. They always had to remove the last sentence from the generated output.

Generated text sounded like AI, not me

The code `-GEN TEXT:lacked personal voice` occurred 6 times and was assigned to 5 participants who said that the generated text did not sound like their voice. It instead sounded like ChatGPT. Participant P10 said that *"[...] the tutorial ended up sounding like ChatGPT, which is not that Nice because if I write something, I would like it to sound like me in the end."* This points to an issue of ownership over the final DIY tutorial written.

difficult to figure out the root cause of issues with generated text

The various issues with the generated text could be indicative of improving the design of prompts for the various operations. It could also be characteristic of the specific LLM that we are using, and thus, other LLMs may or may not suffer from one or all of these issues.

Users should have better control over generated text

Based on these various issues with the generated text. The recommendation code `REC:better control over generated text` was derived and assigned to 7 segments. Users should have finer control over the length and tone of the generated output, and the generated text, by default, should be precise and to the point instead of long paragraphs.

5.1.5 Tool

Participants had positive impressions of the prototype

This section discusses the participants' experience with respect to the prototype itself and not one specific aspect as done in the previous sections. The code `+TOOL:UI/UX` was assigned to 31 segments where the participants expressed that they had a positive impression of the UI/UX of the prototype. 12 Participants expressed that the tool was easy to use, enjoyable, and helpful and that the writing experience of using the prototype was nice.

Text editor was easy to use

The code `+TOOL:text editor easy to use`, occurred 18 times, 10 participants found the text editor easy and intuitive to use. The participants. Hence, they could easily use the text editor to write and edit text for their DIY tutorial.

Image upload was easy, but batch upload, drag and drop would be nice

6 segments were assigned the code `+TOOL:easy image upload` and 5 participants said that adding images to the DIY Tutorial was easy and they did not face issues with uploading images. However, participants expressed that it would be nice to be able to upload multiple images. Participants complained about being unable to drag and drop images into the editor. Thus the recommendation code `REC:batch image upload` was derived from it and assigned to 4 segments.

Participants did get annoyed at the lack of feedback and some of the UI/UX of the editor

The code `-TOOL:UI/UX` was assigned to 25 segments when 10 participants had a negative impression of the UI/UX of the prototype. Three participants expressed that they found the text editor to be not intuitive. One participant had issues with the formatting of the text. Two participants disliked that the editing of the tutorial was locked when an operation was in progress. Another participant said they disliked being unable to cancel an operation when it is in progress. A few participants expressed that there was a lack of feedback from the UI for saving reviews when the choices have loaded and that, at times, it was hard to understand if the cursor had selected something. From these, two recommendation codes were derived `REC:improve UI feedback` and `REC:improve editor UI/UX` both were assigned to 5 segments each.

The onboarding experience was poor

The code `-TOOL:poor onboarding` was assigned to 10 segments and was expressed by 7 participants who said they had a poor onboarding experience. One participant said that they were unaware that the document was saved and were scared that their effort would not be saved. Even though the prototype autosaves the document whenever an edit is performed, Another participant said they were unaware they could undo changes to the text. Thus the recommendation code `REC:improve onboarding` was derived from this and assigned to 7 segments

Overall, participants had a generally favorable impression of the prototype and just wanted to fix a few quirks of its usage. These are usability issues that can be easily fixed.

5.2 Recommendations for Prompt Design

In this section, we will make some brief recommendations for improving the design of LLM prompts for DIY tutorials based on our discussion in the previous section.

Prompts should allow precise control over length, generally be shorter

Our users preferred that the generated text be shorter since they could always elaborate on it if required using the operations. Thus, prompts that generate DIY instructions or sentences should always tend towards generating a short and precise sentence. It would be ideal if controls were given to alter the length of the generated output precisely, for example, using a slider. The prompt design should accommodate the ability to set the length of the generated text. This should also allow the user to change the maximum number of tokens generated for the prompt, allowing the user to manipulate large pieces of text or the entire DIY tutorial if required.

Prompts should enable finer and better control over output

The generated text often contained an extra flowery or redundant sentence. The LLM should either be fine-tuned not to generate redundant sentences or the prompt needs to be designed so that it does not add redundant sentences. Ideally, the generated output should be mechanical, precise, and terse. So that if the user wants to embellish the text, they can do so later. The user should be able to control the content of the output text precisely as they see fit. The design of the prompt should enable fine-grained control over the generated text.

File-based prompts and text generation over multiple files

Our participants overwhelmingly like the image-based controls. Since LLMs are generally moving towards multimodal inputs, prompts should allow the user to send a file containing data about the DIY project and be able to generate instructions for the DIY tutorial. Prompts need to be designed to allow the users to upload different file formats like text, audio, image, or video and generate DIY instructions from these uploaded files. The prompts should also allow multiple files to be selected or included to generate text.

5.3 Recommendations for UI/UX improvements

In this section, we recommend a few proposals for improving the UI/UX of the prototype itself to improve its usability.

Onboarding needs to be overhauled and outline step needs to be improved

The poor onboarding experience of the prototype resulted in participants not being aware of various features of the prototype as well as leading to confusion in the outline step especially. The outline step needs to be overhauled with better hints and instructions to allow the user to understand what they need to do to create a DIY tutorial outline. If the details about the supplies required for the DIY project are auto-generated from an image of the completed DIY project. That would make things really simple for the user. So, each part of the tutorial header or supply, like materials, tools, skills, and estimated time, will have a text box that can be manually edited. This is coupled with a short list of steps to complete the DIY project that can be autogenerated based on the information from the tutorial header. The list of steps should be easy to edit and manipulate for the user. Thus creating a sort of a plan for the DIY tutorial. This is then used to generate the final template outline for the DIY tutorial. The onboarding done in this manner would reduce confusion, improve loading times and help onboard the user better.

Review suggestions as comments in the editor

Review suggestions should appear in the text editor as comments on the appropriate section or piece of the DIY tutorial and should be applicable using a single button press.

Improve image-related features

The features related to images should be improved to allow for basic editing through cropping and rotation of the image. The user should be able to change the meta-data of the images like title and alt. There should be a gallery that allows for uploading all the images related to the DIY project and simply allows dragging or linking the image in the right place. This should support both batch image upload and dragging and dropping the images into the gallery or editor. The metadata for the image could also be autogenerated using the model. Which can later be modified by the user. The editor should allow the selection of multiple images.

Better feedback and improved editor UI/UX

The editor should include sound cues and better visual feedback for actions performed by the user or the system. This would allow for a better experience for the user.

5.4 Limitations of our study

Controlled Experiment

In this section, we will discuss the limitations of our user study. The study design does not capture the reality of writing DIY tutorials, since it was done in a controlled environment under stringent time constraints. Usually, participants would prototype, make mistakes, document their work, and iterate on a DIY project for longer time periods than the 2 hours that the participants performed the study. We also recommended a specific workflow to the participants for writing the tutorial by suggesting they write down the actions on paper while they are working on the

DIY project. Different participants might have very different workflows while working on their DIY projects.

Comparing between different LLMs

The study was done as within groups qualitative user study, if a between-groups user study was done with a baseline control. The impact of LLM on the writing of the DIY tutorial would be more apparent. One case is where no LLM is used, and the other is where an LLM is used. However, such a design does not account for differences in the working styles of different participants. It would be interesting to compare the results between LLMs and the participants working without LLMs.

Experienced DIY tutorials might have different insights

Our study also did not have a lot of makers with experience writing DIY tutorials; there might be some key insights we might be missing due to the nature of the demographics. So, the study could be performed with makers who have experience writing DIY tutorials.

Chapter 6

Summary and future work

In this chapter, we conclude our thesis by providing a summary of our work and present opportunities for future work.

6.1 Summary and Contributions

This thesis explored how large language models can be used to support DIY tutorial authorship.

Powerful, extensible artifact for LLM-powered writing of DIY tutorials

First, we developed a prototype artifact, "DIYmate", that is built to help writing DIY tutorials. . It comes with a powerful text editor and various LLM-powered operations designed to support the task of DIY tutorial writing. It can be extended easily to support other LLMs. We have also documented our design process for using the writing goals design space to create LLM-based writing support tools.

Evaluated the prototype with a user study

Second, we evaluated our prototype with a qualitative user study to understand how DIY tutorial authors would use it to write a DIY tutorial. We used the 3 tasks in the writing process: Plan, Translate, and Review to understand where the prototype worked and where it failed. We found that the prototype helped with the planning process for the participants to help structure the tutorial but the outline step can be improved. Participants found the prototype was extremely useful for the task of translating ideas and images into words. However, participants had to edit the text to fit the DIY tutorial. Lastly, for reviewing, participants found most of the suggestions not useful. We also made some recommendations for improving the design of prompts and the UI/UX of the prototype.

