arXiv:2305.11473v1 [cs.HC] 19 May 2023

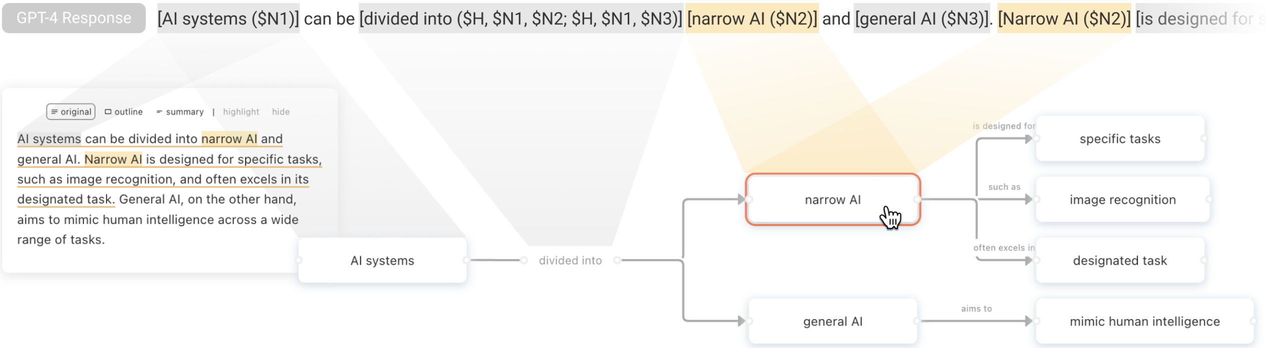
Graphologue: Exploring Large Language Model Responses with Interactive Diagrams

Peiling Jiang[*]
peiling@ucsd.edu
University of California San Diego
La Jolla, California, USA

Jude Rayan[*]
jrayan@ucsd.edu
University of California San Diego
La Jolla, California, USA

Steven P. Dow
spdow@ucsd.edu
University of California San Diego
La Jolla, California, USA

Haijun Xia
haijunxia@ucsd.edu
University of California San Diego
La Jolla, California, USA



[AI systems ($N1)] can be [divided into ($H, $N1, $N2; $H, $N1, $N3)] [narrow AI ($N2)] and [general AI ($N3)]. [Narrow AI ($N2)] [is designed for s

AI systems can be divided into narrow AI and general AI. Narrow AI is designed for specific tasks, such as image recognition, and often excels in its designated task. General AI, on the other hand, aims to mimic human intelligence across a wide range of tasks.

Figure 1: Graphologue simultaneously constructs an interactive diagram as GPT-4 text responses are streamed in.

ABSTRACT

Large language models (LLMs) have recently soared in popularity due to their ease of access and the unprecedented intelligence exhibited on diverse applications. However, LLMs like ChatGPT present significant limitations in supporting complex information tasks due to the insufficient affordances of the text-based medium and linear conversational structure. Through a formative study with ten participants, we found that LLM interfaces often present long-winded responses, making it difficult for people to quickly comprehend and interact flexibly with various pieces of information, particularly during more complex tasks. We present Graphologue, an interactive system that converts text-based responses from LLMs into graphical diagrams to facilitate information-seeking and question-answering tasks. Graphologue employs novel prompting strategies and interface designs to extract entities and relationships from LLM responses and constructs node-link diagrams in real-time. Further, users can interact with the diagrams to flexibly adjust the graphical presentation and to submit context-specific prompts to obtain more information. Utilizing diagrams, Graphologue enables graphical, non-linear dialogues between humans and LLMs, facilitating information exploration, organization, and comprehension.

CCS CONCEPTS

• **Information systems → Users and interactive retrieval**; • **Human-centered computing → Human computer interaction (HCI)**; *Visualization.*

KEYWORDS

Large Language Model, Graphical User Interface

1 INTRODUCTION

Large **L**anguage **M**odels (LLMs) have seen a surge in popularity due to their impressive ability to generate high-quality textual responses to natural language prompts across a wide variety of tasks [11]. Today, billions of people are using interfaces to LLMs, such as ChatGPT, to obtain information and answers to questions. With the potential to dramatically transform how people complete information processing tasks, LLMs are becoming increasingly important tools in various fields [11, 53].

Despite their potential, interactions with LLMs are primarily mediated via text-based conversational interfaces, which presents inherent limitations in terms of supporting complex information activities. As a sequence of symbols, text can be insufficient for communicating concepts that contain complex relationships and structures, often leading to verbose responses that demand substantial effort to digest [16]. In addition, the linear conversation structure can hinder the iterative concept exploration workflows

[*] Both authors contributed equally to this research.

that employ non-linear structures (e.g., follow-up questions), resulting in excessive and verbose conversational exchanges [31]. These intrinsic constraints of text-based conversational interfaces are the bottlenecks for effectively leveraging LLMs for complex information tasks [56].

Graphical representations, on the other hand, can compensate for the aforementioned limitations. Graphs and diagrams can display information in a non-linear manner, enabling the more flexible organization of concepts and reducing the cognitive load needed for comprehension [2, 43, 67]. Interactive graphics can further facilitate the manipulation of information, enabling users to effectively obtain, organize, transform, and make sense of information [28, 37, 38, 69]. For these reasons, graphical representations have been extensively studied and utilized in various fields, including HCI [32], Visualization [17], Cognitive Science [20], Communication [50], and beyond. The goal of this project, therefore, is to capitalize on the advantages of graphical representations to mitigate challenges associated with text-based conversational interfaces in LLM applications. We envision a graphical conversation between humans and LLMs, continuing the fruitful and long-lasting endeavor in HCI [63].

