國立臺灣大學電機資訊學院資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia
College of Electrical Engineering and Computer Science
National Taiwan University

Master's Thesis

DefExtractor: 基於大語言模型雙向互動的數學標識符-定義對的自動提取與視覺化

DefExtractor: LLM-Based Automatic Extraction and Visualization of Mathematical Identifier-Definition Pairs with Bidirectional Interaction

李旻叡
Min-Jui Lee

指導教授：陳炳宇 博士
Advisor: Bing-Yu Chen, Ph.D.

中華民國 113 年 7 月
July, 2024

doi:10.6342/NTU202403869

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

DefExtractor: 基於大語言模型雙向互動的數學標識符-定義對的自動提取與視覺化

DefExtractor: LLM-Based Automatic Extraction and Visualization of Mathematical Identifier-Definition Pairs with Bidirectional Interaction

本論文係李旻叡（學號 R11944018）在國立臺灣大學資訊網路與多媒體研究所完成之碩士學位論文，於民國 113 年 7 月 11 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Graduate Institute of Networking and Multimedia on 11 July 2024 have examined a Master's Thesis entitled above presented by LEE, MIN-JUI (student ID: R11944018) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

	林文杰	張鈞法
（指導教授 Advisor）		

系（所）主管 Director: 鄭卜壬

致謝

這整篇研究的誕生需要感謝許多的人。有了大家的幫助，我才能一步步完成這篇碩士論文。

首先我要感謝我的指導教授，陳炳宇教授，在整段研究過程中給予我充分的自由並提供專業的建議，讓我最初模糊而發散的構想，漸漸收斂並轉變成實際的研究。再來，我要特別感謝紀佩妤教授，爲我的實驗設計提供許多寶貴建議，並在論文撰寫時給予我莫大的幫助與鼓勵，讓我能夠整理出詳細的研究脈絡，並清晰的表達研究的核心主題。接著，我也要感謝研究夥伴曾昱婷，收集了研究中我不太熟悉的顏色領域資訊，幫助我在工具的開發上有了更明確的設計依據，並協助我整理人機協作領域的過往研究，爲論文中相關論述提供了基礎。

此外，感謝實驗室的學長姊、同屆夥伴和學弟妹的陪伴與支持，特別是王秋玄學姐，爲我的實驗問卷設計提供寶貴建議，以及黃靖聘學妹，協助我設計工具介面，讓使用體驗有了極大改進。最後，我也要感謝所有實驗受測者的參與和回饋，幫助我一步步改善工具，感謝實驗室提供設備和資源，讓我能夠順暢的進行工具的前後端開發。

回顧這兩年的碩士班研究生生活，從最初幫助學姊們的研究、中間接了產學合作的專案，到最後發想並實踐自己的研究，期間雖然遇上了許多困難與挫折，但有了教授們與實驗室大家的支持與幫助，我才能克服難關並收穫無數寶貴的經驗。在此，我要向所有幫助過我的人們獻上最誠摯感謝。

doi:10.6342/NTU202403869

中文摘要

讀懂數學式是閱讀科學文章時的必要任務之一。閱讀數學式時，讀者需要找出其中各個符號的定義，而良好的視覺化與結構化的文字解釋，能夠增強這些數學式可讀性。然而，這些設計需要仰賴文章作者的編輯，過程不僅繁瑣且費時。爲此，我們製作了DefExtractor，一個基於大語言模型的編輯工具，能夠幫助文章作者提取並視覺化數學標識符與對應的定義。只要給定由LaTeX編輯的數學式與解釋性文字，DefExtractor便能利用大語言模型辨識其語意，推薦適合的上色設計。對於模型的推薦設計，使用者還能夠進一步提出改正建議讓模型改正，以此達到人與AI模型的雙向互動與協作。我們進行了技術評估，發現DefExtractor的標識符-定義對自動提取流程，在目標使用情境下優於過去的模型。而一項包含十二位受試者的使用者研究，顯示DefExtractor中AI的輔助，能夠有效降低使用者負荷並縮短編輯時間。

關鍵字: 數學式, 數學符號, 編輯工具, 視覺化, 大語言模型, LaTeX

doi:10.6342/NTU202403869

Abstract

Following mathematical formulas is a critical task in scientific paper reading. Readers would trace the definition and the relation of identifiers in a formula. To enhance the readability, it relies on paper authors to create effective visualization and structured text explanations, which is especially challenging for long formulas. We propose DefExtractor, an LLM-based tool that assists authors in extracting and visualizing identifier-definition pairs with AI interactively. Given a LaTeX input, DefExtractor identifies the semantics and automatically suggests colored identifiers and definitions based on the LLM response. Users can modify via text prompt or syntax, where AI adapts the edits iteratively. A technical evaluation showed our pair extraction pipeline outperforms previous model in our target scenario, and a usability study with 12 participants showed that DefExtractor effectively reduced the workloads of authors and shortened editing time compared with a baseline tool.

Keywords: Mathematical formulas, mathematical notation, authoring tools, visualization, LLM, LaTeX

doi:10.6342/NTU202403869

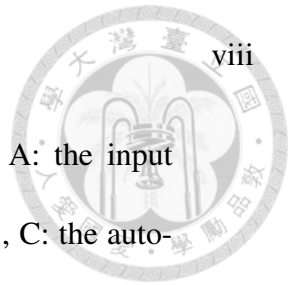
Table of Contents

doi:10.6342/NTU202403869

doi:10.6342/NTU202403869

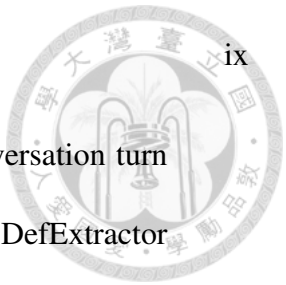
List of Figures

doi:10.6342/NTU202403869



[1] https://twitter.com/andrew_n_carr/status/1346172166077726720

doi:10.6342/NTU202403869



doi:10.6342/NTU202403869

List of Tables

doi:10.6342/NTU202403869

Chapter 1

Introduction

Mathematical formulas convey complex concepts in a concise manner. They are widely used as a critical foundation of physics, computer science, and other fields [22]. Formulas are composed of identifiers, each carrying a corresponding meaning with a symbol. It requires reader to carefully trace each identifier and reason their relations under an overall structure, which can be a mentally challenging task [1, 37]. To well illustrate a formula, scientific authors often provide extensive explanatory prose to define the identifiers [45]. The readers have to therefore shift their focus between these formulas and the explanatory prose frequently [30].

To enhance the readability, researchers and educators have utilized visual designs to highlight the definitions of identifiers. For example, colorization is a simple yet effective method. By applying the same color to the identifiers in a formula and their corresponding definitions in the explanatory prose, this technique establishes visual links to guide readers' attention [22, 5]. For another example, bullet points are a common technique, often used in presentation slides or educational materials. Authors can list the definitions

doi:10.6342/NTU202403869

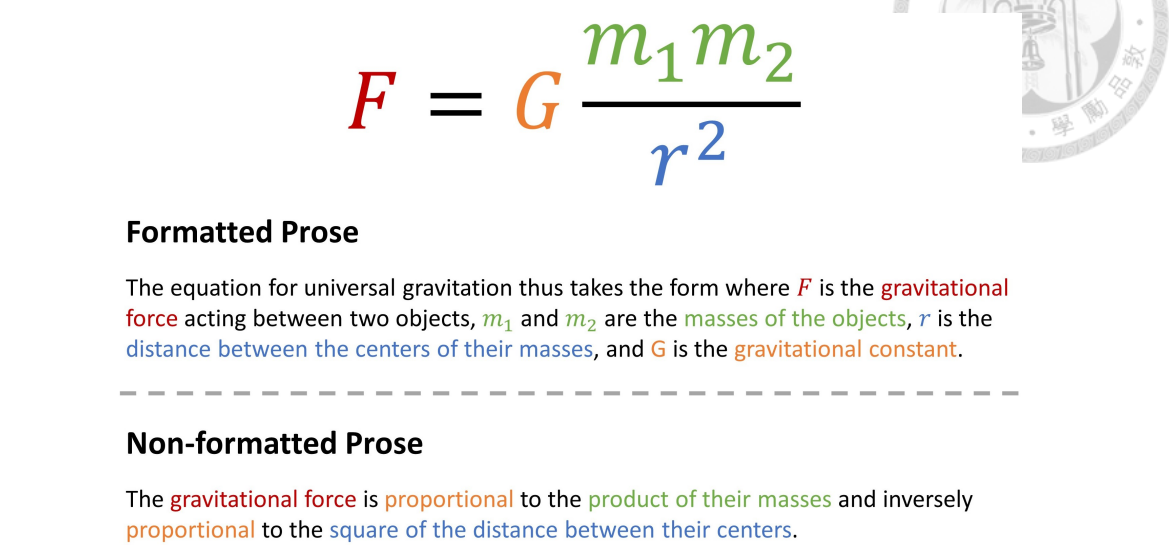


Figure 1.1: *Examples of formatted prose and non-formatted prose explaining the formula for universal gravitation.*

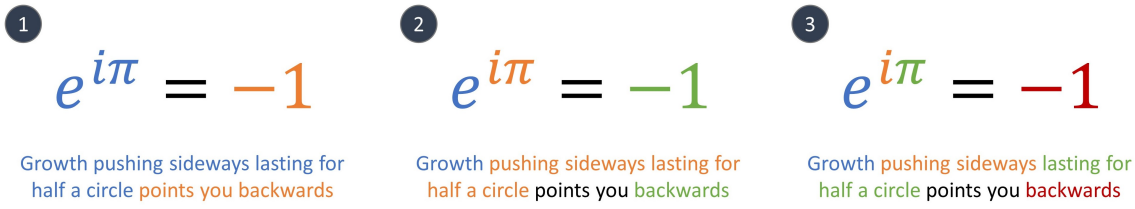


Figure 1.2: *Three different colorization designs for the Euler's identity.*

of identifiers in a structured manner with bullet points, which makes it convenient for readers to compare and comprehend [11, 36].

Currently, to implement these designs in available tools, authors have to manually extract the definitions, followed by typesetting or colorization [22]. With advancements in machine learning, AI introduces the capability to perform such extractions [35]. While prior work has proposed AI-based approaches, many can only handle structured texts [39, 27, 31].

To understand the problem, we collected 18 instances from previous literature [22, 26] that fit our application scenarios and found two types of explanatory prose: formatted and

doi:10.6342/NTU202403869

non-formatted prose. As shown in Figure 1.1, formatted prose provides explanations corresponding to each identifier, while non-formatted prose, on the other hand, provides only a holistic description of the entire formula. The unstructured nature of non-formatted prose requires models to have a deeper understanding of formulas [27]. We utilized the Lee and Na's model [33] to create a pair extraction pipeline, and measured its performance using the collected cases. We found that it performed significantly better on formatted examples compared to non-formatted ones.