6.2 Limitations and Future work

Study usage under normal conditions

One limitation was that we performed the user study in a controlled environment under stringent time constraints to complete a DIY project and write a tutorial. Future work could take a look at providing the prototype online or allowing makers to access it over a longer period of time to complete a DIY project as they would do it normally and conduct a user study that is more naturally suited to the rythm of a DIY project.

Future work on different LLMs or local LLMs

Another limitation is that we only used one LLM model, ChatGPT4.0, in our study. The results might differ if other LLMs are used. Since the capabilities and performance characteristics of different models are so different. Furthermore, our prototype was built to be able to easily swap in another LLM model, only having to rewrite the prompts. Future work could focus on using other LLMs or local LLMs and doing a comparative study between different models for the writing task to see how the results differ.

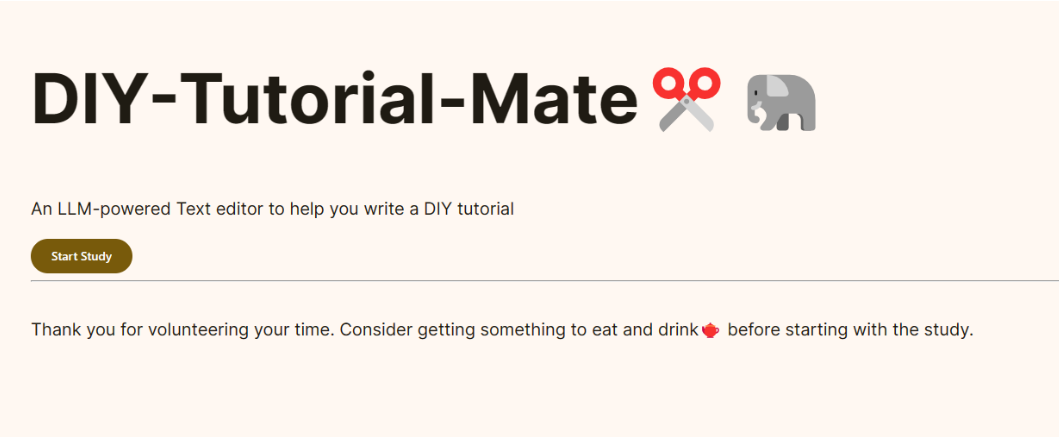
Using LLMs for process-based documentation

One possibility that future work could explore is how useful LLMs could be for process documentation in makers spaces and fab labs. Our prototype focused on the end product of a DIY tutorial, but makers often prototype, test, and work on their DIY projects in an iterative fashion. Thus, LLMs can be used to add value to the documentation process of building a DIY project. Since build-in-progress by Tseng [2016] has been shut down, there is potential for future work to integrate LLMs and create a new website and community.

Appendix A

Appendix A

A.1 Additional screens of the UI

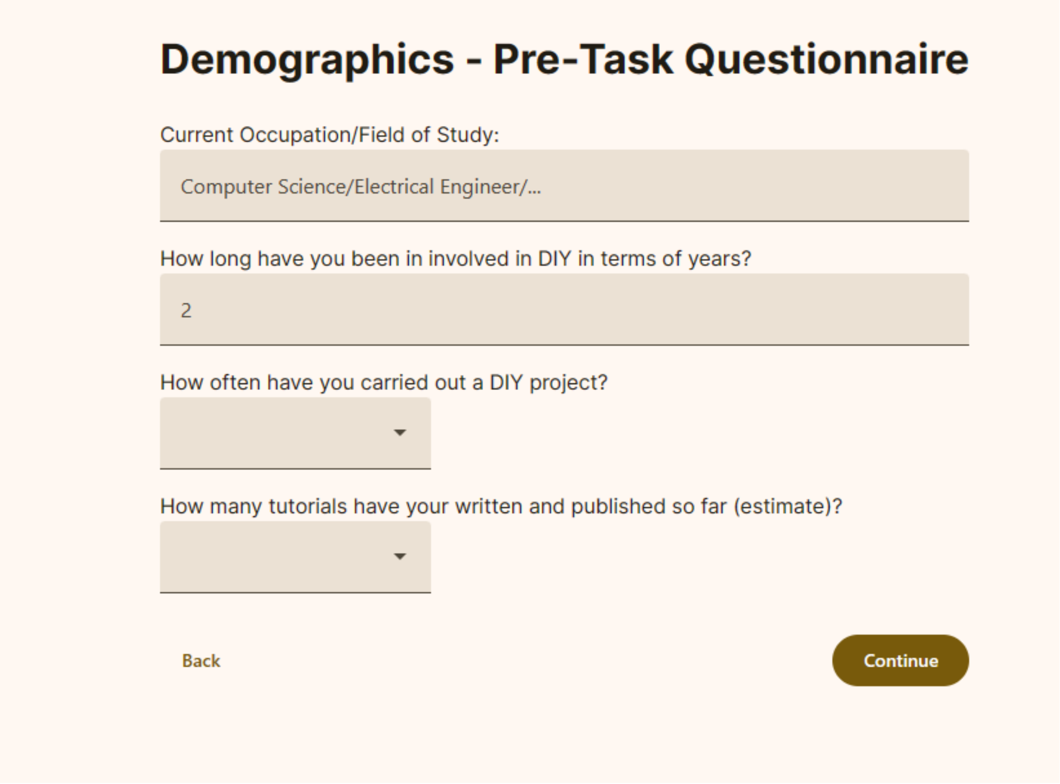


DIY-Tutorial-Mate ✂️ 🐘

An LLM-powered Text editor to help you write a DIY tutorial

Thank you for volunteering your time. Consider getting something to eat and drink 🫖 before starting with the study.

Figure A.1: Welcome screen for the participant



Demographics - Pre-Task Questionnaire

Current Occupation/Field of Study:

Computer Science/Electrical Engineer/...

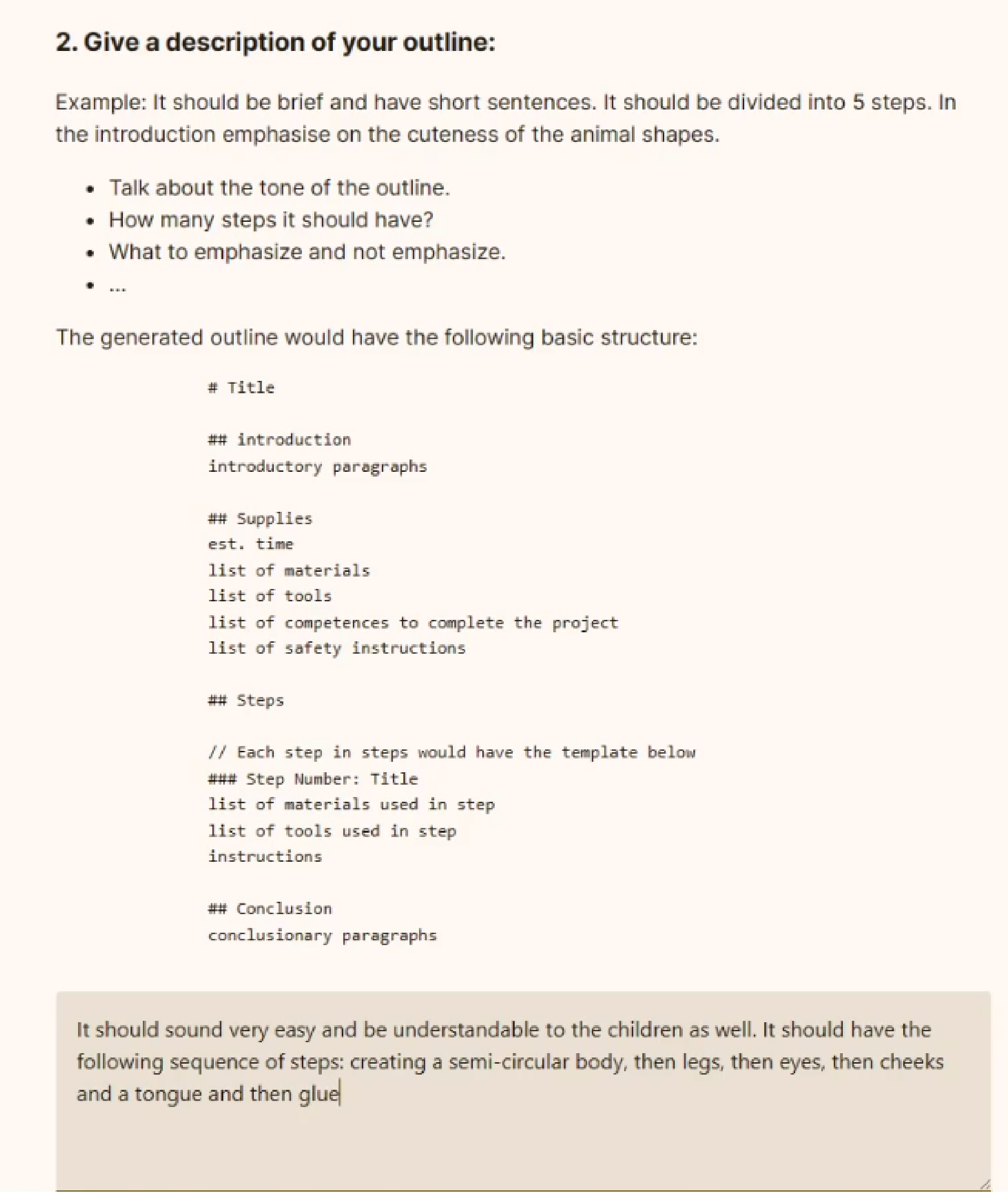
How long have you been in involved in DIY in terms of years?