Because text and graphics are both versatile media that can utilize different formats and styles for different tasks, it is important to target specific tasks for meaningful generalization. We focus on supporting exploratory information-seeking, concept explanation, and question-answering tasks with LLMs using graphical representations. To understand the challenges of utilizing LLMs for these tasks, we conducted a formative study with ten participants to observe how they used ChatGPT to explore and learn about a domain of interest. Participants reported that the text-based responses from LLMs are often verbose and often time-consuming to comprehend. In addition, we observed that the limited text medium and the linear conversational structure imposed many cumbersome interactions, such as repetitive copy-and-paste and back-and-forth scrolling, to carry out complex information tasks.

Informed by findings from the formative study, we designed Graphologue, an interactive system that converts textual responses from LLMs into graphical diagrams, in real-time, to facilitate complex and multi-faceted information-seeking and question-answering tasks. Graphologue employs novel prompting strategies to have LLMs recognize and annotate entities and relationships inline within the generated responses to facilitate real-time construction of node-link diagrams. To avoid overly complex diagrams, Graphologue employs prompting strategies and interaction techniques to enable users to flexibly control the complexity of the diagrams. For example, users can toggle the diagrams to show only salient relationships, collapse branches of the diagrams to reduce the presented information, and combine separate smaller diagrams into one diagram to view all concepts as a whole. To gain more information about concepts presented in the diagram, users can employ direct manipulation of the graphical interface, which is subsequently translated into context-aware prompts for the LLM, enabling users to engage in a "graphical dialogue" with LLMs.

To evaluate the effectiveness of Graphologue in assisting in the graphical interaction with LLM for information-seeking tasks, we conducted a user evaluation with seven experienced LLM users and found that Graphologue helped them quickly grasp key concepts and their connections while ensuring sufficient control of the complexity of the diagrams. Together with other representations that are interactively synchronized with the diagrams, such as the raw text and the outline, Graphologue enabled participants to leverage the combined strengths of different representations to understand LLM responses at various levels and scales. This work thus makes the following contributions:

- A formative study that uncovered limitations in using a conversational interface for complex sensemaking tasks.
- Graphologue, an interactive system that employs prompting strategies and interaction techniques to enable real-time construction of and interaction with node-link diagrams based on LLM-generated information to facilitate comprehension and exploration.
- A qualitative evaluation that provided insights regarding the advantages and disadvantages of the diagrammatic representation, the complimentary usage of multiple representations, and future directions of employing graphical interfaces to interact with LLMs.

2 RELATED WORK

As our research aims to address the many challenges of natural language interfaces for LLMs by using graphical representations, we review prior work on Natural Language User Interfaces and LLMs, generating graphical representations from text, as well as Visualization and multilevel abstraction of information.

2.1 Natural Language User Interfaces and LLMs

The impressive performance and the public release of LLMs have sparked communities' imagination for applying this technology to domains such as programming [18, 54, 65], writing support [22, 30, 74], learning [4, 42, 46], and many others [8, 15, 61], and further promoting the notion of natural language interfaces the HCI community has been exploring.

One key benefit of natural language interfaces is that they enable users to directly articulate their intended actions and goals without learning and utilizing complex manual user interfaces. Pioneering systems, such as SHRDLU [68], Put-that-there [7], and Quickset [24] allowed users to verbally instruct a computer with natural language commands. Later research extended this interaction paradigm to data analysis [29], image editing [41], and more. A limitation of these systems, however, is that they require users to translate their high-level design intention to low-level, rigid commands or queries, limiting the fluidity and expressiveness that afforded by language as a communication medium [71].

Another approach to natural language interfaces has been extracting user intents from their natural expressions to enable less rigid communication between humans and computers by leveraging advanced natural language understanding and domain-specific knowledge. Iris, for example, enables users to describe data analysis goals and disambiguate system interpretations using natural expressions [27]. CrossData infers the desired data values and operations from text intended for reporting data insights rather than instructing the system [19]. Crosspower employs a human-in-the-loop approach, by enabling users to interact with linguistic structures in a video script to convey high-level design goals regarding the graphical content and structures [71].

While the unparalleled language understanding and generation capability of LLMs enables users to obtain meaningful responses with flexible natural language expressions, the inherent intelligence architecture of LLMs poses additional usability challenges. For outcomes with specific requirements, users must go through arduous and time-consuming prompt engineering to craft convoluted prompts in order to obtain results that are closely aligned with their intents [45, 57, 75].

Our work leverages recent advancements in natural language processing but shares the spirit of Sketchpad, a seminal work in HCI that pioneered the graphical communication between humans and machines [63]. Specifically, we leverage GPT-4's generative power to not only obtain information but also to annotate its responses to facilitate the creation of graphical representations of textual responses from the LLMs.