Furthermore, the previous methods do not take into account the subjectivity involved in the designs. As shown in Figure 1.2, even for the same formula, there are different designs to visualize identifier-definition pairs, and the decision highly depends on the human author. Previous models, however, have predetermined strategies and cannot be adjusted based on users' needs [35].

From these issues, we observed that it is critical to include human authors in the editing process. In this work, we propose DefExtractor, an authoring tool that enables users to inspect, revise, and interact with AI for creating visual formulas.

Figure 1.3 shows our Human-AI collaboration workflow: With formula and prose in the LaTeX format, LLM extracts a preliminary set of identifier-definition pairs (see Figure 1.3A). Users can provide feedback to LLM or directly make corrections (Figure 1.3B). Next, users can refine elements within each pair. Users can select identifiers to prompt LLM to extract corresponding definitions, and vice versa (Figure 1.3C). Finally, with the extracted pairs, DefExtractor offers colorization and bullet points designs (Figure 1.3D). DefExtractor outputs the results as a piece of LaTeX code that can be further edited or

doi:10.6342/NTU202403869

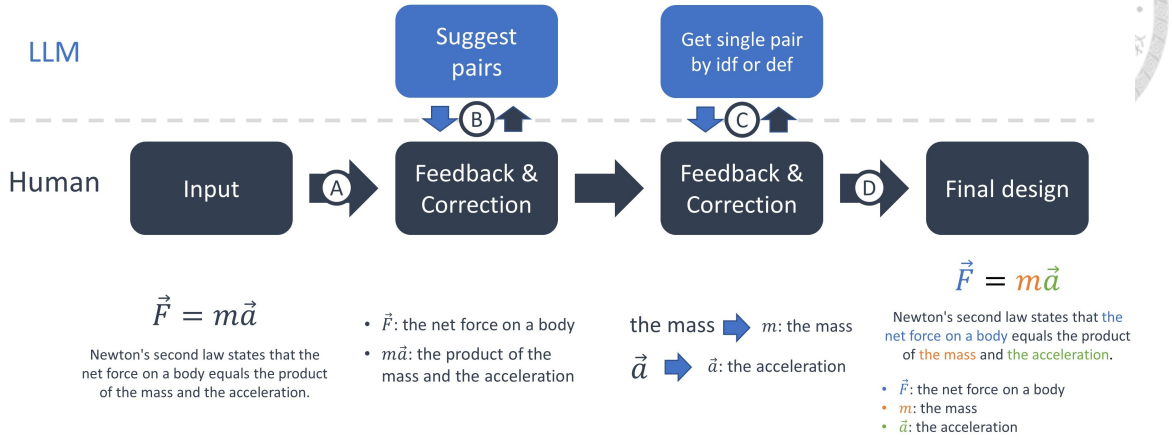


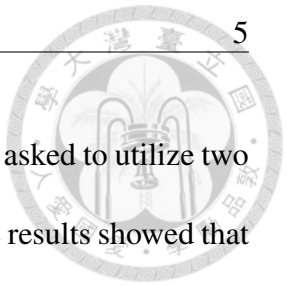
Figure 1.3: *DefExtractor is a human-AI co-editing tool for creating color-coded identifier-definition pairs of mathematical formulas. Given a formula and its prose, LLM extracts a set of identifier-definition pairs (A). Users can provide feedback to the model to adjust the overall pairs (B). Otherwise, they can select identifiers to prompt the model to extract corresponding definitions, and vice versa (C). Finally, DefExtractor offers two design options, colorization and bullet points to enhance readability (D).*

copied in a common LaTeX tool.

In this end-to-end workflow, DefExtractor serves as an intermediary, conveying user needs and feedback through a well-designed prompt structure to LLM, and then visualizing LLM's responses into graphics that are understandable to human users. DefExtractor integrates an interactive interface and bidirectional interaction with LLM, in order to facilitate users in enhancing the readability of mathematical formulas in an iterative process toward the goal.

To verify the effectiveness of DefExtractor, we conducted a technical evaluation and a user study with 12 participants. In the technical evaluation, the pair extraction pipeline used by DefExtractor outperforms the one based on Lee and Na's model. Moreover, DefExtractor achieves performance on non-formatted examples that is close to its perfor-

doi:10.6342/NTU202403869



mance on formatted examples. In the user study, the participants were asked to utilize two versions of DefExtractor, one with AI assistance and one without. The results showed that AI assistance can effectively reduce completion time and the workload from authors. Furthermore, all participants agreed that they would like to use DefExtractor for extracting and visualizing identifier-definition pairs. Specifically, this research makes four contributions.

- A human-AI collaboration workflow for identifier-definition extraction and visualization of mathematical formulas.
- An interactive authoring tool, DefExtractor, with a real-time LLM to support iterative refinements of identifier-definition pairs.
- An end-to-end pipeline that converts a raw formula input to effective prompts for live LLM feedback that enables iterative user edits.
- A user study with 12 participants to evaluate how DefExtractor facilitates the authoring process.

doi:10.6342/NTU202403869

Chapter 2

Related Work

We built our research upon three fundamental areas, including math augmentation, identifier-definition extraction, and human-AI collaboration.

2.1 Reading Augmentation in Math

It can be a multifaceted process to follow mathematics. It requires readers integrate their understanding of numbers, words, and identifiers [1]. The comprehension of identifiers is particularly challenging, as their meanings within mathematical formulas often rely on the explanatory prose [45]. Consequently, during the reading process, readers must frequently shift their focus between formulas and explanatory prose, placing a certain burden on them. [30]

Making mathematical formulas more readable is an issue that has been studied for a long time [13]. There are various approaches, such as visual design [22, 49], interactive articles [24, 25, 10], reading interface [21, 2], and even augmented reality technology [9], and their subjects have high diversity. e-Proofs [2] fade out parts of the text that are not

doi:10.6342/NTU202403869

currently the focus, thereby guiding readers attention. Idyll [10] assists authors in designing interactive articles, enabling readers to understand the impact of various parameters by changing their numerical values within formulas.

Some researches attempt to visualize the relationship between identifiers and definitions. ScholarPhi [21] dynamically presents the definition next to formulas using interactive tooltips and equation diagrams. Head et al.'s research [22] systemized various visual designs capable of enhancing readability, among which colorization is the most pervasive, which establishes visual links between identifiers and definitions. Living Papers [24] is a language toolkit that can present the definitions through colorization and interactive tooltips. FFL [46] is a markup language for augmenting formulas, which supports colorization and label designs.

DefExtractor utilizes colorization and bullet points to visualize identifier-definition pairs. The use of color to guide reader attention is employed in many fields. SCIM [16] attempts to assign different colors to content from different topics in articles to aid in the skim process. Many programming editors such as VSCode[1] utilize color to help users distinguish different types of terms. On the other hand, bullet points are a classic method to organize information [11, 36]. They are concise and structured, helping readers quickly grasp key points, and their brevity allows them to be utilized in situations with limited space, such as slides.

[1] https://code.visualstudio.com/

doi:10.6342/NTU202403869

2.2 Identifier-Definition Extraction

Identifier-definition extraction is one of the representative tasks in mathematical language processing, aiming to match identifiers with their corresponding definitions [35]. This issue has no strict formatting requirements for the processed text, but it tends to resemble formatted prose [39, 27, 31]. Over the years, research has progressed from rule-based approaches [38, 3], statistical language models [48, 39], to the current mainstream neural network language models [15, 33].

However, the unstructured nature of non-formatted prose is still a challenge. Jo et al. [27] mentioned that if identifiers do not appear in explanatory prose, predicting them must rely on the model's understanding of common symbol usage patterns across papers. Alexeeva et al. [3] acknowledged that their approach may result in errors when identifiers used in formulas is not the same those used in prose. In summary, to handle non-formatted prose, models need to be familiar with the usage patterns of symbols and have a deep understanding of formulas, rather than relying on the structure of the prose.

Large language models(LLMs) demonstrate exceptional abilities in understanding context and semantics [51], significantly altering how language models are utilized [44] and providing us with opportunities to handle non-formatted prose. Therefore, DefExtractor leverage prompt engineering [18, 34] to conduct identifier-definition extraction with LLMs. The subsequent technical evaluation and user study confirmed that the performance outperforms Lee and Na's model [33] in our scenario and is sufficient to meet user needs (section 6.2).

doi:10.6342/NTU202403869

2.3 Human-AI Collaboration

With the advancement of machine learning and AI, the concept of human-AI collaboration has become increasingly prevalent [41]. People are starting to use AI to assist in various tasks across different fields, such as healthcare [32] and data science [42].

In the field of academic reading and writing, people are beginning to use language models to assist with understanding text and generating content. Synergi [28] combines LLMs to provide a structured outline of research threads, assisting readers in literature review. Sparks [17] use AI for idea generation, aiding authors in science writing. Metaphorian [29] supports the creation of scientific metaphors with LLMs to reflect the authors' requirements. Slide4N [43] is a human-AI collaborative interface that helps users generate slides from computational notebooks.

In these tools, AI plays a collaborative role with the users, providing assistance and helping users iterate based on their needs rather than solely relying on AI-generated results. It adjusts and iterate according to the user's requirements. In the human-AI interaction guidelines proposed by Amershi et al. [4], the necessity of allowing users to correct AI errors is also emphasized.

DefExtractor adopts a multi-step collaborative workflow. After the AI suggests an initial set of pairs, users can switch to individual pair for detailed modifications. For each pair, users can specify a symbol for the AI to find the corresponding definition, or vice versa. Additionally, inspired by applications like chatbots [52], DefExtractor allows users to use natural language to suggest more complex modifications to the AI.

doi:10.6342/NTU202403869

Chapter 3

Design Process

During the design process of DefExtractor, we first created an early prototype called MaugVLink to preliminarily validate the rationale of our design direction and to gather user feedback. Based on this prototype and preliminary user study, we established several design goals to guide the subsequent design and development of DefExtractor.

3.1 Early prototype: MaugVLink

Compared to the subsequent DefExtractor, our early prototype, MaugVLink, features a simpler human-AI collaboration workflow. As shown in Figure 3.1, when users input formulas and prose, MaugVLink automatically suggests a set of identifier-definition pairs. If users are not satisfied with the suggestions, they can use the interactive interface to make modifications.