2

How often have you carried out a DIY project?

How many tutorials have your written and published so far (estimate)?

Figure A.2: Demographics form shown to the participant



2. Give a description of your outline:

Example: It should be brief and have short sentences. It should be divided into 5 steps. In the introduction emphasise on the cuteness of the animal shapes.

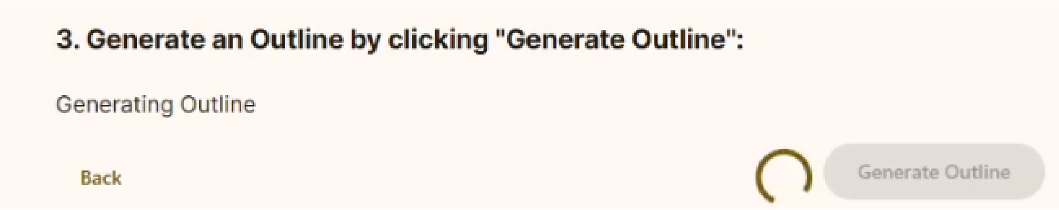
- Talk about the tone of the outline.
- How many steps it should have?
- What to emphasize and not emphasize.
- ...

The generated outline would have the following basic structure:

```
# Title

## introduction
introductory paragraphs

## Supplies
est. time
list of materials
list of tools
list of competences to complete the project
list of safety instructions

## Steps

// Each step in steps would have the template below
### Step Number: Title
list of materials used in step
list of tools used in step
instructions

## Conclusion
conclusionary paragraphs
```

It should sound very easy and be understandable to the children as well. It should have the following sequence of steps: creating a semi-circular body, then legs, then eyes, then cheeks and a tongue and then glue

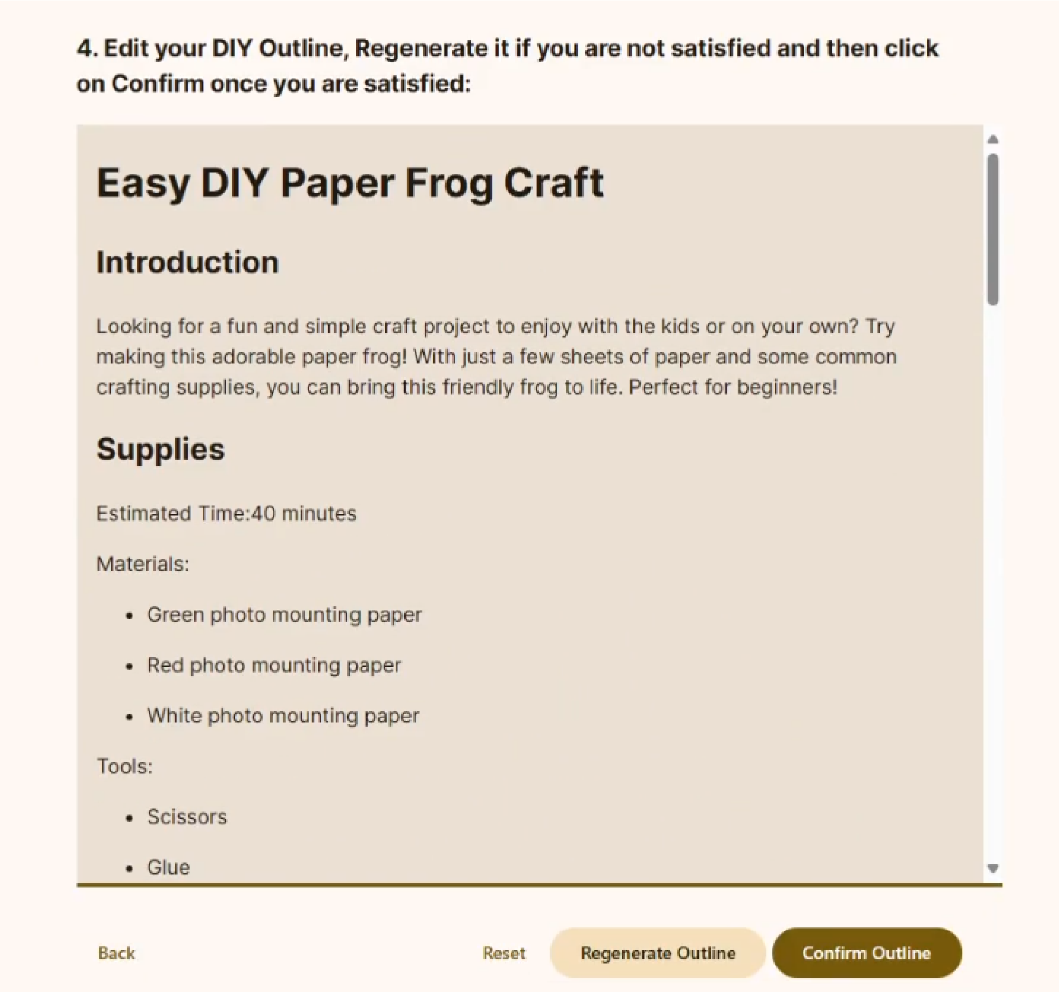
Figure A.3: Outline description step in outline step



3. Generate an Outline by clicking "Generate Outline":

Generating Outline

Figure A.4: Generating the outline



4. Edit your DIY Outline, Regenerate it if you are not satisfied and then click on Confirm once you are satisfied:

Easy DIY Paper Frog Craft

Introduction

Looking for a fun and simple craft project to enjoy with the kids or on your own? Try making this adorable paper frog! With just a few sheets of paper and some common crafting supplies, you can bring this friendly frog to life. Perfect for beginners!

Supplies

Estimated Time:40 minutes

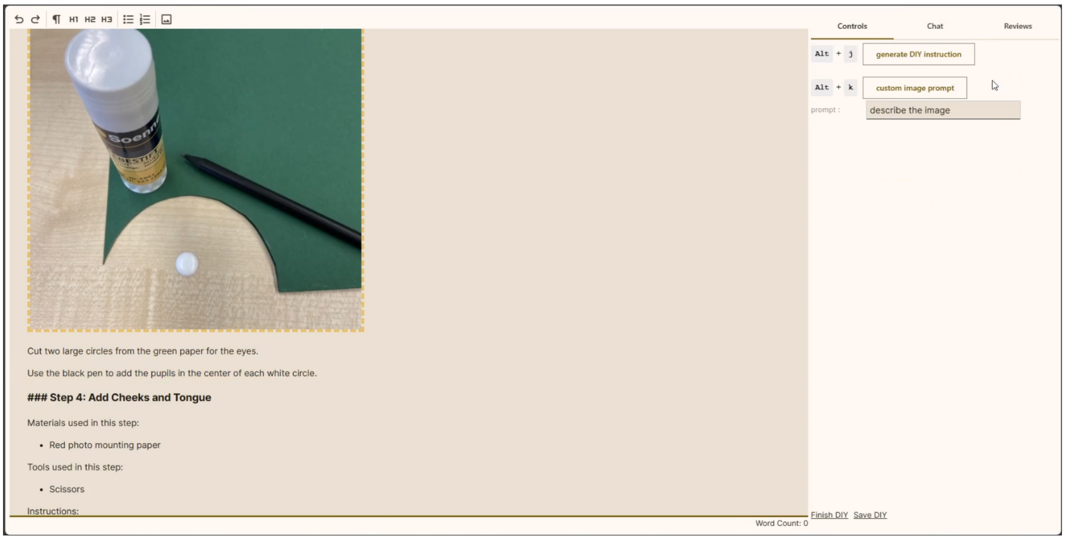
Materials:

- Green photo mounting paper
- Red photo mounting paper
- White photo mounting paper

Tools:

- Scissors
- Glue

Figure A.5: Generated outline in an editor, the user can confirm this outline for the editor



prompt : describe the image

Cut two large circles from the green paper for the eyes.