2.2 Text to Graphical Representations

Graphical representations are prevalent across various domains to enhance communication and sensemaking [5]. By leveraging human aptitude for visual information processing, they offer advantages in comprehension, memory, and inference of the content [1], making them an effective tool to improve information understanding and learning [37, 39, 47, 58, 62]. In addition, the generation and modification of graphical representations, such as sketching and annotating, make them ideal for sensemaking tasks [12, 31]. Consequently, significant research in HCI and visualization has explored the design and creation of graphical representations. The recent advancement of Natural Language Processing (NLP) has further enabled the automatic generation of graphical content such as visualizations [51], 3D scenes [14], animations [35], and videos [26].

For example, systems have been developed to generate visualizations and link existing ones with natural language descriptions to assist the communication and comprehension of data insights [3, 25, 40, 48]. Techniques have been proposed to automatically generate videos from natural descriptions [60], structured markdown documents [21], or informal conversations [73] to create a visual consumption experience without significant manual effort.

Closely related is work that explores the generation of node-link diagrams based on the text from a variety of sources such as video transcripts [33, 44, 64], documents [23], and social media data [36]. For example, More et al. leveraged NLP to generate Unified Modeling Language (UML) diagrams from natural language specifications to facilitate the analysis of software systems [49]. ConceptGuide compiles the transcripts from multiple YouTube videos of a certain topic and then constructs a concept map revealing the various relationships between the videos in order to ease the video-based learning process [44, 64].

Unlike previous work that generates diagrams using existing static text content, this work explores the interaction with node-link diagrams generated from the dynamic text from LLMs. We explore promoting strategies that enable real-time construction of and interaction with the diagrams to facilitate the comprehension of the information provided by LLMs.

2.3 Multilevel Abstraction and Visualization of Information

Extensive research in HCI and Visualization has investigated interaction and visualization techniques that allow users to quickly grasp an overview of complex information while maintaining access to detailed, low-level information or system functionality.

In the field of information visualization, the 'focus + context' design principle states that users need both detail and overview to make sense of information [13]. Information of interest and importance should be displayed in detail, while relevant context should be presented simultaneously to show how that information is connected to the context.

Bederson and Hollan introduced semantic zooming, which displays information at varying levels of detail corresponding to the scale within a zoomable user interface [6]. Norman et al. proposed the concept of progressive disclosure, suggesting that interfaces should progressively inform users about a system by gradually providing pieces of information that contribute to the overall understanding [52]. Xia et al. examined the flexible representational transformation, enabling users to adjust content representations semantically, structurally, and temporally according to their needs, rather than conforming to a single representation imposed by the user interface [72]. This concept was later applied to a program visualization system, allowing programmers to visually inspect program behaviors at different levels of scope and abstraction [34]. Victor explored how varying levels of abstraction over data, procedures, and iteration could help explain complex system behaviors [66].

Graphologue builds upon these prior works to ensure the graphical diagrams are easy to understand by enabling users to flexibly control the levels of detail of the diagrams and synchronizing them with the original text which provides the full context.

3 FORMATIVE STUDY

We conducted a formative study aiming to uncover the prevailing experiences and challenges associated with the use of conversational interfaces to interact with LLMs. The results of this study informed the design choices we made for Graphologue.

3.1 Participants and Procedure

Ten participants with a variety of ChatGPT experiences were recruited, including 2 first-time users, 6 casual users who are familiar with ChatGPT, and 2 experienced users who use it daily with advanced prompting techniques and have developed applications using OpenAI's API. The study sessions were conducted over Zoom for an hour each, and participants received 15 USD as compensation.

Participants completed a pre-task survey that collected demographic information and their experience with ChatGPT. They were then asked to select one topic (from Neuro-divergence, Supply and Demand, Northern Lights, and Inflation) to explore using ChatGPT. Participants were provided with a task description document containing questions related to the chosen topic, which were designed to help them broadly and deeply explore the topic. Participants were given 30 minutes to explore the topic and then interviewed to reflect on their experiences with a focus on the usability pain points of using ChatGPT to obtain, manage, and understand information. Participants were also encouraged to share functionality that they thought would help circumvent the issues they had encountered. The interviews were recorded, transcribed, and analyzed using the reflexive thematic analysis method [9].

3.2 Findings and Discussion

All participants explored the concepts and questions mentioned in their assigned task descriptions, engaging in an average of nine conversational exchanges with ChatGPT. We present the key themes of the **C**hallenges that emerged from the interviews.

C1. Response Content is Verbose and Lacks Structure. Participants expressed concerns over ChatGPT's explanations and commented that *"it was definitely very easy to get overwhelmed by information thrown at [them] sometimes"* (P3). Even if the questions were intentionally framed with a specific scope for short responses, ChatGPT provided long answers (P5). If participants sensed that ChatGPT was generating a redundant response with extraneous background information, they tended to click 'Stop generating' button (P3). When participants asked a follow-up question, many made similar comments that *"ChatGPT tends to repeat the whole thing, and [they] would have to skim over some stuff that [they] already know"* (P7). It was clear that *"ChatGPT was trying to be as exhaustive as possible answering [their] questions"* (P3).

Regarding the format of information presented, P3 found bullet-point content was easier to understand. P5 commented that *"there's not really a visual hierarchy in the text,"* making it difficult to navigate the large amount of text. Participants suggested functionalities to circumvent these issues, including being able to see *"different formats"* (P3), having information management capabilities, like collapse (P6), and shortening parts to prevent repetition (P5).