To create automatic suggestions, we used the model developed by Lee and Na. [33] since it had the best performance in Symlink [31], which is highly related to our use case. Symlink aims to extract pairs of mathematical symbols and their corresponding descrip-

doi:10.6342/NTU202403869

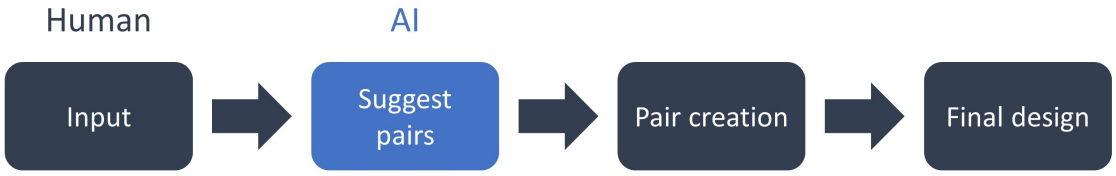


Figure 3.1: *The workflow of our early prototype (MaugVLink).*

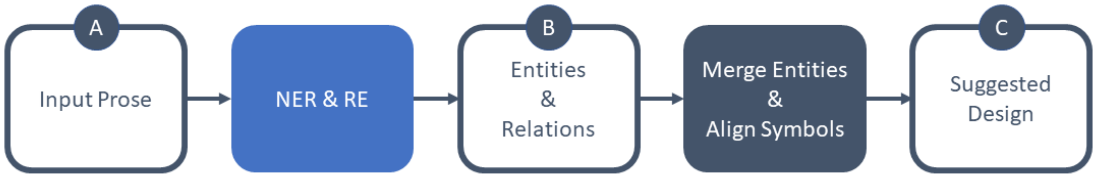


Figure 3.2: *Pair extraction pipeline in MaugVLink. The blue step is the Symlink [31] task, which is composed of two sub-tasks, Named Entity Recognition (NER) and Relation Extraction (RE). We use Lee et al.'s model [33] to achieve this task.*

tions from scientific documents. This involves two sub-tasks: Named Entity Recognition (NER) and Relation Extraction (RE). In NER task, the model will extract entities from the input prose. These entities include mathematical symbols and terminology descriptions, with three tags: SYMBOL, PRIMARY, and ORDERED, corresponding to the categories of mathematical symbols, standalone definitions, and descriptions of multiple terms, respectively. In RE task, the model will identify the relationships between these entities. The relations have four types: DIRECT, establishing a link between a symbol and its definition; COUNT, connecting a description with a symbol that represents the number of instances; COREF-SYMBOL, linking co-referred symbols; and COREF-DESCRIPTION, linking co-referred descriptions.

With Lee and Na.'s model, we can extract identifier-definition pairs automatically. Take Figure 3.3B as an example; the prose has two DIRECT relations, each with a SYMBOL and a PRIMARY entity. We will merge all the entities in the same relation and treat

doi:10.6342/NTU202403869

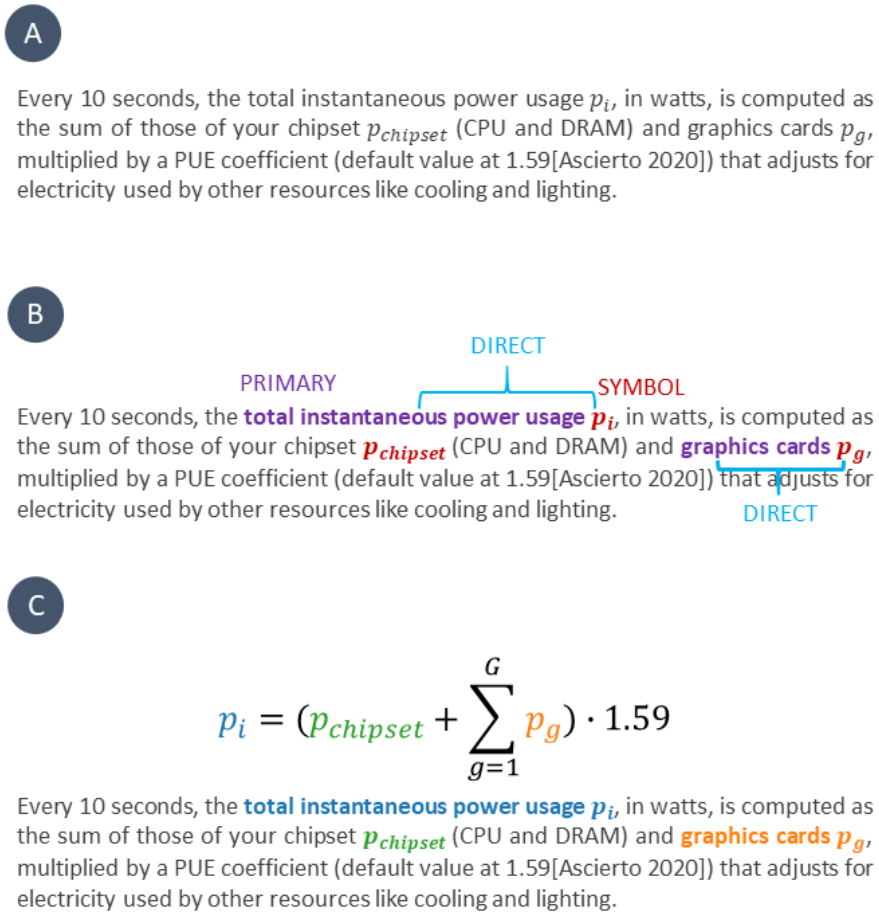
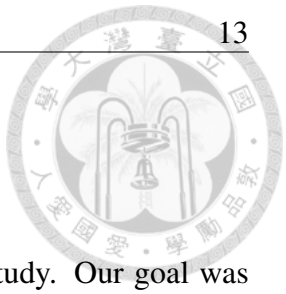


Figure 3.3: *An example of pair extraction pipeline in MaugVLink. A: the input prose form [40], B: the output of Lee and Na.'s model [33], C: the auto-suggested design by the MaugVLink.*

them as a single pair. Thus, "total instantaneous power usage" and "p_i" will be a pair, and "graphics cards" and "p_g" will be another. A DIRECT or COUNT relation has a symbol and a definition entity, so merging them is reasonable. As for COREF-SYMBOL and COREF-DESCRIPTION relations imply that multiple symbols or descriptions have a relation, so we will merge them as well. We can use a rule-based method to locate these selected symbols in prose in the formula. In the end, we will assign distinct colors to each pair, creating the auto-suggested design, as illustrated in Figure 3.3C.

doi:10.6342/NTU202403869



3.2 Preliminary User Study

To evaluate the MaugVLink, we conducted a preliminary user study. Our goal was to answer two research questions: First, is the design of MaugVLink for creating pairs reasonable, user-friendly, and easy to learn? (S1) Second, is MaugVLink robust enough and applicable to users' practical needs? (S2)

As a result, our study consisted of two parts. In the first part, we asked the participants to use MaugVLink to replicate a given colorization design (S1). In the second part, we had the participants provide their own target formulas and corresponding explanations and asked them to use MaugVLink for colorization (S2).

3.2.1 Methodology

3.2.1.1 Participants

We recruited 8 participants (4 women and 4 men, aged 22 to 29) via an internal lab mailing list. One was a PhD student; the others were MSc students. Most of them were beginners with LaTeX (6/8).

3.2.1.2 Procedure

We started the study with a short introduction and a tutorial on MaugVLink. Then, we had the participants practice using MaugVLink to complete two simple specified examples. Once they felt ready, we asked the participants to replicate a given colorization design shown in Figure 3.4. Then, the participants were asked to fill out a questionnaire. After that, we had the participants provide their own formulas and corresponding

doi:10.6342/NTU202403869

$$\cos\theta = \frac{< a, b >}{||a|| \cdot ||b||} = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i a_i}\sqrt{\sum_{i=1}^{n} b_i b_i}}$$

The angle between two vectors is calculated by finding the inner product between the first and second vectors and dividing by the length of each then take the arc cosine of that value.

Figure 3.4: *The colorization design participants had to replicate, which is from* [1].

explanations then use MaugVLink to colorize them. They were asked to fill out another questionnaire as well. After that, we interviewed the participants for a few questions.

3.2.1.3 Measures

We use the system usability scale (SUS) [7] to evaluate the usability of MaugVLink. In addition, we referred to the study conducted by Wang et al. [43] and utilized three questionnaires to evaluate MaugVLink's UI design, AI suggestion, and output quality on a 7-point Likert scale. Note that, in the first part of the study, the designs provided to the participants were predetermined by us. Therefore, the questionnaires for the first part only address the UI design and do not include evaluations of AI suggestions and output quality.

3.2.2 Results and Discussion

After each part of the study, we had the participants complete the SUS questionnaire individually. The average reached 76.25 (min = 60, max = 90, σ = 11.46) in the first part and 74.69 (min = 55, max = 97.5, σ = 15.68) in the second part. The scores in the second part are more dispersed, and the average is slightly lower. This variation is attributed to the diverse nature of cases in the second part. The SUS scores for both parts exceeded 70,

doi:10.6342/NTU202403869

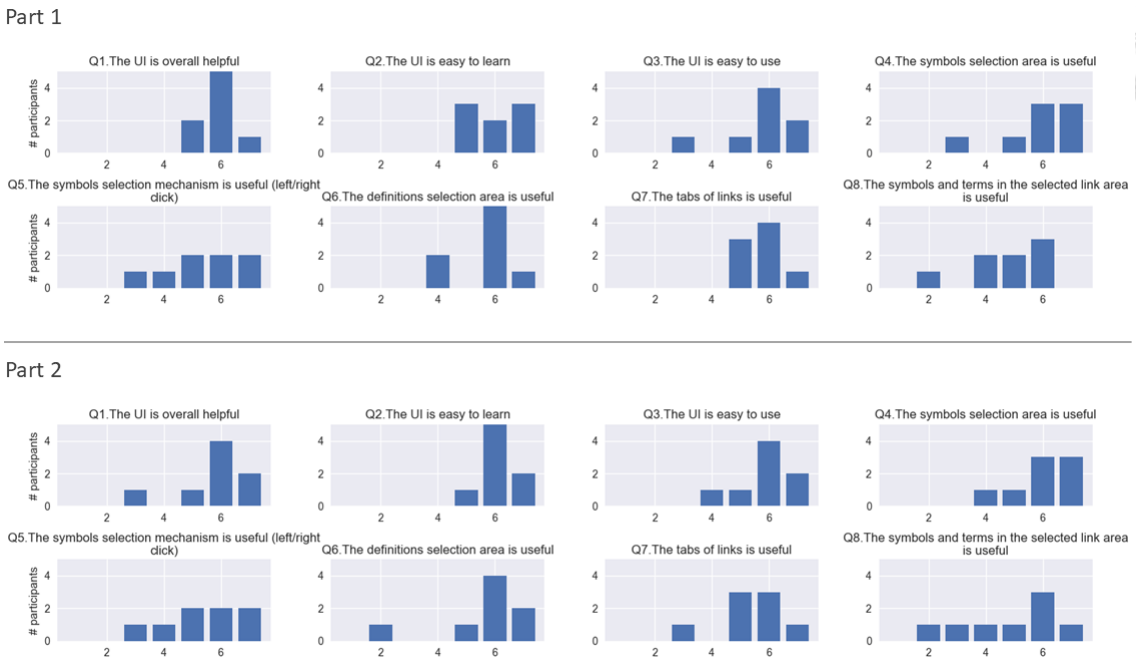


Figure 3.5: *Participants' ratings on UI design, on a 7-point Likert scale (1 = "strongly disagree" and 7 = "strongly agree").*

which is acceptable [6].