Use the black pen to add the pupils in the center of each white circle.

Step 4: Add Cheeks and Tongue

Materials used in this step:

- Red photo mounting paper

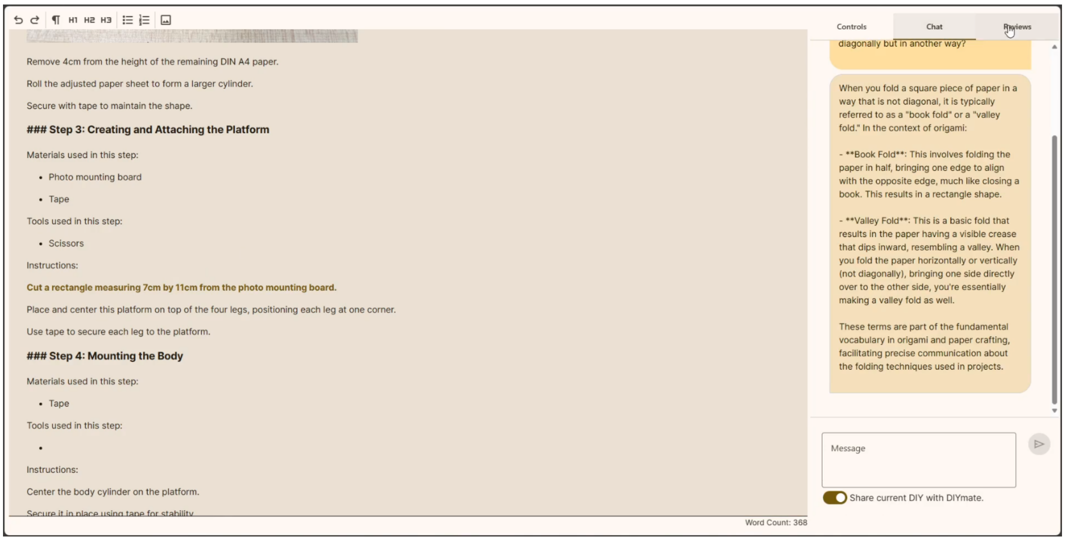
Tools used in this step:

- Scissors

Instructions:

Word Count: 0

Figure A.6: Image operations shown in the sidebar on the right



Remove 4cm from the height of the remaining DIN A4 paper.

Roll the adjusted paper sheet to form a larger cylinder.

Secure with tape to maintain the shape.

Step 3: Creating and Attaching the Platform

Materials used in this step:

- Photo mounting board
- Tape

Tools used in this step:

- Scissors

Instructions:

Cut a rectangle measuring 7cm by 11cm from the photo mounting board.

Place and center this platform on top of the four legs, positioning each leg at one corner.

Use tape to secure each leg to the platform.

Step 4: Mounting the Body

Materials used in this step:

- Tape

Tools used in this step:

-

Instructions:

Center the body cylinder on the platform.

Secure it in place using tape for stability

Word Count: 368

diagonally but in another way?

When you fold a square piece of paper in a way that is not diagonal, it is typically referred to as a "book fold" or a "valley fold." In the context of origami:

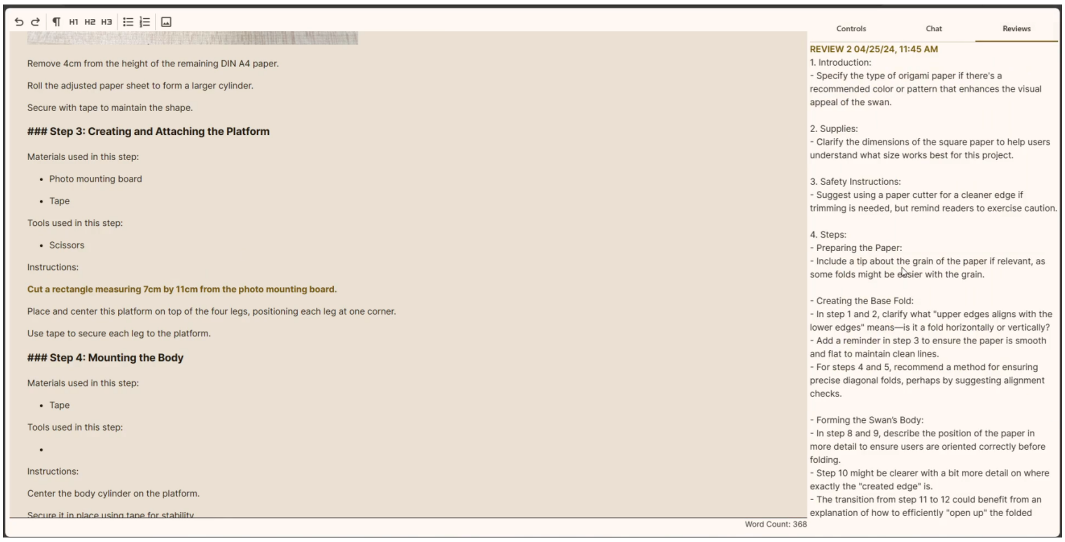
- **Book Fold**: This involves folding the paper in half, bringing one edge to align with the opposite edge, much like closing a book. This results in a rectangle shape.

- **Valley Fold**: This is a basic fold that results in the paper having a visible crease that dips inward, resembling a valley. When you fold the paper horizontally or vertically (not diagonally), bringing one side directly over to the other side, you're essentially making a valley fold as well.

These terms are part of the fundamental vocabulary in origami and paper crafting, facilitating precise communication about the folding techniques used in projects.

● Share current DIY with DIYmate.

Figure A.7: Chat tab shown in the sidebar



Remove 4cm from the height of the remaining DIN A4 paper.

Roll the adjusted paper sheet to form a larger cylinder.

Secure with tape to maintain the shape.

Step 3: Creating and Attaching the Platform

Materials used in this step:

- Photo mounting board
- Tape

Tools used in this step:

- Scissors

Instructions:

Cut a rectangle measuring 7cm by 11cm from the photo mounting board.

Place and center this platform on top of the four legs, positioning each leg at one corner.

Use tape to secure each leg to the platform.

Step 4: Mounting the Body

Materials used in this step:

- Tape

Tools used in this step:

-

Instructions:

Center the body cylinder on the platform.

Secure it in place using tape for stability

REVIEW 2 04/25/24, 11:45 AM

1. Introduction:
- Specify the type of origami paper if there's a recommended color or pattern that enhances the visual appeal of the swan.

2. Supplies:
- Clarify the dimensions of the square paper to help users understand what size works best for this project.

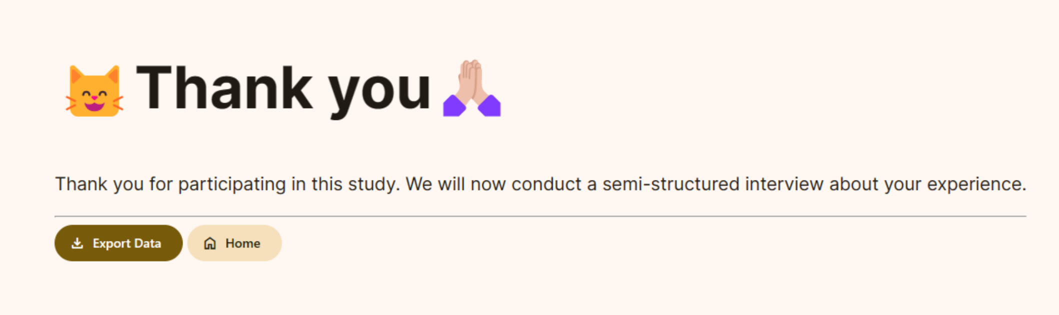
3. Safety Instructions:
- Suggest using a paper cutter for a cleaner edge if trimming is needed, but remind readers to exercise caution.

4. Steps:
- Preparing the Paper:
- Include a tip about the grain of the paper if relevant, as some folds might be easier with the grain.

- Creating the Base Fold:
- In step 1 and 2, clarify what "upper edges aligns with the lower edges" means—is it a fold horizontally or vertically?
- Add a reminder in step 3 to ensure the paper is smooth and flat to maintain clean lines.
- For steps 4 and 5, recommend a method for ensuring precise diagonal folds, perhaps by suggesting alignment checks.

- Forming the Swan's Body:
- In step 8 and 9, describe the position of the paper in more detail to ensure users are oriented correctly before folding.
- Step 10 might be clearer with a bit more detail on where exactly the "created edge" is.
- The transition from step 11 to 12 could benefit from an explanation of how to efficiently "open up" the folded

Figure A.8: Reviews tab shown in the editor



😸 Thank you 🙏

Thank you for participating in this study. We will now conduct a semi-structured interview about your experience.