C2. Lack of Flexible Interaction with the Response Text. All participants extracted some parts of the ChatGPT responses and tried to query them as a prompt to explore them further. However, they expressed the desire to interact with the response directly rather than through a series of conversations. For example, P2 wished they could highlight a part of the ChatGPT response, and directly ask *"Oh, what does this line mean?"*. If participants were unsure of certain parts of the response, they had to manually write it down, or copy-paste from the original text, and prompt each question one by one to get clarification, which is time-consuming and *"increases [their] mental load"* (P9).

C3. Lack of Organization Across Multiple Responses. Keeping track of the various questions and answers previously encountered was found challenging for all participants, as P1 noted that they *"struggled a little bit to like remember everything."* This was further exacerbated by *"redundant answers [that] lack organization"* and form a single stream of questions and answers (P3).

Almost all participants wanted the ability to organize the information collected during the multiple back-and-forth exchanges. Due to a lack of organization, P5 had to *"scroll through a lot"* to find relevant information in a previous response, and suggested a bookmarking technique that would enable them to *"annotate stuff that you want to go back to later."* P2 suggested providing an overview of what information has been explored with bullet points. Alternatively, P2 and P4 recommended organizing the responses spatially in a mind-map form that visualizes *"the connections or the relations between the questions [they] ask"*, which will be *"very useful in terms of understanding the whole topic"* (P2).

3.3 Summary

The formative study reveals that the participants found several challenges with respect to the quantity, organization, and presentation of, and interaction with the ChatGPT responses. They found the responses to be verbose, making it difficult to track, process, and comprehend. The linear conversation contributes to the disorganization of the information embedded in a series of exchanges with ChatGPT. Additionally, there is no direct interactive control over textual responses, which makes it hard for users to specify their intent in follow-up prompts. Overall, the findings indicate that a better representation of information is needed to enable intuitive understanding and flexible exploration of information from LLMs.

4 DESIGN GOALS AND RATIONALE

Based on findings from the formative study and the iterative prototyping and evaluation process, we derive four design goals for a diagram-oriented interaction with LLMs. We first describe the **D**esign goals, and then a scenario (Section 5) to ground how these goals can lead to fluid interaction with LLM-generated information.

D1. Diagram as Entry Point. Our goal was to use diagrams to facilitate the comprehension of LLM responses. However, we found first presenting text from LLMs, and then displaying the diagrams generated from the text, increases rather than decreases the cognitive effort. This is because LLMs, especially GPT-4, take significant time to generate a full response. Users read the response as it is being generated, and the full diagrams presented after the text become new information a user needs to process. Moreover, the diagrams may not match users' mental picture, requiring more time to process the difference. Therefore, we propose generating diagrams *concurrently* with the text and ensuring diagrams are the entry point to the LLM-generated information to aid comprehension.

D2. Flexible Control of Diagram Complexity. Diagrams can be difficult to understand if they are overly complicated. Therefore, care must be taken to manage the complexity of the diagram. The perceived complexity of a diagram can come from two sources: the amount of information in the original text and the presentation mechanism of the diagrams. To avoid overwhelming the users with complexity, the users should be able to flexibly control the amount of information to be visualized in the diagram and how the available information should be revealed.

D3. Diagram-Based Exploration. By utilizing the diagrams as the main interface with LLMs, typical information tasks should be supported through the interaction with the diagrams. Users should be able to acquire more information from the diagrams such as further exploring an unfamiliar concept by requesting more explanations or examples. Similarly, users should be able to trim parts of the diagrams if they are irrelevant to a user's goal.

D4. Synchronized Interaction Between Diagrams and Text. From our informal user testings during system development, we found users often refer back to the original text for two reasons. Firstly, although node-link diagrams help users quickly grasp the main concepts and connections, they may need to consult the original text for details about specific concepts or relationships they find intriguing or challenging to understand from the diagrams. Secondly, since we use GPT-4 to identify entities and relationships for diagram construction, occasional recognition errors can result in inaccurate visualizations. In these cases, users rely on the original text to verify their understanding and to correct misconceptions. Therefore, we propose that the text and diagrams remain synchronized, allowing users to easily locate relevant text from the diagrams and vice versa to leverage the combined strengths of different representations and ensure an efficient information-processing experience.

5 ENVISIONED SCENARIO

We describe a scenario that illustrates the workflow of using Graphologue, a system that we have implemented to support the above design goals for exploratory information seeking.

While working on a deadline, Margaret felt a tremor and confirmed a 4.5 magnitude earthquake online. Living in an earthquake zone but lacking knowledge about them, she wanted to use ChatGPT, a tool that she had been using lately, to learn more about earthquakes. However, weary from hours of writing, she preferred a quicker way to understand the topic. She recently heard about Graphologue, a tool that converts LLM text responses into diagrams for easier comprehension, and decided to give it a shot.

In Graphologue, she started by typing 'What is an earthquake?'. As response text streamed into the interface from the LLM, she noticed a node-link diagram was being constructed piece by piece on the side simultaneously (**D1**). By glancing at the diagram, she quickly understood that 'tectonic plates' - ('shift along') - 'faulty line' and - ('generates') - 'seismic waves.' As a scientist, she wanted to ensure the logical relationship was correct. When she pointed to 'seismic waves,' she noticed the corresponding text was highlighted, allowing her to easily refer to the original text for full details (**D4**).