The participants' ratings for various UI elements are shown in Figure 3.5. There is no obvious difference in the scores between the two parts. Most participants scored positive for the overall UI design ($\mu_{Q_1,part1} = 5.88$, $\mu_{Q_1,part2} = 5.75$), expressing that they found the UI design is easy to learn ($\mu_{Q_2,part1} = 6.00$, $\mu_{Q_2,part2} = 6.13$) and easy to use ($\mu_{Q_3,part1} = 5.75$, $\mu_{Q_3,part2} = 5.88$). The UI element with highest rating is the symbols selection area ($\mu_{Q_4,part1} = 5.88$, $\mu_{Q_4,part2} = 6.00$), and the one with lowest rating is the symbols and terms in the selected pair ($\mu_{Q_8,part1} = 4.75$, $\mu_{Q_8,part2} = 4.88$). The average scores for each UI element are all above the median value of 4, indicating generally positive evaluations.

Ratings for AI suggestion and output quality were only conducted in the second part, and the results are shown in Figure 3.6. The ratings for AI suggestion were quite polar-

doi:10.6342/NTU202403869

Figure 3.6: *Participants' ratings on AI suggestion and output quality in the second part, on a 7-point Likert scale (1 = "strongly disagree" and 7 = "strongly agree").*

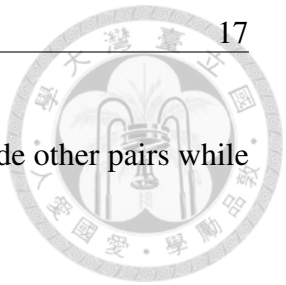
ized, reflecting the AI system's strong performance in some cases but weaker performance in others. As for the evaluation of output quality, most participants provided positive feedback ($\mu_{Q_{12}} = 5.75$). They agreed that the output results, while aesthetically pleasing ($\mu_{Q_{14}} = 6.13$), also enhanced the readability of the formulas ($\mu_{Q_{13}} = 6.25$).

3.3 Design Goals

Inspired by previous literature and the preliminary user study of early prototypes, we formulated the following four goals to guide the design and development of DefExtractor.

- **G1: Support pair-oriented editing** DefExtractor is designed for the extraction and visualization of identifier-definition pairs, so editing must be oriented around these pairs, allowing users to quickly add, delete, or hide them. In Head et al.'s study [22], authors using colorization expressed a desire for editing tools to help ensure consistency in the

doi:10.6342/NTU202403869



colors within a pair. In our preliminary user study, users want to hide other pairs while editing one pair to avoid distraction.

- **G2: Support designs for different layout.** Researchers often need to present their studies in various formats [24], such as papers, web articles, slides, and so on. However, these formats can differ significantly in the layouts. For instance, it's possible to use an entire paragraph to explain a formula in papers, whereas slides need to remain concise. Therefore, the output format and design of DefExtractor must be flexible to allow users to make adjustments accordingly.

- **G3: Suggest identifier-definition pairs automatically.** Extracting identifier-definition pairs is tedious and time-consuming, especially for non-formatted prose, as there is no fixed design pattern (Figure 1.2) and repeated manual adjustments are required. Automating this process with AI can alleviate the burden on authors [22]. DefExtractor should integrate AI to suggest identifier-definition pairs, facilitating editing and adjustments for users. The preliminary user study indicated that the pipeline based on Lee and Na.'s model could not handle non-formatted prose, necessitating the creation of a new pipeline to improve DefExtractor.

- **G4: Allow users to correct and provide feedback on suggestions.** AI suggestions are not always accurate. Amershi et al. [4] proposed human-AI interaction guidelines, suggesting that when AI makes mistakes, the tool should support users in making corrections and even provide feedback to the AI. The preliminary user study also confirmed that AI suggestions could not perfectly match users' needs, and users still needed to

doi:10.6342/NTU202403869

make manual corrections. However, providing effective feedback is not an easy task for those unfamiliar with AI technology [50]. Therefore, DefExtractor should provide support, such as offering feedback templates for users.

doi:10.6342/NTU202403869

Chapter 4

DefExtractor

Based on the design goals proposed in the previous section (section 3.3), we improved the human-AI collaboration workflow and user interface, and developed a new pair extraction pipeline with prompt engineering based on a large language model.

4.1 Collaboration Workflow

DefExtractor has three pages: Formula Editing, Pair Extraction, and Output Design. The entire editing process is roughly depicted in Figure 1.3, with the two intermediate stages (Figure 1.3 B, C) contained within the Pair Extraction page.

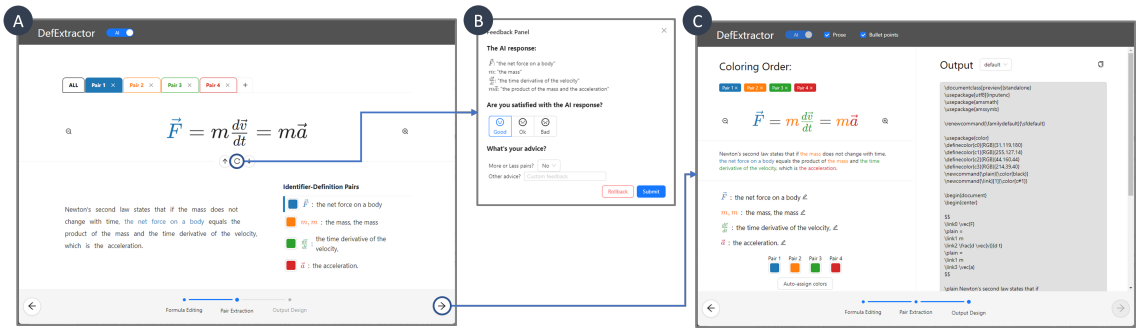


Figure 4.1: *The UI design of DefExtractor. (A): Pair Extraction page; (B): Feedback panel; (C): Output Design page.*

doi:10.6342/NTU202403869

In Formula Editing page, users can edit formulas and explanatory prose with LaTeX on the left side, while the right side renders the results in real-time. Upon entering Pair Extraction page, the AI automatically suggests a set of identifier-definition pairs and presents them with colorization (Figure 1.3 A, G3).

In the preliminary user study, we observed that users tended to adopt two different strategies alternately. One strategy is overall inspection, where users preferred to have a overview of all pairs. The other strategy is the modification of details in each pair, where users wanted to focus on a single pair and wished for other to be hidden. Therefore, DefExtractor can switch between these two modes. As shown in Figure 4.1 A, there are tabs on the formula. Clicking the ”All” tab provides an overview mode, while the other tabs provides a single-pair mode, allowing users to focus on editing a single pair and hides the others. Users can add or remove the identifiers and definitions in a pair by clicking on symbols or terms (G1). In the bottom right corner, there is a list of pairs where users can check the content of each pair or modify their colors.

Between the formula and the prose, there is a reset button that can open the feedback panel (Figure 4.1 B, G4). In the overview mode, this feedback applies to all pairs suggestion. Users can use feedback templates to request AI to provide more or fewer pairs, or they can use the customized one (Figure 1.3 B). In the single-pair mode, there are three buttons: upward arrow, reset, and downward arrow. The downward arrow represents finding the corresponding definition for the current identifier, while the upward arrow does the opposite. The reset button represents feedback as well, allowing users to request AI to provide definitions or identifiers of different lengths or quantities (Figure 1.3 C).

doi:10.6342/NTU202403869

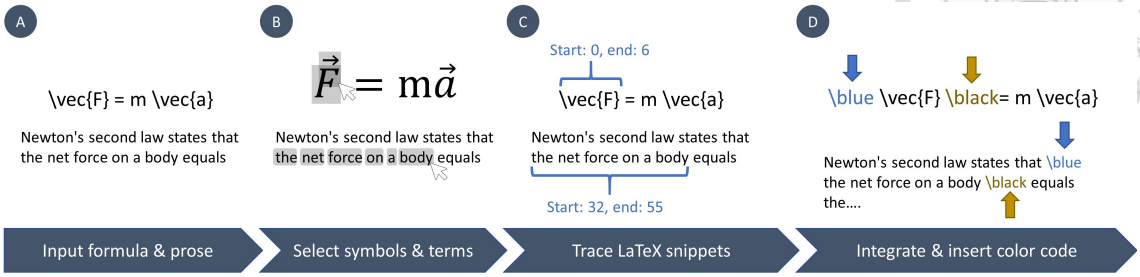


Figure 4.2: *The output creation pipeline. (A): Input the formula in LaTeX; (B): Select the interactive symbols and terms. User can do it manually, or with AI; (C): Trace the LaTeX snippets with customized KaTeX package; (D): Insert the color code in the output.*

In the Output Design page (Figure 4.1 C), the left side displays a preview of the final design, while the right side shows the output LaTeX code. At the top left corner, there is a Color order bar where users can select pairs to be included in the final design. The purpose of this design is to separate the extraction and visualization of pairs, allowing users to quickly test different designs without removal or addition of existing pairs. Additionally, this Color order represents the rendering order, and users can change it by dragging the tabs. This design accounts for the possibility of overlap between pairs, where the order determines the final color of the terms or symbols.

DefExtractor offers two designs: colorization and bullet points (Figure 1.3 D). When explanatory prose is too lengthy, colorization may not be suitable for slide design. In such cases, users can disable prose and use bullet points alone (G2). After completing the design, users can directly copy the LaTeX code from the right side and render it with other LaTeX tools, such as Overleaf[1].

doi:10.6342/NTU202403869

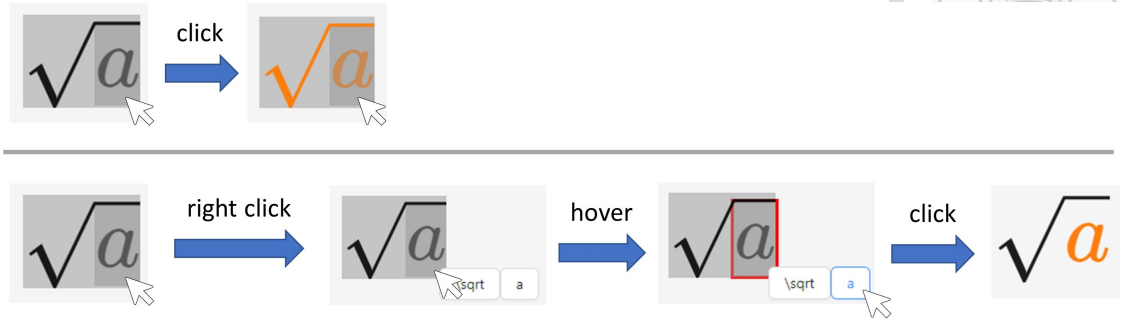


Figure 4.3: *Two ways of symbol selections. Left click is the default methods, which will select all the symbols (top); Right click is for the detail selection. (bottom)*

4.2 Interactive Interface and Output Creation

We design a server-based infrastructure to interact with DefExtractor's front-end interface built with React.js. The backend is developed with Flask framework in Python. The output creation pipeline of DefExtractor is outlined in Figure 4.2. Firstly, DefExtractor utilizes KaTeX[2] to render input LaTeX mathematical formulas with html elements. Due to the complexity of typesetting, these generated html elements often overlap. Thus, DefExtractor offers two different methods for manual selection, as illustrated in Figure 4.3.