Figure A.9: Screen shown at the end of the study thanking the user

A.2 Prompts used in the prototype

In this section, we list out all the prompts used in the different operations in the prototype.

The prompts usually have a role attached to them. The role decides who the message is intended for or from. The system role is used for messages by the system. The user role is used to get the API to respond to a prompt crafted in the application and designates messages from the user.

A.2.1 Continue

System You are a DIY Tutorial Assistant helping the user to continue write a DIY tutorial. **User** DIY: [text before the cursor] Continue the Tutorial:

A.2.2 Elaborate

System You are a DIY Tutorial assistant helping the author to elaborate parts of their DIY tutorial.

User DIY: [tutorial text] Describe "[text selection]" in more Detail:

A.2.3 Freeform

System You are a DIY tutorial assistant.

User DIY: [tutorial text] [custom instruction by user]

A.2.4 Conclusion

System You are a DIY tutorial assistant.

User DIY: [tutorial text without conclusion] Generate a conclusion for this DIY tutorial:

A.2.5 Introduction

System You are a DIY tutorial assistant.

User DIY: [tutorial text without introduction] Generate an introduction for this DIY tutorial:

A.2.6 Image instruction

System You are a DIY Tutorial Assistant helping the user generate DIY instruction from an image

User DIY text before the image: [before text] DIY text after the image: [after text] Image has title [title] and alternate text [alt]. [image data] Generate a SINGLE DIY instruction from the above image ending with '':

A.2.7 Custom image prompt

System You are an Assistant for a DIY tutorial who gives an SINGLE answer to the user for their prompt about an image.

User DIY text before the image: [before text] DIY text after the image: [after text] Image has title [title] and alternate text [alt]. [image data] Generate a response for the prompt {[custom Instruction]}} for the above image ending with '':

A.2.8 Meta prompt

System You are a Prompt Engineer. You suggest ONLY a SINGLE effective and helpful custom prompt instruction for the above text from a DIY Tutorial.

User DIY: [tutorial text] Next prompt:

A.2.9 Next sentence

System You are a DIY Tutorial sentence generator helping the user generate the next sentence.Just repond with next sentence ONLY. **User** DIY: [tutorial text with a blank in the current sentence] Next Sentence:

A.2.10 Outline

System You are a DIY Tutorial Assistant that generates outline for a DIY Tutorial for the author to start with.

User

Given the description of the DIY Project: [project description] The outline to be generated is described as follows: [outline description]

Generate a DIY tutorial outline in the following JSON format:

~~~
```JSON
{
"title": "Title of the DIY Project",
"heroshot_alt_text": "Alternate text for the hero shot",
"introduction": "Introduction to the DIY Project",
"materials":["material 1","material 2"],
"tools":["tool 1","tool 2"],
"estimated_time":"Estimated time to finish the project",
"competences":["competence 1","competences 2"],
"safety_instruction":["safety 1","safety 2","safety 3"],
"steps":[
    {
        "index": 0,
        "title": "step title",
        "image_alt_text":"Alt. text for image",
        "materials_in_step":["material 1","material 2"],
~~~