In the next diagram, she saw 'Richter Scale,' which measures the intensity of earthquakes, and wanted to know the earthquakes that measured 'Magnitude 7.' She clicked on the 'Magnitude 7' node and then the 'Examples' button, and noticed more connections and nodes forming from the node, listing three prior earthquakes, including the '2010 Haiti earthquake' (**D3**). Seeing that, she recalled the devastating damages of that earthquake that she saw on TV.

Each paragraph and its corresponding diagram explain one particular aspect of the earthquake, allowing her to gain information about the earthquake piece by piece (**D2**). After she went through all these diagrams, she wanted to know how these concepts are related together as a whole. She switched to the "merged diagram" view, and immediately she saw all the smaller diagrams animate and combine into one diagram showing the complete picture (**D2**). She then said to herself, "This is groundbreaking."

6 PROMPTING FOR DIAGRAM GENERATION

A primary design goal for Graphologue is to have diagrams as the entry point for people to receive information from LLMs (D1). To this end, the common prompt chaining strategy [70] is inadequate—as getting the original response from the LLM, and further processing it in additional rounds can lead to a long wait time, and the users would have to read the original text response. To mitigate the waiting time, and allow the diagrams to be constructed simultaneously as the response streams in, we iteratively develop our prompts to have the LLM annotate the entities and relationships inline with the tokens for the initial response. This enables Graphologue to provide both the text responses and the diagrams simultaneously.

We develop and test the following prompting strategies with OpenAI's GPT-4, the most advanced and publicly available LLM to date. A full list of original prompts can be found in Appendix A.

6.1 Diagram Construction

We outline our key prompt components, which work together to instruct GPT-4 to generate an initial response that facilitates dynamic diagram construction (D1) and enables easy control over the complexity of presented information (D2) through interactive diagrams (Appendix A.1).

6.1.1 Dividing responses into paragraphs. Smaller and more manageable diagrams, based on portions of the response, effectively prevent users from being overwhelmed by an excessive number of nodes and connections in a single diagram derived from the full response. To achieve this, we instruct GPT-4 to structure its initial response into separate paragraphs, each focusing on a single theme, aspect, or topic, and corresponding with one diagram (Figure 3.b). Interacting with separate diagrams allows users to navigate through different sections of the response more easily, resulting in reduced information overload and improved comprehension.

6.1.2 Annotating Entities. GPT-4 is instructed to annotate entities in the paragraphs to serve as *nodes* in the node-link diagrams. While entity labeling and co-reference resolution are classic tasks in NLP, traditional NLP techniques struggle to perform well without complete sentences [55]. However, we found GPT-4 excels in annotating tokens as entities *simultaneously during their initial generation*, as shown in Figure 2.

We instruct GPT-4 to assign a unique identifier (such as `$N1` in `[Artificial Intelligence (AI) ($N1)]`) for each entity referring to the same concept, i.e., co-reference, to enable entities to be associated by the relationships. GPT-4 has surprisingly superior co-reference capabilities and manages identifiers automatically, avoiding repetition and ensuring consistent labeling throughout the response. As shown in the example paragraph A in Appendix A.1, `Artificial Intelligence (AI)`, `AI systems`, and `It` were all co-referenced and identified as `$N1`.

6.1.3 Annotating Relationships and Saliency. In addition to entities, we prompt GPT-4 to identify and annotate relationships between these entities inline (Figure 2). These relationships serve as the *links* in the node-link diagram. A single relationship described in the text may encompass multiple connections between different pairs of entities, and GPT-4 is instructed to include them in a single annotation, which we utilize to organize the connections together when rendering them on the canvas (like "such as" in Figure 3.c). GPT-4 demonstrates an impressive ability to identify nearly all relationships and associate the corresponding entities with annotations.