To trace corresponding LaTeX snippets from the rendered symbols, we customized the KaTeX package following Gobert et al.'s approach [19]. For explanatory prose, we split it into terms and create buttons for users to select. Finally, DefExtractor will merge this snippets into the strings of identifier and definition, then integrate them into bullet points and apply color codes using Azad's method [5].

For selecting default colors, we employ two approaches. The first involves a fixed

[1] https://www.overleaf.com/
[2] https://github.com/Khan/KaTeX/

doi:10.6342/NTU202403869

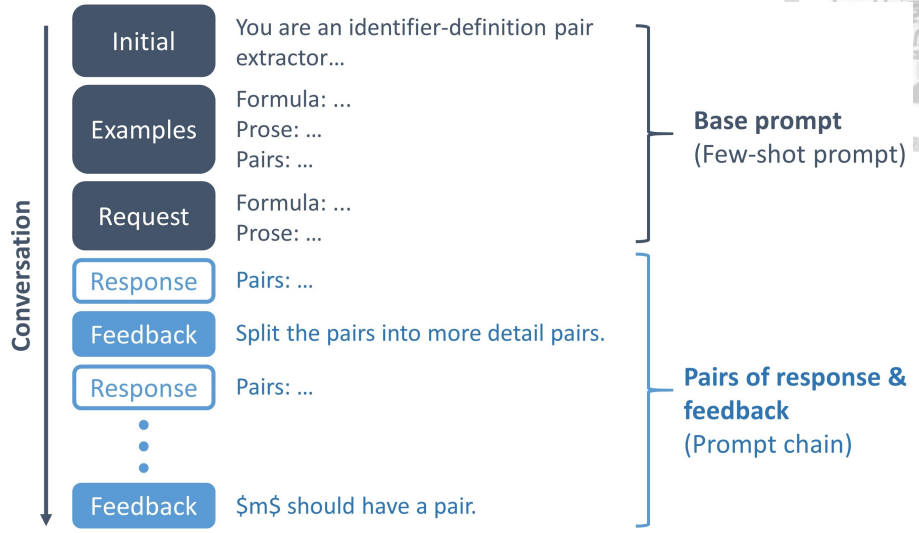


Figure 4.4: *Prompt structure of DefExtractor with an LLM. Each conversation turn consists of a base few-shot prompt and a prompt chain that DefExtractor processes to reveal in the UI.*

color palette, where we use Tableau Classic 10[3] as the default colors. The second method is inspired by Healey's approach [23], where we extract regular polygons on the inscribed circle in the CIE LUV color space [47]. By doing so, we can ensure that each color is as far apart from each other as possible in the CIE LUV color space. Besides, these colors satisfy the property of linear separation, a concept traced back to D'Zmura's research [14]. Both of these criteria are important for selecting effective colors for readers to identify [23].

4.3 Prompt Engineering

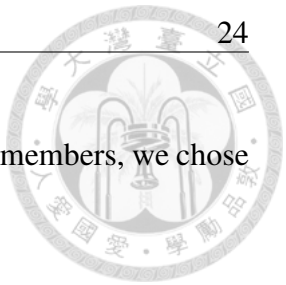
We informally compared the performance of two popular LLMs, GPT-4[4] and Gemini[5], on identifier-definition extraction for non-formatted prose with handcrafted prompts,

[3] https://help.tableau.com/current/pro/desktop/en-us/formatting_create_custom_colors.htm

[4] https://openai.com/gpt-4

[5] https://gemini.google.com/app

doi:10.6342/NTU202403869



and did not find significant differences. Due to the familiarity of team members, we chose GPT-4 to be the LLM behind DefExtractor.

The prompt structure in DefExtractor is illustrated in Figure 4.4, which is a queue of natural language sentences. When the user makes the initial request, a few-shot prompt is created and used as the base prompt. There are three kinds of base prompts, one designed for the overview mode, which can directly extract all pairs based on the formula and prose, while the other two are designed for the single-pair mode, capable of extracting a complete pair based on the selected identifier or definition. As the conversation progresses with AI responses and user feedback, the dialogue becomes longer. These paired responses and feedback constitute a chain that is appended to the base prompt. With these conversation records, the LLM can memorize the entire iterative process, thus yielding better results.

In prompt engineering, we found that requesting the LLM to provide the locations of identifiers and definitions (Figure 4.2 C) yielded less than ideal results. It may be due to the increased complexity of the task, and the examples in few-shot prompts are too complex, which may lead to misdirection. Some guidelines in prompt engineering suggest keeping tasks as simple and clear as possible [18]. Therefore, we make the LLM focus on text rather than locations. When DefExtractor receives the LLM responses, it employs a rule-based approach to track the positions of identifiers and definitions for output generation (Figure 4.2 D).

doi:10.6342/NTU202403869

Chapter 5

Technical Evaluation

To evaluate the performance of the new pair extraction pipeline (section 4.3) based on a large language model, we collected instances from previous literature that fit our application scenario and conducted a technical evaluation using these instances.

5.1 Dataset

Fred Hohman and colleagues [26] collected 7 articles and other 47 examples containing novel mathematical symbol designs. Later, Head et al. [22] expanded and revised this collection, resulting in 47 documents, which they used to propose the concept of math augmentation. From these two collections, we identified cases that perfectly matched our usage scenario, ultimately collecting 16 articles containing a total of 18 mathematical formulas. Our requirements are as follows:

- The design attempts to annotate the mathematical expressions with identifier-definition pairs.
- The explanatory text is continuous prose, rather than individual definitions.

doi:10.6342/NTU202403869

NER	Early prototype (SciBERT-based)			DefExtractor (GPT-4)		
	Symbol F1	Term F1	F1	Symbol F1	Term F1	F1
Formatted	0.47	0.65	0.59	0.68	0.49	0.57
Non-formatted	0.02	0.02	0.02	0.57	0.59	0.59
All	0.14	0.19	**0.17**	0.60	0.57	**0.58**

Table 5.1: *Results of Named Entity Recognition (NER) task.*

Three of these cases were used as examples for our few-shot prompt, so we evaluated the pipeline performance using the remaining 15 cases. Among these, there were 11 non-formatted and 4 formatted cases. Unlike the preliminary user study (4 non-formatted, 4 formatted), in actual cases, non-formatted instances are more common, which further motivates us to improve the pair extraction pipeline.

5.2 Measure

We followed Symlink's approach [31] by breaking down the entire task into two subtasks: Named Entity Recognition (NER) and Relation Extraction (RE), and evaluated the performance of each subtask separately. Notably, since our use cases did not classify entities or relations, we made some adjustments to the evaluation method. For the NER task, we used MUC-5 evaluation metrics [8], considering partial boundary matching. For the RE task, we used standard precision, recall, and F-score metrics. As the dataset did not classify relations, we only considered the entities within the relations, disregarding the relation types.

doi:10.6342/NTU202403869

RE	Early prototype (SciBERT-based)			DefExtractor (GPT-4)		
	Precision	Recall	F1	Precision	Recall	F1
Formatted	0.67	0.35	0.43	0.68	0.53	0.53
Non-formatted	0.03	0.01	0.02	0.72	0.58	0.58
All	0.20	0.10	**0.13**	0.71	0.57	**0.57**

Table 5.2: *Results of Relation Extraction (RE) task.*

5.3 Results

We compared the performance of the early prototype and DefExtractor, with the results shown in Table 5.1 and Table 5.2. As indicated by the preliminary user study, the early prototype performed poorly on non-formatted cases. In the NER task, the early prototype achieved an F1 score of 0.58 on formatted cases but only 0.02 on non-formatted cases. Similarly, in the RE task, the F1 scores were 0.42 and 0.02, respectively. This demonstrates that the model by Lee and Na [33] used in the early prototype was indeed incapable of handling non-formatted cases.

In contrast, DefExtractor, which uses GPT-4 and prompt engineering, achieved similar performance on both non-formatted and formatted cases. In the NER task, the F1 scores were 0.59 and 0.60, respectively, while in the RE task, the scores were 0.56 and 0.58. This makes DefExtractor's average performance significantly better than that of the early prototype. Moreover, even in the formatted cases, DefExtractor achieved performance that was comparable to or even better than that of the early prototype.

However, in NER and RE tasks, DefExtractor still achieves an F1 score of only 0.58 and 0.59, respectively. Figure 5.1 shows one of the worst cases, where the NER F1 score

doi:10.6342/NTU202403869

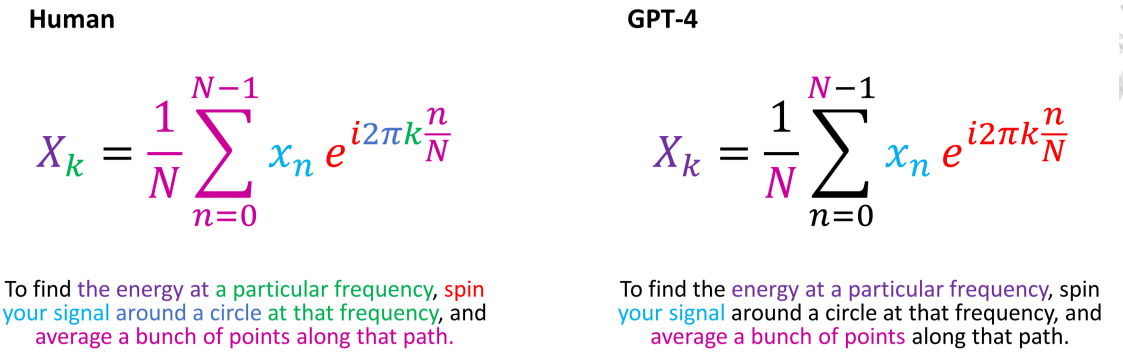


Figure 5.1: *One of the worst cases; On the left is the actual human design, and on the right is the DefExtractor's suggestion. The NER F1 score is 0.39, and the RE F1 score is 0.33.*

is 0.39 and the RE F1 score is 0.33. We can observe that although the scores are not high, DefExtractor still somewhat understood this formula. It correctly identified that X_k represents energy, x_n represents signal, and N represents a bunch of points. Such suggestions may still be helpful for users. To further validate the effectiveness of DefExtractor's suggestions and workflow, we conducted another user study, which will be discussed in the next chapter (chapter 6).

doi:10.6342/NTU202403869

Chapter 6

User Study

We conducted a user study to evaluate whether users found our proposed human-AI collaboration workflow in DefExtractor helpful to create identifier-definition pairs of mathematical formulas. We designed a within-subject study to compare the user experience and authoring process between DefExtractor and a version without AI assistance. Our goal was to investigate whether AI involvement would reduce completion time and workload from users.