```
            "tools_in_step":["tool 1","tool 2"],
            "instructions":["instruction 1","instruction 2"]
        },
        {
            "index": 1,
            "title": "step title",
            "image_alt_text":"Alt. text for image",
            "materials_in_step":["material 1","material 2"],
            "tools_in_step":["tool 1","tool 2"],
            "instructions":["instruction 1","instruction 2"]
        }],
"conclusion":{
        "final_image_alt_text":"Alt. text for final image",
        "text":"Summarize the DIY tutorial"
}
}
```

A.2.11 Replace

System You are a DIY Tutorial assistant helping the user with filling in the blanks in their tutorial. Respond ONLY with text that fits the blank. Omit any introductory or conclusionary text.

User DIY: [tutorial text with blank] Sentence before blank: [sentence before blank] Fill in the Blank given by blank with [number of words in text]:

A.2.12 Rewrite selection

System You are a DIY Tutorial assistant.Helping rewrite the DIY tutorial. Just repond with rewritten text ONLY.

User DIY: [text before and after selection] The text to be rewritten that fills in the blank: [selection text] Rewritten to be [user how to rewrite] :

A.2.13 Rewrite Sentence

System You are a DIY Tutorial assistant.Helping rewrite the DIY tutorial. Just repond with rewritten text ONLY.

User DIY: [text before and after selection] The text to be rewritten that fills in the blank: [selection text] Rewrite this sentence to be [user how to rewrite] , only rewrite this specific sentence and nothing more:

Appendix B

Appendix B

B.1 Instructions for DIY task and Informed Consent

Study Task Instructions – Using LLMs to support DIY Tutorial Authorship

The task is to complete a fun and exciting craft DIY project on the theme of 'Animal' and write a DIY tutorial about it using the DIYMate App. You will be provided with a list of supplies given below and can choose any material to include or exclude in your project. You are free to use any technique or skill to complete the task to craft a DIY animal. Choose any animal that resonates with you if you think it is feasible to complete it within an hour. Since you also need to write and complete a DIY tutorial about it. You will also need to click pictures of your DIY project and upload them on to the app.

A list of supplies:

- Construction Paper
- Origami paper
- Photo Mounting Board paper
- Felt-tip pens
- Pencils
- Eraser
- Scissors
- Exacto-knife
- A4 Paper
- Ruler
- Glue

Things to consider:

1. You will be provided with an iPhone for the purpose of taking pictures of the DIY project and uploading the images on the tutorial. Let me know if you want to upload pictures so that I can help you get the images on to the computer to upload them into the app.
2. You can search for the general techniques and skills on the internet if you are unfamiliar with something you want to do in your DIY project. But please don't search and copy an existing DIY tutorial to complete your task.
3. If you have any questions or want to take a break, please don't hesitate to ask.
4. Your screen and your audio will be recorded when using the application.
5. After your task you will be asked to participate in a semi-structured interview about your experience with the task.

Some ideas to get you to brainstorm:

- Origami Cranes, Cats, Dogs or Rabbits
- Paper sheep 🐏
- Popup Penguin
- Long-tongued Crocodile.

Informed Consent Form

Using Large Language Models to support DIY Tutorial Authorship

PRINCIPAL INVESTIGATOR: Shailesh Iyer
Media Computing Group
RWTH Aachen University
Phone: +4915753620804
Email: shailesh.iyer@rwth-aachen.de

Purpose of study: The goal of this study is to understand how large language models can be used to support DIY tutorial authorship. Participants will be asked to complete a DIY project and write an DIY Tutorial for their DIY project on the DIYMate app. The App is powered by ChatGPT and has different kinds of controls to help assist participants with writing a DIY tutorial.

Procedure: Participation in this study involves 2 phases. In the first phase, you will first be asked to fill out a demographic form in the DIYMate application. After submitting that form, you will be presented with DIY Task and asked to create an outline for your tutorial once you have finalised your DIY project task. You will be asked to complete their DIY Project and write a DIY tutorial about it while using the controls and systems available in the DIYMate app. This should take an hour to complete.

In the second phase, after you have completed your DIY Project and tutorial you will be asked to participate in a semi-structured interview about your experience writing a DIY tutorial with the DIYmate app.

Risks/Discomfort: You may become fatigued during participation in the study. You can ask to take a break at any point in time during the study, and will be given opportunities to take a break. You will also use sharp tools so caution is advised to safely cut things and not injure yourself. Should the completion of DIY task or participation in the interview become distressing to you, it will be terminated immediately.

Benefits: This study will allow us to better understand how large language models can be integrated into writing applications to help support writing DIY tutorials.

Alternatives to Participation: Participation in this study is voluntary. You are free to withdraw or discontinue participation at any point in time during the study.

Costs and Compensation: You are participating in this study voluntarily and involves no cost to you. There will be snacks and drinks provided to you during and after the participation.

Data Recorded: The screen that you are using the application on and your audio will be recorded during the course of the first phase when you are using the application. In the second phase, your audio will be recorded in the interview. At no point will your video be recorded. Only the screen and audio data relevant to the study will be recorded.

Confidentiality: All information collected during the study period will be kept strictly confidential. You will be identified through identification numbers. No publications or reports from this project include any identifying information on any participant. If you agree to join this study, please sign your name below.

____ I have read and understood the information on this form.
____ I have had the information explained to me.
____ I give my permission to record my screen usage and audio for this study.

______________________________	______________________________	____________
Participants Name	Participant's Signature	Date
	______________________________	____________
	Principal Investigator	Date

If you have any questions regarding this study, please contact Shailesh Iyer at +4915753620804 or email: shailesh.iyer@rwth-aachen.de

B.2 Study Protocol

B.2.1 Context

DIY tutorials are important for knowledge exchange in the DIY community. They help preserve the vitality and spirit of the community. But, they suffer from various issues due to poor quality like, incomplete lists of materials and tools; poor communication due to bad formatting of text and images; not anticipating the skills of the DIY maker required to complete a DIY project.

Large language models have been shown to have various broad capabilities applicable to a variety of writing tasks with very little training. They have been shown to be good at generate a step by step sequence of events and can even generating an outline from a small description of the document.

For this reason, we conducted a study where we sought to investigate if Large language models can help support the task of writing a DIY tutorial for a DIY tutorial author. We built a prototype and conducted a user study of makers using the prototype.

B.2.2 Research Question

> "How can we use LLMs to assist DIY Tutorial Authors in authoring/writing high-quality DIY tutorials?"

we further split this question down into 3 distinct parts as relating to the cognitive process model of writing: 1. Can we use ChatGPT to improve planning the DIY tutorial? 2. Can we use ChatGPT to improve writing the DIY tutorial? 3. Can we use ChatGPT to improve reviewing the DIY tutorial.

B.2.3 Task

The participant will be given an open ended DIY project to craft an animal using the craft materials given and will be asked to write the DIY tutorial alongside completing this DIY project. We chose this task for its simplicity, thus allowing most participants to complete the DIY without the need for them referring to an existing tutorial. Since based on the kind of craft they are doing they can go about it in different ways thus allowing for their own creativity to be represented. Thus, different kinds of DIY tutorials will be written. The list of supplies is given in the next section.