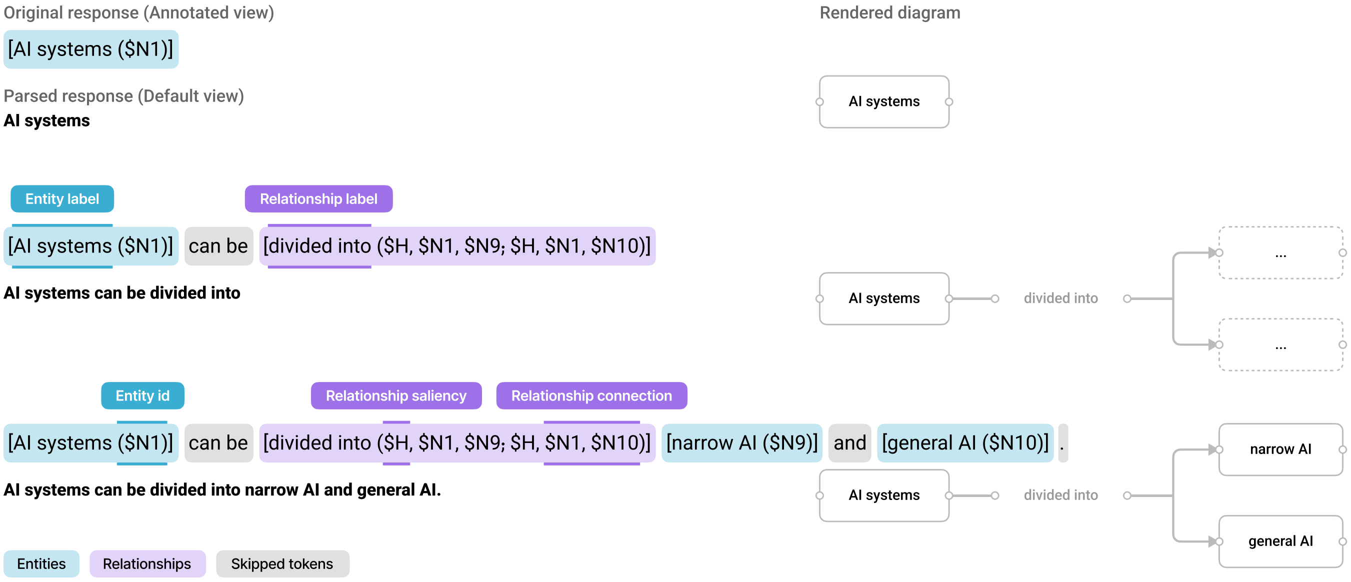


Figure 2: As GPT-4 responses stream in, Graphologue parses them in real-time, removes inline annotations for the interface, extracts entities and relationships, and constructs the corresponding diagrams.

However, annotating all relationships can result in the inclusion of less important ones in the text, which may hinder users from grasping the main idea and nature of the original response. Consequently, this may lead to information overload and clutter when rendered as a diagram. To avoid this, we leverage saliency filters to manage the complexity of the rendered diagram. We instruct GPT-4 to annotate each relationship pair with saliency levels for the connections, designated as either *high* ($H) or *low* ($L). By default, we render the diagram with only the high-saliency relationships to mitigate clutter, while the user can adjust the saliency level control on the interface to show all relationships.

One intriguing aspect of relationship annotations is GPT-4's ability to *prospectively* annotate the involved entities. Even before an entity has been mentioned and labeled in the response, the connecting relationship annotation already incorporates its identifier (Figure 2). This allows us to add new nodes to the existing diagram structure before the corresponding node entities have finished streaming in. Instead, a placeholder node will appear to help draw the user's attention, enabling a more fine-grained and responsive unveiling of the dynamically constructed diagrams.

6.2 Error Prevention and Correction

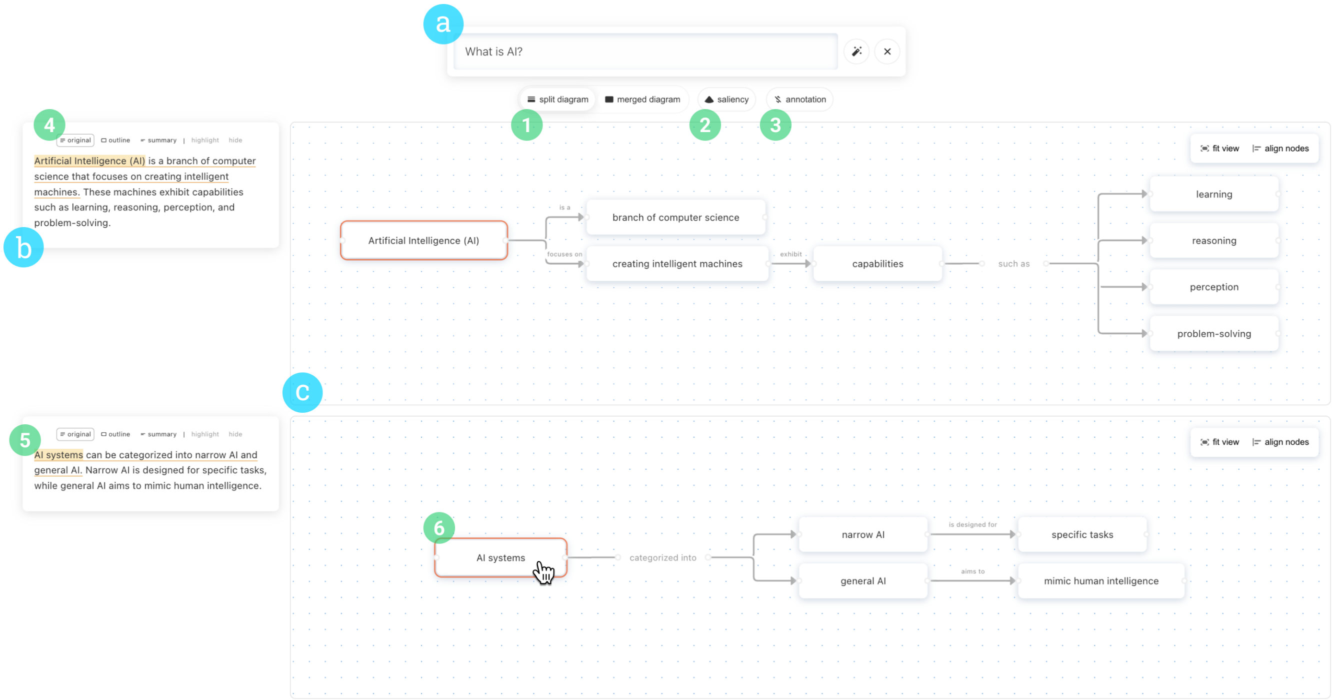
GPT-4 demonstrates powerful inline annotation capabilities. Nevertheless, it is not perfect and occasionally makes mistakes. We develop our prompts and interactions of Graphologue to prevent and correct the errors in GPT-4 annotations.