6.1 Study Design

6.1.1 Participants

We recruited 12 participants (8 males and 4 females, aged 23 to 29) via an internal lab mailing list of over 30 recipients. The participants were all postgraduate students. The majority were in computer science-related studies, while one was major in design. They did not receive any compensation for their participation. Most of them were not familiar with creating mathematical formulas with LaTeX, where 9 participants rated their

doi:10.6342/NTU202403869

proficiency below 4 at a 7-point Likert scale.

6.1.2 Materials

We selected two formulas from our inspection set that have the same degree of complexity. Each formula had 13 and 10 selectable symbols, respectively, with explanatory prose consisting of 32 and 30 words, respectively. Both of these formulas are fundamental Newton's law, ensuring that the participants' familiarity is comparable. We confirmed with participants in the session that they had seen both formulas but had never edited in a written format.

6.1.3 Procedure

We used a 2×2 within-participant design with two independent variables, AI (AI disabled vs. enabled in DefExtractor) and TASK (T1, T2). Each participant completed two tasks with a counterbalance order of these variables to ensure an equal distribution of participants across the combinations.

We began the experiment with a brief introduction to DefExtractor without introducing the AI aspect to prevent bias. Then, we walked through participants the DefExtractor Editing UI of the condition with a warm-up task. Next, we asked participants to complete a formal task (T1, T2). We recorded editing time, video, and user interaction log in our tool. As shown in Figure 6.1, each formal task included a formula and its corresponding explanatory prose, neither colored nor provided with identifier-definition pairs. The design depended on the participants' choices. After completing each task, participants filled

doi:10.6342/NTU202403869

1

$$\vec{F} = m\frac{dv}{dt} = m\vec{a}$$

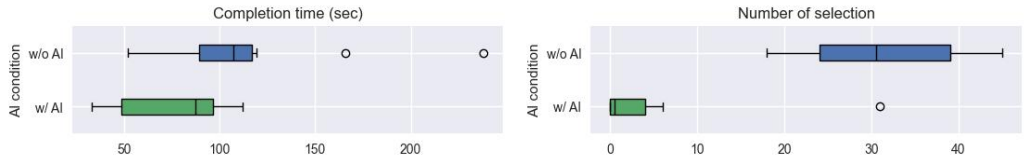
Newton's second law states that if the mass does not change with time, the force equals the product of the mass and the time derivative of the velocity, which is the acceleration.

2

$$F = G\frac{m_1 m_2}{r^2}$$

Newton's law of universal gravitation says that the gravitational force is proportional to the product of their masses and inversely proportional to the square of the distance between their centers.

Figure 6.1: *The two input formulas as task materials (T1, T2) that we provided in our study, including 13 and 10 selectable symbols generated by KaTeX respectively.*



(a) *Task completion time (*TIME*).* **(b)** *The number of selections (*SELECTION*).*

Figure 6.2: *Effect of AI on (a) task completion time and (b) number of selections.*

a workload questionnaire based on the NASA-TLX and without weighting process [20].

After completing both tasks, participants filled a post-study questionnaire to provide an overall evaluation of DefExtractor. Each 60-minute in-person study session ended with a semi-structured interview to gather user feedback and suggestions.

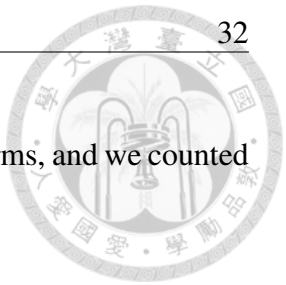
6.2 Results and Findings

All 12 participants completed both tasks under our time constraints. We report quantitative and qualitative results.

6.2.1 Task Completion

In terms of the authoring process, we analyzed participants' completion time TIME and number of selections SELECTION. When users manually added or removed identifiers

doi:10.6342/NTU202403869



or definitions in pairs, they used mouse clicks to select symbols and terms, and we counted these number of selections.

For TIME, the three-way ANOVA showed that none of the interactions between the three factors (PHASE, TASK, AI) were statistically significant (p: PHASE*TASK = 0.82, PHASE*AI = 0.70, TASK*AI = 0.22, PHASE*TASK*AI = 0.47). Using a two-tailed t-test, we found significant differences under different AI conditions (p = 0.03), while PHASE (p = 0.51) and TASK (p = 0.93) did not. As shown in Figure 6.2a, the completion time TIME is significantly shorter with AI assistance (w/o AI: mean = 113.33, std = 46.26, w/ AI: mean=75.33, std = 28.32).

The results for SELECTION were similar, but the interaction was more pronounced. The three-way ANOVA results is bellow: p: PHASE*TASK = 0.41, PHASE*AI = 0.05, TASK*AI = 0.02, PHASE*TASK*AI = 0.41. The results under the two-tailed t-test were the same, with no significant effects for PHASE (p = 0.78) and TASK (p = 0.64) except for AI conditions (p = 0.00). As shown in Figure 6.2b, the number of selections SELECTION was significantly fewer with AI assistance (w/o AI: mean = 31.33, std = 9.11, w/ AI: mean = 4.08, std = 8.36).

6.2.2 Workload

In terms of workloads, the impact of AI is illustrated in Figure 6.3. The results of the two-tailed t-test indicate significant differences in Mental Demand (p = 0.02), Physical Demand (p = 0.01), and Temporal Demand (p = 0.00). When using a one-tailed t-test to determine whether the values are lower with AI, significant differences are also observed

doi:10.6342/NTU202403869

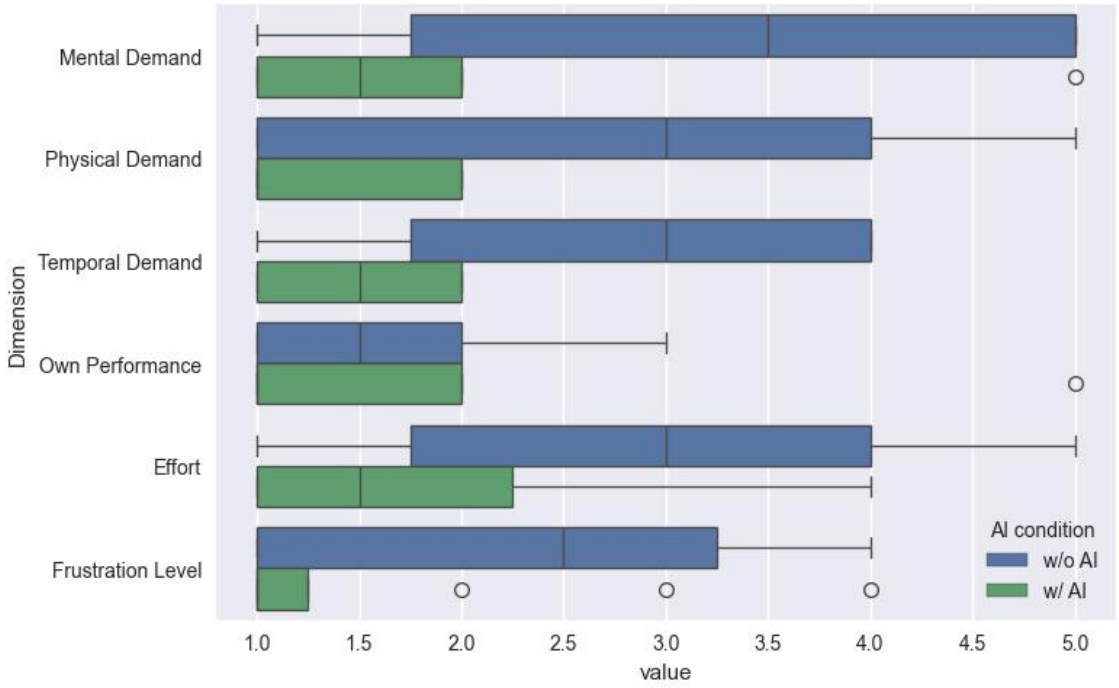


Figure 6.3: *Rating of the Workload under different AI conditions on 7-point Likert scale (1 = very little, 4 = normal, 7 = very much).*

in Effort ($p = 0.03$) and Frustration Level ($p = 0.03$). The absence of a difference in Own Performance dimension is understandable, as the role of AI is to expedite the editing process, while the final design is still determined by the user. Overall, AI assistance effectively reduces the workload of the users.

6.2.3 Rating and Feedback

Figure 6.4 shows the results of the post-study questionnaire. Most participants agreed that DefExtractor was easy to use (1 neutral, 4 agree, 7 strongly agree) and easy to learn (1 somewhat agree, 6 agree, 5 strongly agree). They also agreed that they would be likely use DefExtractor for extracting and visualizing identifier-definition pairs in mathematical formulas (4 agree, 8 strongly agree).

doi:10.6342/NTU202403869

Figure 6.4: *The overall rating of DefExtractor in the post-study questionnaire on 7-point Likert scale (1 = strongly disagree, 4 = neutral,, 7 = strongly agree).*

Participants praised DefExtractor for its assistance with coloring and typesetting, finding the overall editing process much easier compared to other common tools such as MS Word or PowerPoint (P1˜4, P6, P8, P10, P11). Some participants appreciated the automatic color recommendations, as it spared them the hassle of choosing colors, and they were satisfied with the aesthetic appeal (P4, P8, P10). Several participants mentioned the pair-oriented design, finding it convenient to modify the colors of entire pairs at once or directly add or remove entire pairs (P3, P8). P2 further noted that DefExtractor helped him discover how coloring could augment formulas, making him more inclined to use it for creating slides or notes.

In terms of AI evaluation, the majority of participants were satisfied with the performance of the AI (1 neutral, 1 somewhat agree, 6 agree, 4 strongly agree). Although not all participants agreed that the AI suggestions met their expectations (1 somewhat disagree, 1 somewhat agree, 4 agree, 6 strongly agree), all participants at least somewhat agreed that AI was helpful (3 somewhat agree, 2 agree, 7 strongly agree).

Many participants mentioned that with the AI suggestions, they only needed to check

doi:10.6342/NTU202403869

and modify some details, greatly speeding up the editing process (P2˜3, P6˜8, P10, P12). Several participants believed that AI could help them review the meaning of formulas, considering such assistance also helpful for understanding unfamiliar ones (P1, P4, P9). P9 even felt that AI performed better than humans, believing that AI had a more precise and comprehensive understanding of identifiers.