B.2.4 Supplies for DIY task

Materials

- Construction Paper (different colors)
- Origami Paper (different colors)
- Photo mounting Board paper (different colors)
- A4 paper
- Felt pens
- Pencils
- Ruler

Tools:

- Scissors
- Exacto-knife
- Glue stick and liquid
- Paper clips
- Adhesive tape

B.2.5 Pre-Study questionnaire

The participants will be asked to fill this form before being given the task.

Aim: To get a rough idea about the participants involvement in DIY culture, previous DIY experience and experience authoring a DIY tutorial. If so, have they published them online.

1. Personal Information: Current occupation/Field of study

2. How long have you been in involved in DIY in terms of years? (numeric)

3. How often do they make a DIY project? (a choice selection between):

    - Almost never
    - Once a week
    - Once a month
    - Every Couple of Months
    - Once a year

4. How many tutorials have your written and published so far (estimate)? (a choice selection between)

    - None
    - Atleast one
    - Atleast five
    - Atleast 10
    - Atleast 25
    - Atleast 50
    - More than 50

B.2.6 Post Study Semi Structured Interview

The participants after the completion of their task and writing the DIY tutorial will then be given a 10-15 minutes semi structured interview.

1. How does your general experience about the writing process compare to using DIYMate system for writing a DIY tutorial?
2. How did DIYMate support your workflow in terms of ideation, outlining or planning the structure of the tutorial?
3. How did DIYMate support your workflow in terms writing the DIY tutorial?
4. How did DIYMate support your workflow of authoring the DIY tutorial in terms for getting feedback and reviews on your DIY Tutorial?
5. What parts were frustrating for you when using DIYMate ?
6. What parts were enjoyable for you when using DIYMate and what controls did you enjoy using?

B.2.7 Data Collected

Data collected from the participant.

1. Screen and audio recordings when using the system.
2. Audio for the semi structured interview.
3. Logs of what operations were used and which buttons were clicked when using the system.
4. Number of words in the final DIY tutorial and number of words inserted into the DIY tutorial.
5. DIY tutorial and the initial outline

6. Chat history.
7. Review history.

B.2.8 Experimental Setup

1. All the required tools and materials will be made available on a table. (list of materials and tools per project is provided later)
2. Another table with laptop hosting DIYmate, a monitor, HDMI cable, keyboard and mouse. The monitor would be in an mirrored display configuration.
3. Laptop with obs for audio and video recording using obs.
4. Phone for audio recording.
5. An additional phone to allow participants to take pictures of their DIY project for their tutorial.
6. Sheet of paper mentioning the task with the specific knowhow and knowledge, but can also search the internet for it.
7. Additional paper for writing/scribbling notes.

B.2.9 Experimental Procedure

Before the arrival of the participant.

- Setup the Experiment based on Experimental setup.
- Test if the audio and video recording are working
- Prepare drinks and snacks for participants
- Setup the consent form
- Have a pen ready

After the arrival of the participant

- Greet them and thank them for taking the time to participate in the study.
- Introduce the participant to the purpose of the study
- Request the participant to read and sign the consent form. Explain the contents if necessary. Ask for their consent in recording their audio and their screen recording.
- Tell them that they are allowed to take a break whenever they want to.
- Ask them if they are feeling healthy and well to participate in the study.
- Explain that the study will be as follows:
  1. A pre-task questionnaire in the app itself.
  2. then they will be presented with a DIY task, and also allowed access to DIY mate where they can generate the initial outline and then continue.
  3. Post that they will be requested to finish the DIY task and write down the DIY Tutorial.
  4. we will then end the study with a semi-structured interview regarding their experience using the system.
- Ask the participant if they have any questions.
- Hand them the phone to allow them to take pictures.
- Explain how to upload the pictures to the DIY tutorial.
- Tell the participant that while doing the crafting process that they can ask for help if needed and look for specific things online but not explicitly search for DIY tutorials.
- Setup the DIYmate app in the monitor, along with the keyboard and mouse.
- Ask if they are ready to start.
- Tell them that their audio and screen recording has started.

- Ask them to fill out the questionnaire
- Present them the task once they are on the outline screen
- Ask them to generate the outline based on task.
- Once they are on the editor screen. Ask them to finish the task and work on the tutorial side by side.
- Once they feel that they are finished with the DIY task and tutorial.
- Ask them to click on the finish tutorial link.

B.2.10 After the Task

- export their logging data and tutorial
- Start recording the audio for the semi-structured interview make them aware that their audio is being recorded.
- Then conduct the 15 minute semi-structured interview regarding their experience using the system.
- Stop recording the audio and screen capture. ## End of Study:
- Thank the participant for their time and effort.
- Offer them snacks and drinks.
- Ask if they have any other questions and comments.

B.2.11 Participants

The target maker community has people from diverse backgrounds. for this study around 13 participants would be needed. They will be recruited from the university campus.

B.2.12 Experimental Design

Within groups design:

- All participants are given the same open ended DIY project to craft an animal
- Each participant will complete one DIY project and write one DIY tutorial about it.

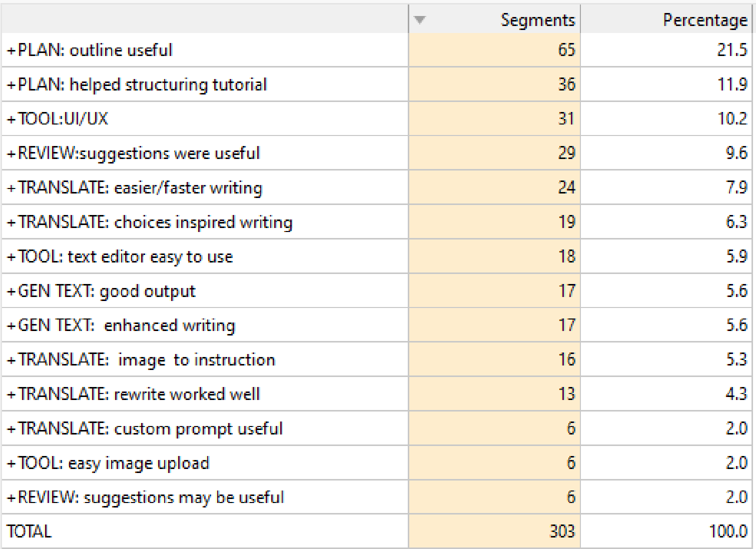
Data collected:

1. One Pre-task questionnaire regarding their experience in DIY and writing DIY tutorials.
2. Screen and audio recordings when using the system.
3. One Post-task semi-structured interview regarding their experience in using the system. We will collect an audio recording for this and transcribe it.
4. Quantitative data
   (a) Regarding which operations were used. What instructions were given to start the operation.
   (b) Chat history.
   (c) Reviews history.
   (d) How many generated words were inserted, total amount of words for the tutorial.
   (e) Initial outline generated
   (f) Final DIY tutorial
5. Estimated Duration: 90-120 minutes.

Appendix C

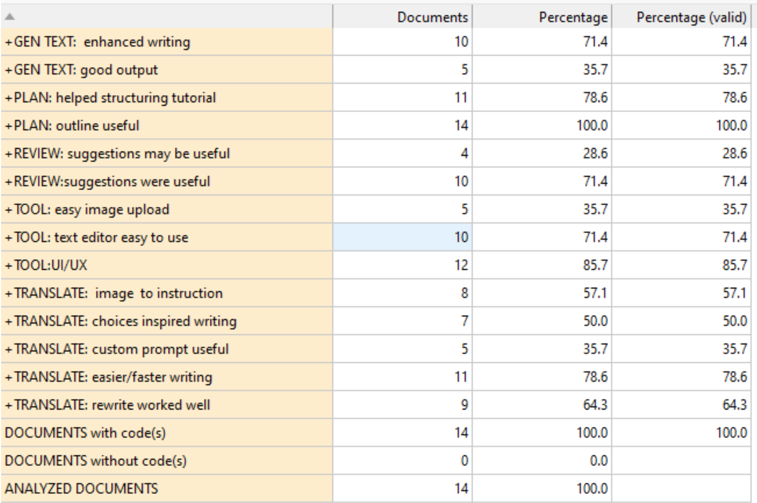
Appendix C

C.1 Codes from Evaluation Coding



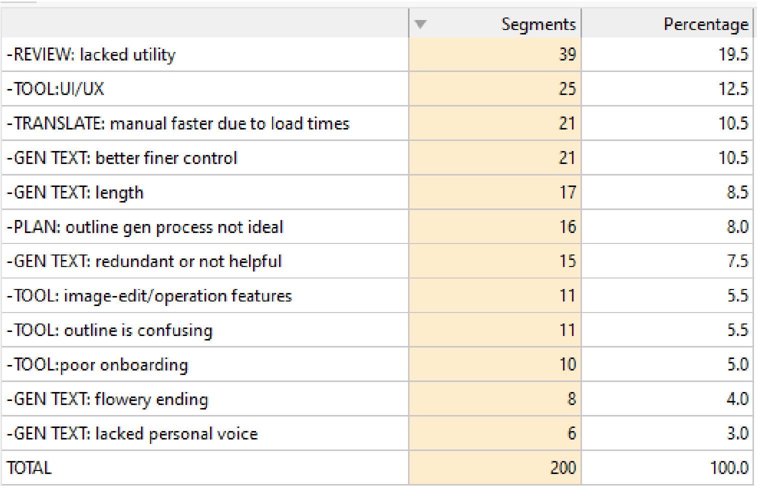
	Segments	Percentage
+PLAN: outline useful	65	21.5
+PLAN: helped structuring tutorial	36	11.9
+TOOL:UI/UX	31	10.2
+REVIEW:suggestions were useful	29	9.6
+TRANSLATE: easier/faster writing	24	7.9
+TRANSLATE: choices inspired writing	19	6.3
+TOOL: text editor easy to use	18	5.9
+GEN TEXT: good output	17	5.6
+GEN TEXT: enhanced writing	17	5.6
+TRANSLATE: image to instruction	16	5.3
+TRANSLATE: rewrite worked well	13	4.3
+TRANSLATE: custom prompt useful	6	2.0
+TOOL: easy image upload	6	2.0
+REVIEW: suggestions may be useful	6	2.0
TOTAL	303	100.0

Figure C.1: Table showing the segments per positive code



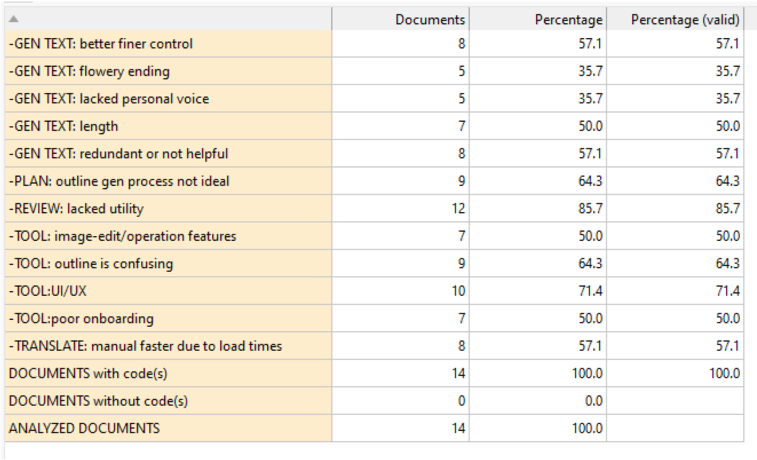
	Documents	Percentage	Percentage (valid)
+GEN TEXT: enhanced writing	10	71.4	71.4
+GEN TEXT: good output	5	35.7	35.7
+PLAN: helped structuring tutorial	11	78.6	78.6
+PLAN: outline useful	14	100.0	100.0
+REVIEW: suggestions may be useful	4	28.6	28.6
+REVIEW:suggestions were useful	10	71.4	71.4
+TOOL: easy image upload	5	35.7	35.7
+TOOL: text editor easy to use	10	71.4	71.4
+TOOL:UI/UX	12	85.7	85.7
+TRANSLATE: image to instruction	8	57.1	57.1
+TRANSLATE: choices inspired writing	7	50.0	50.0
+TRANSLATE: custom prompt useful	5	35.7	35.7
+TRANSLATE: easier/faster writing	11	78.6	78.6
+TRANSLATE: rewrite worked well	9	64.3	64.3
DOCUMENTS with code(s)	14	100.0	100.0
DOCUMENTS without code(s)	0	0.0	
ANALYZED DOCUMENTS	14	100.0	

Figure C.2: Table showing Participants per positive code



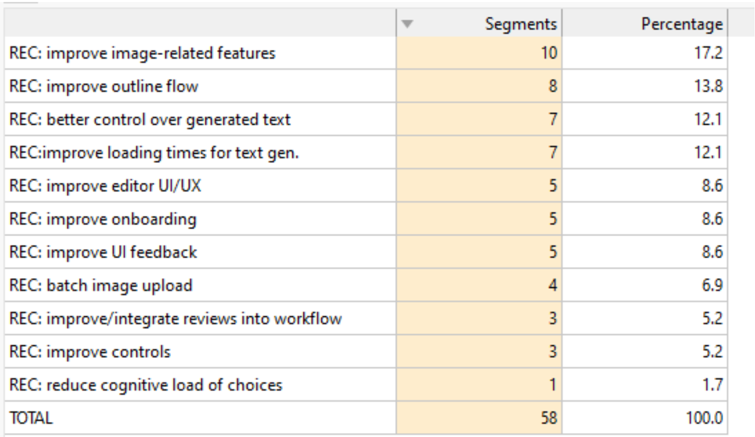
	Segments	Percentage
-REVIEW: lacked utility	39	19.5
-TOOL:UI/UX	25	12.5
-TRANSLATE: manual faster due to load times	21	10.5
-GEN TEXT: better finer control	21	10.5
-GEN TEXT: length	17	8.5
-PLAN: outline gen process not ideal	16	8.0
-GEN TEXT: redundant or not helpful	15	7.5
-TOOL: image-edit/operation features	11	5.5
-TOOL: outline is confusing	11	5.5
-TOOL:poor onboarding	10	5.0
-GEN TEXT: flowery ending	8	4.0