6.2.1 Avoiding Recurring Problems. As we develop the prompting rules mentioned above, we identify several common errors made by GPT-4, which resulted in impaired response time, redundant information, non-parsable responses, or incorrect diagrams. These problems include assistant-style responses (e.g., "Sure, I can help you with it..."), inconsistent annotation formats (e.g., missing square brackets), misidentifying conjunctive adverbs as entities (e.g., "therefore" or "since then"), and repeating tokens inside and outside of the annotation (e.g., `narrow AI [narrow AI ($N9)]`). We iteratively tune our prompts to explicitly avoid these behaviors and instruct GPT-4 to respond concisely and consistently.

6.2.2 Self-Correction with Additional Rounds of Processing. Two other types of annotation errors in recognizing entities and relationships are addressed through additional rounds of prompting: *dead-end relationships*, which involve annotating relationships that connect non-existent entities; and *orphan entities*, which include annotating entities that are not involved in any relationships within the rest of the response. These mistakes result in empty nodes (nodes with no labels, showing "..." instead) and orphan nodes (nodes disconnected from the rest of the diagram), which hinder the user's ability to follow and digest the diagram for information comprehension. To address these issues, we identify such errors upon completing each paragraph and prompt GPT-4 to self-correct these inconsistencies, asking for a corrected version of each sentence that contains errors one by one.

In the self-correction prompt, the previously generated paragraph serves as context. We pinpoint the specific sentence requiring correction and dynamically describe the issue, e.g., "entities labeled $N11 and $N12 are mentioned but lack connecting relationships." We instruct GPT-4 to re-annotate the sentence or slightly rewrite it if needed for better annotation.

The correction process is done in parallel, and other interactions with the diagram remain unblocked. Once an updated annotation is complete, the diagram is adjusted and animated to reflect the changes. We found that GPT-4 is able to improve the annotation and correct many errors when asked again with issues directly pointed out [59], and we allow users to modify the graph further as needed, which is to be introduced in Section 7.3.3.



Artificial Intelligence (AI) is a branch of computer science that focuses on creating intelligent machines. These machines exhibit capabilities such as learning, reasoning, perception, and problem-solving.

AI systems can be categorized into narrow AI and general AI. Narrow AI is designed for specific tasks, while general AI aims to mimic human intelligence.

Figure 3: The Graphologue interface, including the question input box (a), text response blocks (b), and the diagrams (c). The user can switch between the default split-diagram and merged-diagram views (1), modify the saliency filter (to show all or only high saliency ones) (2), toggle showing the raw GPT-4 response with annotations or the parsed text (3), and change the text block display between Original, Outline, and Summary (4). When a user hovers on a node in the diagram (6), all co-referenced nodes and their corresponding text tokens are highlighted (5).

6.3 Information Quantity Adjustment

We limit the initial response from GPT-4 to fewer than four paragraphs, each containing around 2–3 sentences. This would allow the users to use simple diagrams as the starting point of the interactive exploration process without being overwhelmed by complex diagrams. From the initial diagrams, however, the amount of information can be further adjusted, resulting in response text and diagrams of varying lengths and levels of complexity, allowing users to flexibly reduce the amount of information or delve into more detailed and elaborate content as desired (D2).

6.3.1 Summarizing Each Paragraph of the Response. Saliency filtering helps prevent overwhelming users with excessive information, but sometimes they may want to quickly grasp the most important idea of each paragraph for a brief understanding of the response. To enable this, we prompt GPT-4 to generate a short, one-sentence summary for each paragraph immediately after it is completed. This captures the key concept, and the corresponding diagram is simplified to include only 3–5 nodes, allowing for a concise and easily digestible view. (Figure 4).

Maintaining the identifiers of entities present in the original text and the summary is crucial for enabling the diagram to transit smoothly between different levels of complexity and retaining context for users to digest. To achieve this, we provide GPT-4 with the annotated response when prompting for summaries. This approach ensures that the same entities are matched across different text and diagrams, allowing for synchronized interactions between the diagram and the text, as introduced in Section 7.2.

6.3.2 Asking for More Details. Being able to customize the content based on individual needs and interests facilitates engagement and in-depth understanding at varied levels and scales [10]. We allow users to request more information about the response, including expanding specific paragraphs of the response (Figure 6.b) with additional explanations or examples, as well as introducing new paragraphs and corresponding diagrams that explore additional aspects of the subject matter (Figure 6.c).

7 GRAPHOLOGUE INTERFACE

Graphologue transforms textual responses generated by LLM into interactive diagrams, utilizing inline entity and relationship annotations streamed and interleaved with the original response. In this section, we introduce Graphologue's rich interactions powered by LLM and interface design.