6.2.4 Discussion

Although the statistical results of TIME indicated that it is shorter, three participants experienced longer editing times with AI assistance ($\text{TIME}_{\text{w/ AI}} - \text{TIME}_{\text{w/o AI}}$: P2 = 19, P5 = 17, P12 = 13). According to their feedback, this was because they found the tasks too simple, leading to rapid manual editing speeds. Their manual editing times were 93, 75, and 96 seconds, notably lower than the average of 113.33 seconds. They primarily spent time for checking, with the actual number of clicks still being lower with AI assistance ($\text{SELECTION}_{\text{w/ AI}} - \text{SELECTION}_{\text{w/o AI}}$: P2 = -12, P5 = -24, P12 = -33). Two of them were still satisfied with the AI suggestions (1 neutral, 1 somewhat agree, 1 strongly agree), and all of them at least agreed that the AI suggestions were helpful(2 somewhat agree, 1 strongly agree).

In terms of bidirectional interaction with AI, 6 participants provided feedback to the AI, while the other 6 participants thought that the AI's performance was already satisfactory and only required manual modification for some details. Among the 6 participants who provided feedback, except for one who did it twice, the others only did it once. Such passive interactive scenarios may echo the findings of Zamfrescu-Pereira et al [50]. Some

doi:10.6342/NTU202403869

participants attributed this to the simplicity of the task.

As for the feedback content, 5 participants requested more pairs from the AI in the overview mode (P1, P5, P8˜9, P12), while one participant asked for shorter definitions in the single-pair mode (P4). Two participants provided interesting feedback in the warm-up phase (P5, P8). P5 attempted to request the AI to suggest only identifiers on the right side of the equation, and P8 asked the AI to ignore vector symbols; both of these feedback requests yielded satisfactory results.

doi:10.6342/NTU202403869

Chapter 7

Limitations and Future Work

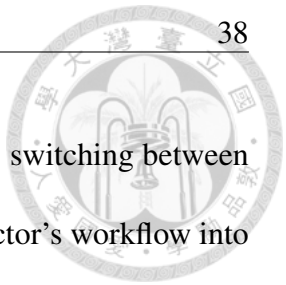
7.1 Limitations

Our user study aimed to demonstrate that AI assistance could shorten editing time and reduce burden. For fairness of comparison, we conducted the experiment with two fixed tasks (Figure 6.1). However, real-world scenarios may be more complex. Some participants felt the formulas in the study were too simple, and they believed that DefExtractor's effectiveness would be more apparent in more complex tasks (P2, P3). Therefore, the robustness of DefExtractor and its performance in high-difficulty tasks need further experimentation for validation.

7.2 Compatibility With Other Authoring Tools

DefExtractor is a standalone tool that focuses solely on the extraction and design of identifier-definition pairs and does not support other complex editing tasks, such as embedding charts. The singularity of its functionality was dissatisfying to some participants (P4˜5, P7, P12). Its output is a piece of LaTeX code, allowing users to continue using

doi:10.6342/NTU202403869



other LaTeX tools, like Overleaf, for more complex editing. However, switching between such tools may still incur additional workloads. Integrating DefExtractor's workflow into larger editing tools could facilitate the wider adoption of this design pattern.

7.3 Mathematical Formulas Construction

LaTeX, as one of the mainstream typesetting languages for mathematical formulas, is widely used in scientific and technical documents. Many language models related to mathematics primarily deal with mathematical symbols composed in LaTeX [35]. However, mathematical formulas in real world may often lack LaTeX source code for use. Additionally, LaTeX takes time to master, which may increase the learning curve for using DefExtractor. Some AI models try to convert mathematical formula images into LaTeX [12], and integrating them may further enhance the usability of DefExtractor. Furthermore, some participants believe that DefExtractor has reading assistance capabilities (P1, P4, P9). Making DefExtractor not limited to LaTeX input could enhance its versatility.

doi:10.6342/NTU202403869

Chapter 8

Conclusion

In this paper, we present DefExtractor, an LLM-based interactive tool that helps authors extract and visualize identifier-definition pairs with AI in an iterative process. DefExtractor takes a LaTeX input to identify the semantics. It automatically suggests colored identifiers and definitions based on the LLM response. Users can edit the suggestions via text prompt or syntax, for AI to adapt the user changes. A technical evaluation showed that DefExtractor outperforms previous model in our scenario, and it is able to deal with the non-formatted cases. A usability study with 12 participants demonstrated that DefExtractor effectively reduced the workloads of authors by shortening editing time compared with a baseline tool.

doi:10.6342/NTU202403869

Bibliography

[1] T. L. Adams. Reading mathematics: More than words can say. *The reading teacher*, 56(8):786–795, 2003.

[2] L. Alcock. e-proofs: Student experience of online resources to aid understanding of mathematical proofs. In *Proceedings of the 12th Conference on Research in Undergraduate Mathematics Education. Raleigh, NC: Special Interest Group of the Mathematical Association of America on Research in Undergraduate Mathematics Education*. Citeseer, 2009.

[3] M. Alexeeva, R. Sharp, M. A. Valenzuela-Escárcega, J. Kadowaki, A. Pyarelal, and C. Morrison. Mathalign: Linking formula identifiers to their contextual natural language descriptions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2204–2212, 2020.

[4] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–13, 2019.

[5] K. Azad. Colorized math equations. *Better Explained*, 2017.

[6] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[7] J. Brooke. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189(3):189–194, 1996.

[8] N. Chinchor and B. M. Sundheim. Muc-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*, 1993.

[9] N. Chulpongsatorn, M. S. Lunding, N. Soni, and R. Suzuki. Augmented math: Authoring

doi:10.6342/NTU202403869

ar-based explorable explanations by augmenting static math textbooks. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–16, 2023.

[10] M. Conlen and J. Heer. Idyll: A markup language for authoring and publishing interactive articles on the web. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 977–989, 2018.

[11] V. F. de Santana, R. de Oliveira, L. D. A. Almeida, and M. C. C. Baranauskas. Web accessibility and people with dyslexia: a survey on techniques and guidelines. In *Proceedings of the international cross-disciplinary conference on web accessibility*, pages 1–9, 2012.

[12] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush. Image-to-markup generation with coarse-to-fine attention. In *International Conference on Machine Learning*, pages 980–989. PMLR, 2017.

[13] A. N. Dragunov and J. L. Herlocker. Designing intelligent and dynamic interfaces for communicating mathematics. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 236–238, 2003.

[14] M. D'Zmura. Color in visual search. *Vision research*, 31(6):951–966, 1991.

[15] D. Ferreira, M. Thayaparan, M. Valentino, J. Rozanova, and A. Freitas. To be or not to be an integer? encoding variables for mathematical text. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 938–948, 2022.

[16] R. Fok, H. Kambhamettu, L. Soldaini, J. Bragg, K. Lo, M. Hearst, A. Head, and D. S. Weld. Scim: Intelligent skimming support for scientific papers. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 476–490, 2023.

[17] K. I. Gero, V. Liu, and L. Chilton. Sparks: Inspiration for science writing using language models. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference*, pages 1002–1019, 2022.

[18] L. Giray. Prompt engineering with chatgpt: a guide for academic writers. *Annals of biomedical engineering*, 51(12):2629–2633, 2023.

[19] C. Gobert and M. Beaudouin-Lafon. i-latex: Manipulating transitional representations be-

doi:10.6342/NTU202403869

tween latex code and generated documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2022.

[20] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[21] A. Head, K. Lo, D. Kang, R. Fok, S. Skjonsberg, D. S. Weld, and M. A. Hearst. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2021.

[22] A. Head, A. Xie, and M. A. Hearst. Math augmentation: How authors enhance the readability of formulas using novel visual design practices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2022.

[23] C. G. Healey. Choosing effective colours for data visualization. In *Proceedings of Seventh Annual IEEE Visualization'96*, pages 263–270. IEEE, 1996.

[24] J. Heer, M. Conlen, V. Devireddy, T. Nguyen, and J. Horowitz. Living papers: A language toolkit for augmented scholarly communication. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13, 2023.

[25] F. Hohman, M. Conlen, J. Heer, and D. H. P. Chau. Communicating with interactive articles. *Distill*, 5(9):e28, 2020.

[26] F. Hohman and other contributors. Awesome mathematical notation design. 2020.

[27] H. Jo, D. Kang, A. Head, and M. A. Hearst. Modeling mathematical notation semantics in academic papers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3102–3115, 2021.

[28] H. B. Kang, T. Wu, J. C. Chang, and A. Kittur. Synergi: A mixed-initiative system for scholarly synthesis and sensemaking. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.

[29] J. Kim, S. Suh, L. B. Chilton, and H. Xia. Metaphorian: Leveraging large language models

doi:10.6342/NTU202403869

to support extended metaphor creation for science writing. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, pages 115–135, 2023.

[30] A. Kohlhase, M. Kohlhase, and T. Ouypornkochagorn. Discourse phenomena in mathematical documents. In *Intelligent Computer Mathematics: 11th International Conference, CICM 2018, Hagenberg, Austria, August 13-17, 2018, Proceedings 11*, pages 147–163. Springer, 2018.

[31] V. D. Lai, A. P. B. Veyseh, F. Dernoncourt, and T. H. Nguyen. Semeval 2022 task 12: Symlink-linking mathematical symbols to their descriptions. *arXiv preprint arXiv:2202.09695*, 2022.

[32] Y. Lai, A. Kankanhalli, and D. Ong. Human-ai collaboration in healthcare: A review and research agenda. 2021.

[33] S.-M. Lee and S.-H. Na. Jbnu-cclab at semeval-2022 task 12: Machine reading comprehension and span pair classification for linking mathematical symbols to their descriptions. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1679–1686, 2022.

[34] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[35] J. Meadows and A. Freitas. Introduction to mathematical language processing: Informal proofs, word problems, and supporting tasks. *Transactions of the Association for Computational Linguistics*, 11:1162–1184, 2023.

[36] A. Miniukovich, A. De Angeli, S. Sulpizio, and P. Venuti. Design guidelines for web readability. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, pages 285–296, 2017.

[37] M. Österholm. Characterizing reading comprehension of mathematical texts. *Educational studies in mathematics*, 63:325–346, 2006.

[38] R. Pagael and M. Schubotz. Mathematical language processing project. *arXiv preprint arXiv:1407.0167*, 2014.

doi:10.6342/NTU202403869

[39] M. Schubotz, A. Grigorev, M. Leich, H. S. Cohl, N. Meuschke, B. Gipp, A. S. Youssef, and V. Markl. Semantification of identifiers in mathematics for better math information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 135–144, 2016.

[40] O. Shaikh, J. Saad-Falcon, A. P. Wright, N. Das, S. Freitas, O. Asensio, and D. H. Chau. Energyvis: interactively tracking and exploring energy consumption for ml models. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2021.

[41] D. Wang, E. Churchill, P. Maes, X. Fan, B. Shneiderman, Y. Shi, and Q. Wang. From human-human collaboration to human-ai collaboration: Designing ai systems that can work together with people. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pages 1–6, 2020.