-GEN TEXT: lacked personal voice	6	3.0
TOTAL	200	100.0

Figure C.3: Table showing the segments per negative code



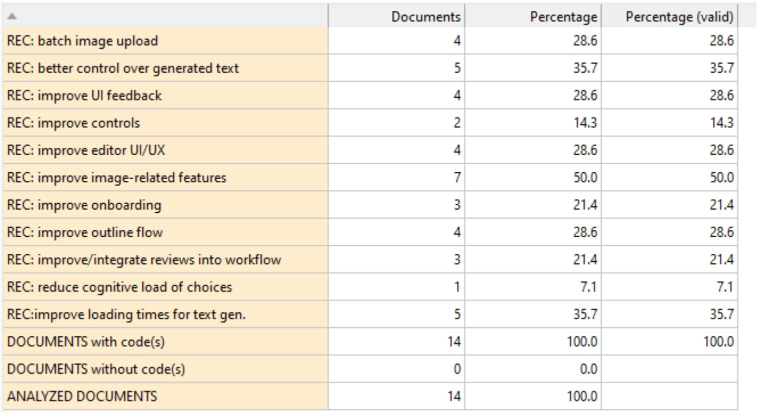
	Documents	Percentage	Percentage (valid)
-GEN TEXT: better finer control	8	57.1	57.1
-GEN TEXT: flowery ending	5	35.7	35.7
-GEN TEXT: lacked personal voice	5	35.7	35.7
-GEN TEXT: length	7	50.0	50.0
-GEN TEXT: redundant or not helpful	8	57.1	57.1
-PLAN: outline gen process not ideal	9	64.3	64.3
-REVIEW: lacked utility	12	85.7	85.7
-TOOL: image-edit/operation features	7	50.0	50.0
-TOOL: outline is confusing	9	64.3	64.3
-TOOL:UI/UX	10	71.4	71.4
-TOOL:poor onboarding	7	50.0	50.0
-TRANSLATE: manual faster due to load times	8	57.1	57.1
DOCUMENTS with code(s)	14	100.0	100.0
DOCUMENTS without code(s)	0	0.0	
ANALYZED DOCUMENTS	14	100.0	

Figure C.4: Table showing the participants per negative code



	Segments	Percentage
REC: improve image-related features	10	17.2
REC: improve outline flow	8	13.8
REC: better control over generated text	7	12.1
REC:improve loading times for text gen.	7	12.1
REC: improve editor UI/UX	5	8.6
REC: improve onboarding	5	8.6
REC: improve UI feedback	5	8.6
REC: batch image upload	4	6.9
REC: improve/integrate reviews into workflow	3	5.2
REC: improve controls	3	5.2
REC: reduce cognitive load of choices	1	1.7
TOTAL	58	100.0

Figure C.5: Table showing the segments per recommendation code



	Documents	Percentage	Percentage (valid)
REC: batch image upload	4	28.6	28.6
REC: better control over generated text	5	35.7	35.7
REC: improve UI feedback	4	28.6	28.6
REC: improve controls	2	14.3	14.3
REC: improve editor UI/UX	4	28.6	28.6
REC: improve image-related features	7	50.0	50.0
REC: improve onboarding	3	21.4	21.4
REC: improve outline flow	4	28.6	28.6
REC: improve/integrate reviews into workflow	3	21.4	21.4
REC: reduce cognitive load of choices	1	7.1	7.1
REC:improve loading times for text gen.	5	35.7	35.7
DOCUMENTS with code(s)	14	100.0	100.0
DOCUMENTS without code(s)	0	0.0	
ANALYZED DOCUMENTS	14	100.0	

Figure C.6: Table showing the participants per recommendation code

Bibliography

Yana Boeva and Peter Troxler. Makers. In *The Handbook of Peer Production*, chapter 17, pages 225–237. John Wiley & Sons, Ltd, 2020. ISBN 978-1-119-53715-1. doi: 10.1002/9781119537151.ch17. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119537151.ch17.

Leah Buechley, Daniela K. Rosner, Eric Paulos, and Amanda Williams. DIY for CHI: Methods, communities, and values of reuse and customization. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 4823–4826, New York, NY, USA, April 2009. Association for Computing Machinery. ISBN 978-1-60558-247-4. doi: 10.1145/1520340.1520750. https://dl.acm.org/doi/10.1145/1520340.1520750.

Alex Calderwood, Vivian Qiu, Katy Ilonka Gero, and Lydia B Chilton. How novelists use generative language models: An exploratory user study. 2020.

Matthew A. Dalton, Audrey Desjardins, and Ron Wakkary. From diy tutorials to diy recipes. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, page 1405–1410, New York, NY, USA, April 2014. Association for Computing Machinery. ISBN 978-1-4503-2474-8. doi: 10.1145/2559206.2581238. URL https://dl.acm.org/doi/10.1145/2559206.2581238.

Audrey Desjardins and Ron Wakkary. Manifestations of everyday design: Guiding goals and motivations. In *Proceedings of the 9th ACM Conference on Creativity & Cognition*, C&amp;C '13, pages 253–262, New York, NY, USA, June 2013. Association for Computing Machinery. ISBN 978-1-4503-2150-1. doi: 10.

1145/2466627.2466643. https://dl.acm.org/doi/10.1145/2466627.2466643.

Linda Flower and John R. Hayes. A Cognitive Process Theory of Writing. *College Composition and Communication*, 32 (4):365–387, 1981. ISSN 0010-096X. doi: 10.2307/356600. https://www.jstor.org/stable/356600.

Katy Gero, Alex Calderwood, Charlotte Li, and Lydia Chilton. A design space for writing support tools using a cognitive process model of writing. In Ting-Hao "Kenneth" Huang, Vipul Raheja, Dongyeop Kang, John Joon Young Chung, Daniel Gissin, Mina Lee, and Katy Ilonka Gero, editors, *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, page 11–24, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.in2writing-1.2. URL https://aclanthology.org/2022.in2writing-1.2.

Frederic Gmeiner and Nur Yildirim. Dimensions for Designing LLM-based Writing Support. In *In2Writing Workshop*, 2023.

John R. Hayes. Modeling and Remodeling Writing. *Written Communication*, 29(3):369–388, July 2012. ISSN 0741-0883. doi: 10.1177/0741088312451260. https://doi.org/10.1177/0741088312451260.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models. http://arxiv.org/abs/2001.08361, January 2020.

Stacey Kuznetsov and Eric Paulos. Rise of the expert amateur: Diy projects, communities, and cultures. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, page 295–304, New York, NY, USA, October 2010. Association for Computing Machinery. ISBN 978-1-60558-934-3. doi: 10.1145/1868914.1868950. URL https://dl.acm.org/doi/10.1145/1868914.1868950.

Marcel Lahaye, Vivian Isabel Reinartz, Sarah Sahabi, and Jan Borchers. Towards authoring tools for diy tutorials: From tutorial user strategies to guidelines (free template included!). In *Proceedings of Mensch und Computer 2023*, MuC '23, page 380–386, New York, NY, USA, September 2023. Association for Computing Machinery. ISBN 9798400707711. doi: 10.1145/3603555.3608530. URL https://dl.acm.org/doi/10.1145/3603555.3608530.

Mina Lee, Percy Liang, and Qian Yang. 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*, CHI '22, pages 1–19, New York, NY, USA, April 2022. Association for Computing Machinery. ISBN 978-1-4503-9157-3. doi: 10.1145/3491102.3502030. https://dl.acm.org/doi/10.1145/3491102.3502030.

Florian Lehmann, Niklas Markert, Hai Dang, and Daniel Buschek. Suggestion Lists vs. Continuous Generation: Interaction Design for Writing with Generative Models on Mobile Devices Affect Text Length, Wording and Perceived Authorship. In *Mensch Und Computer 2022*, pages 192–208, Darmstadt Germany, September 2022. ACM. ISBN 978-1-4503-9690-5. doi: 10.1145/3543758.3543947. https://dl.acm.org/doi/10.1145/3543758.3543947.

Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. 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*, CHI '23, pages 1–34, New York, NY, USA, April 2023. Association for Computing Machinery. ISBN 978-1-4503-9421-5. doi: 10.1145/3544548.3581225. https://dl.acm.org/doi/10.1145/3544548.3581225.

Robert Phillips, sarah silve, and Sharon Baurley. The practical maker: Investigating the definitions and requirements of and exploring the motivations behind bespoke making. In *Crafting the Future*, Gothernburg, April 2013. https://researchonline.rca.ac.uk/1863/.

Laria Reynolds and Kyle McDonell. Prompt Programming for Large Language Models: Beyond the Few-Shot

Paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, pages 1–7, New York, NY, USA, May 2021. Association for Computing Machinery. ISBN 978-1-4503-8095-9. doi: 10.1145/3411763.3451760. https://dl.acm.org/doi/10.1145/3411763.3451760.

Johnny Saldaña. *The Coding Manual for Qualitative Researchers*. SAGE, Los Angeles, 2nd ed edition, 2013. ISBN 978-1-4462-4736-5 978-1-4462-4737-2.

Tiffany Tseng. Build in Progress: Building Process-Oriented Documentation. In *Makeology*. Routledge, 2016. ISBN 978-1-315-72651-9.

Tiffany Tseng and Mitchel Resnick. Product versus process: Representing and appropriating DIY projects online. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS '14, pages 425–428, New York, NY, USA, June 2014. Association for Computing Machinery. ISBN 978-1-4503-2902-6. doi: 10.1145/2598510.2598540. https://dl.acm.org/doi/10.1145/2598510.2598540.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. http://arxiv.org/abs/1706.03762, August 2023.

Ron Wakkary, Markus Lorenz Schilling, Matthew A. Dalton, Sabrina Hauser, Audrey Desjardins, Xiao Zhang, and Henry W.J. Lin. Tutorial authorship and hybrid designers: The joy (and frustration) of diy tutorials. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 609–618, New York, NY, USA, April 2015. Association for Computing Machinery. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702550. URL https://dl.acm.org/doi/10.1145/2702123.2702550.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent Abilities of Large

Language Models. http://arxiv.org/abs/2206.07682, October 2022.

Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: Story writing with large language models. In *27th International Conference on Intelligent User Interfaces*, IUI '22, page 841–852, New York, NY, USA, March 2022. Association for Computing Machinery. ISBN 978-1-4503-9144-3. doi: 10.1145/3490099.3511105. URL https://dl.acm.org/doi/10.1145/3490099.3511105.

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

Index

Typeset May 24, 2024