[42] D. Wang, J. D. Weisz, M. Muller, P. Ram, W. Geyer, C. Dugan, Y. Tausczik, H. Samulowitz, and A. Gray. Human-ai collaboration in data science: Exploring data scientists' perceptions of automated ai. *Proceedings of the ACM on human-computer interaction*, 3(CSCW):1–24, 2019.

[43] F. Wang, X. Liu, O. Liu, A. Neshati, T. Ma, M. Zhu, and J. Zhao. Slide4n: Creating presentation slides from computational notebooks with human-ai collaboration. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2023.

[44] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[45] M. Wolska and M. Grigore. Symbol declarations in mathematical writing. 2010.

[46] Z. Wu, J. Li, K. Ma, H. Kambhamettu, and A. Head. Ffl: A language and live runtime for styling and labeling typeset math formulas. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–16, 2023.

[47] G. Wyszecki and W. S. Stiles. *Color science: concepts and methods, quantitative data and formulae*, volume 40. John wiley & sons, 2000.

doi:10.6342/NTU202403869

[48] K. G. Yoko, M.-Q. Nghiem, Y. Matsubayashi, and A. Aizawa. Extracting definitions of mathematical expressions in scientific papers. In *Proceedings of the 26th Annual Conference of JSAI*, pages 1–7, 2012.

[49] H. I. Yung and F. Paas. Effects of computer-based visual representation on mathematics learning and cognitive load. 2015.

[50] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2023.

[51] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[52] Q. Zheng, Y. Tang, Y. Liu, W. Liu, and Y. Huang. Ux research on conversational human-ai interaction: A literature review of the acm digital library. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–24, 2022.

doi:10.6342/NTU202403869

Chapter 9

Appendix

ID	Formula	Source
F0	$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_N x_N$ $y = \beta_0 + f1(x_1) + f_2(x_2) + ... + f_N(x_N)$	https://fredhohman.com/papers/19-gamut-chi.pdf
F1	$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{i2\pi k \frac{n}{N}}$	https://web.archive.org/web/20120418231513/www.altdevblogaday.com/2011/05/17/understanding-the-fourier-transform/
F2	$c^2 = a^2 + b^2$	https://betterexplained.com/articles/surprising-uses-of-the-pythagorean-theorem/
F3	$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i)(\omega_i \cdot \mathbf{n}) d\omega_i$	https://chuckleplant.github.io/2017/05/28/light-shafts.html
F4	$L(s, \theta) = L_o e^{-\tau s} + \frac{1}{\tau} E_{sun} S(\theta)(1 - e^{-\tau s})$	
F5	$L(s, \theta, \phi) = (1 - D(\phi)) L(s, \theta)$	
F6	$\frac{\partial \mathcal{L}}{\partial x_i} = \sum_{j \le n} \frac{\partial \mathcal{L}}{\partial y_j} \frac{\partial y_j}{\partial x_i}$	https://taliesin.ai/projects/edu/indaba-2019/
F7	$e^{i\pi} = -1$	https://betterexplained.com/articles/math-and-analogies/
F8	$e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^{1 \cdot n}$	https://betterexplained.com/articles/colorized-math-equations/
F9	$\frac{dR}{dt} = \alpha \cdot R - \beta \cdot R \cdot F$	https://www.redblobgames.com/x/1913-equation-formatting/
F10	$g(y) = \beta_0 + f_1(x_1) + f_2(x_2) + ... + f_M(x_M)$	https://interpret.ml/gam-changer/
F11	$p_i = (p_{chipset} + \sum_{g=1}^{G} p_g) \cdot 1.59$	https://arxiv.org/pdf/2103.16435.pdf
F12	$Y_{it} = \alpha + \beta \text{Group}_i + \gamma \text{Time}_t + \delta(\text{Group}_i \times \text{Time}_t) + \varepsilon_{it}$	https://twitter.com/andrewheiss/status/1312948770032807936
F13	$MeanSquareError = \frac{1}{n}\sum_{i=1}^{n}(Y_{prediction_i} - Y_i)^2$	https://jalammar.github.io/visual-numpy/
F14	$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi k \frac{n}{N}}$ $x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k \cdot e^{i2\pi k \frac{n}{N}}$	https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/
P0	$e^{ix} = \cos(x) + i\sin(x)$	https://betterexplained.com/articles/colorized-math-equations/
P1	$\Pr(H\|E) = \frac{\Pr(E\|H)\Pr(H)}{\Pr(E\|H)\Pr(H) + \Pr(E\|\text{not } H)\Pr(\text{not } H)}$	https://betterexplained.com/articles/an-intuitive-and-short-explanation-of-bayes-theorem/
P2	$\cos\theta = \frac{<a,b>}{\|\|a\|\|\cdot\|\|b\|\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i a_i}\sqrt{\sum_{i=1}^{n} b_i b_i}}$	https://twitter.com/andrew_n_carr/status/1346172166077726720

Table 9.1: *Technical evaluation dataset; P0 ~2 are the cases used as examples for our few-shot prompt.*

doi:10.6342/NTU202403869

Questions (7-point Likert scale)	
Descriptions	**Endpoints**
NASA-TLX	
How much mental and perceptual activity was required? Was the task easy or demanding, simple or complex?	Low/High
How much physical activity was required? Was the task easy or demanding, slack or strenuous?	Low/High
How much time pressure did you feel due to the pace at which the tasks or task elements occurred? Was the pace slow or rapid?	Low/High
How successful were you in performing the task? How satisfied were you with your performance?	Good/Poor
How hard did you have to work (mentally and physically) to accomplish your level of performance?	Low/High
How irritated, stressed, and annoyed versus content, relaxed, and complacent did you feel during the task?	Low/High
Overall	
I found DefExtractor easy to learn	Strongly disagree/Strongly agree
I found DefExtractor easy to use	Strongly disagree/Strongly agree
I would likely use DefExtractor to extract and visualize the identifier-definition pairs	Strongly disagree/Strongly agree
I am satisfied with the AI suggestions in DefExtractor	Strongly disagree/Strongly agree
I found that the AI suggestions in DefExtractor meet my expectations	Strongly disagree/Strongly agree
I found that the AI suggestions in DefExtractor are helpful	Strongly disagree/Strongly agree

Table 9.2: *Questionnaires in the user study*

Role	**Content**
System	You are a symbol-definition pair extractor. Given a Formula and its corresponding Prose, you will extract the symbol-definition pairs from them. Note that the Symbols should be substrings of the Formula, and the Definitions should be substrings of the Prose. Do not output anything other than symbol-definition pairs. I will give you some feedback to your response, modify your response with my feedback.
User	Formula: $\Pr(H\|E) = \frac{\Pr(E\|H)\Pr(H)}{\Pr(E\|H)\Pr(H)+\Pr(E\|\text{not } H)\Pr(\text{not } H)}$; Prose: "The chance evidence is real (supports a hypothesis) is the chance of a true positive among all positives (true or false)"
Assistant	$\Pr(H\|E)$: "The chance evidence is real"; $\Pr(E\|H)\Pr(H)$: "the chance of a true positive", "true"; $\Pr(E\|\text{not } H)\Pr(\text{not } H)$: "false"
User	Formula: $\cos\theta = \frac{<a,b>}{\|\|a\|\|\cdot\|\|b\|\|}$; Prose: "The angle between two vectors is calculated by finding the inner product between the first and second vectors and dividing by the length of each then take the arc cosine."
Assistant	θ: "angle"; a, b: "two vectors", "the first and second vectors"; <, >: "inner product"; ||: "the length"; cos: "arc cosine"
User	Formula: $e^{ix} = \cos(x) + i\sin(x)$; Prose: "Growth in a perpendicular direction over time is circular: here are the horizontal and vertical coordinates."
Assistant	e: "Growth"; i: "perpendicular direction"; x: "time"; $\cos(x)$: "horizontal"; $i\sin(x)$: "vertical"
User	Formula: $ {input formula} $; Prose: " {input prose} "

Table 9.3: *The base prompt for pair extraction*

doi:10.6342/NTU202403869

Role	Content
System	You are a definition extractor. Given a Formula, its corresponding Prose and some Symbols, you will extract the definition of Symbols in the Prose. Note that the Symbols are the substrings of the Formula, and the definitions should be substrings of the Prose. Do not output anything other than symbol-definition pairs. I will give you some feedback to your response, modify your response with my feedback.
User	Formula: $\Pr(H\|E) = \frac{\Pr(E\|H)\Pr(H)}{\Pr(E\|H)\Pr(H)+\Pr(E\|\text{not } H)\Pr(\text{not } H)}$; Prose: "The chance evidence is real (supports a hypothesis) is the chance of a true positive among all positives (true or false)"; Symbols: $\Pr(E\|H)$
Assistant	$\Pr(H\|E)$: "The chance evidence is real"
User	Formula: $\cos\theta = \frac{<a,b>}{\|\|a\|\|\cdot\|\|b\|\|}$; Prose: "The angle between two vectors is calculated by finding the inner product between the first and second vectors and dividing by the length of each then take the arc cosine." Symbols: a, b
Assistant	a, b: "two vectors", "the first and second vectors"
User	Formula: $e^{ix} = \cos(x) + i\sin(x)$; Prose: "Growth in a perpendicular direction over time is circular: here are the horizontal and vertical coordinates." Symbols: e
Assistant	e: "Growth"
User	Formula: $ {input formula} $; Prose: " {input prose} "; Symbols: $ {input identifier} $

Table 9.4: *The base prompt for extracting definition given identifier*

Role	Content
System	You are a symbol extractor. Given a Formula, its corresponding Prose and some Definitions, you will extract the symbols of Definitions in the Formula. Note that the Definitions are the substrings of the Prose, and the symbols should be substrings of the Formula. Do not output anything other than symbol-definition pairs. I will give you some feedback to your response, modify your response with my feedback.
User	Formula: $\Pr(H\|E) = \frac{\Pr(E\|H)\Pr(H)}{\Pr(E\|H)\Pr(H)+\Pr(E\|\text{not } H)\Pr(\text{not } H)}$; Prose: "The chance evidence is real (supports a hypothesis) is the chance of a true positive among all positives (true or false)"; Definitions: "the chance of a true positive", "true"
Assistant	$\Pr(H\|E)$: "The chance evidence is real"
User	Formula: $\cos\theta = \frac{<a,b>}{\|\|a\|\|\cdot\|\|b\|\|}$; Prose: "The angle between two vectors is calculated by finding the inner product between the first and second vectors and dividing by the length of each then take the arc cosine." Definitions: "two vectors", "the first and second vectors"
Assistant	a, b: "two vectors", "the first and second vectors"
User	Formula: $e^{ix} = \cos(x) + i\sin(x)$; Prose: "Growth in a perpendicular direction over time is circular: here are the horizontal and vertical coordinates." Definitions: "inner product"
Assistant	e: "Growth"
User	Formula: $ {input formula} $; Prose: " {input prose} "; Definitions: " {input definition} "

Table 9.5: *The base prompt for extracting identifier given definition*

doi:10.6342/NTU202403869