7.1 Constructing the Interactive Diagrams

To enable users to utilize the dynamically constructed node-link diagrams as the entry point of their comprehension process, it is essential to have these diagrams responsively updated as new entities and relationships are identified. To accomplish this, we

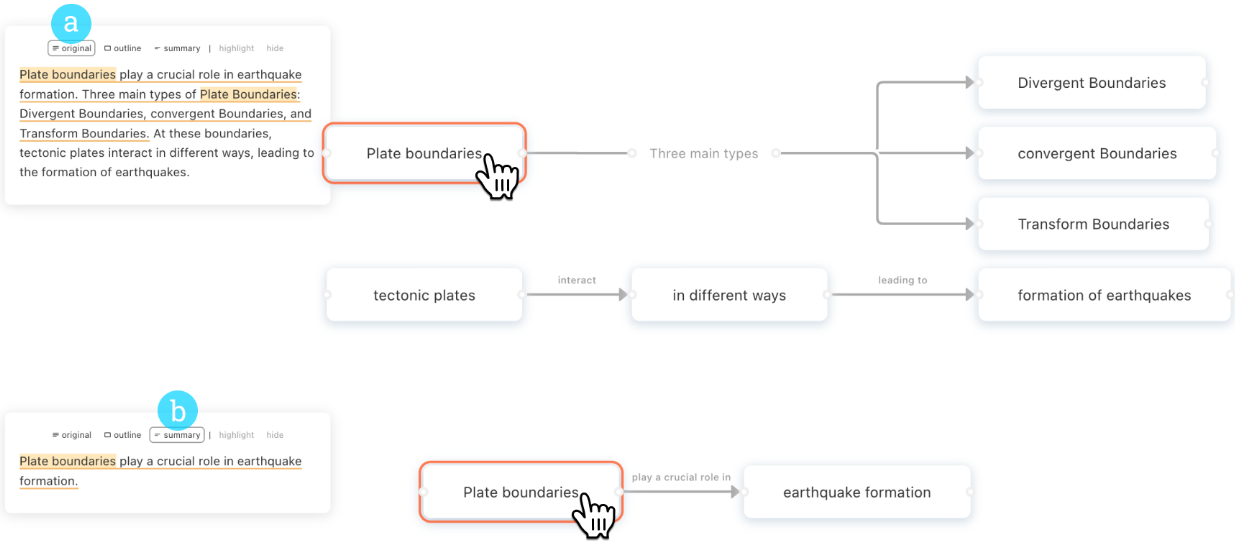


Plate boundaries play a crucial role in earthquake formation. Three main types of Plate Boundaries: Divergent Boundaries, convergent Boundaries, and Transform Boundaries. At these boundaries, tectonic plates interact in different ways, leading to the formation of earthquakes.

Plate boundaries play a crucial role in earthquake formation.

Figure 4: The original response (a) and its summary (b) from GPT-4, with the corresponding diagrams.

parse the text streamed in from GPT-4 and promptly add new nodes or relationships to diagrams as they are completed. This ensures that users have an up-to-date visual representation of the information to facilitate their understanding and exploration of the content.

The diagrams are designed to closely reflect the underlying annotated text. Several design choices are made to achieve this goal. For example, when multiple entities are extracted with the same identifier through co-reference understanding, we use the longest token as the node label in the diagram, as it may contain the most information about the concept. When an annotated relationship describes a connection involving an entity that has not yet been streamed in from the response, we add a placeholder node indicating the incoming entity. Once the actual entity comes in, the placeholder node transforms into a real node (Figure 2), allowing responsive construction of the diagram.

7.2 Bidirectional Synchronization

Graphologue's diagram view serves as a rich and interactive interface for users to comprehend and explore information. However, users may occasionally need to refer back to the original textual response for details or to verify entities and relationships that may have been inaccurately extracted and visualized. To facilitate these processes, entity and relationship annotation information is stored and synchronized across different blocks of text and diagrams. When a user hovers over a node in the diagram, all co-referenced nodes in other individual diagrams are highlighted, as are the corresponding entities in the text and their originating sentences. This allows the user to quickly locate the term in the original response (Figure 3.5). Hovering over the edges in the diagram highlights the relationship tokens in the original response. Additionally, selecting a node highlights itself and connecting edges in the diagram, which in turn highlights the corresponding set of tokens in the response.

On the other hand, as users read the textual response to gain a detailed explanation of their topic of interest, they may switch to the diagram view from time to time to guide their comprehension of the long and unstructured text, for which they need to quickly locate the relevant node in the diagram. To support this, when the mouse cursor is hovering on the text, the corresponding nodes and relationships in the diagram are highlighted to enable quick navigation, as shown in Figure 6.a.

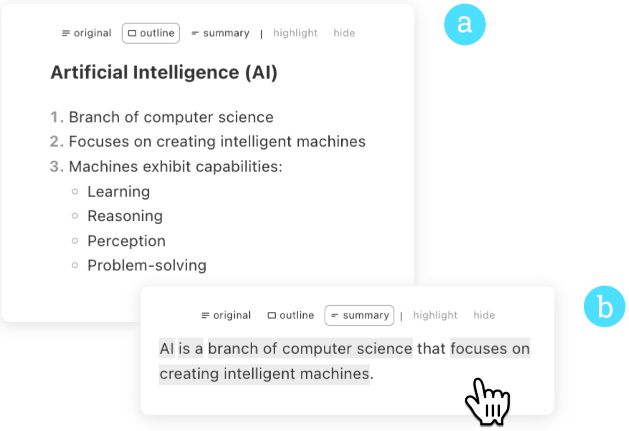
Users can collapse nodes to mitigate information overload as they progressively interact with the diagram (Section 7.3.2). When a node is collapsed, text tokens corresponding to the hidden leaf nodes are greyed out to reduce visual saliency compared to other parts of the response that correspond to active nodes in the diagram. This allows users to focus on important parts of the text and the diagram while minimizing distractions and clutter from less pertinent or uninterested information from both ends.

7.3 Interaction with Diagram Nodes

Graphologue emphasizes using diagrams as the entry point and primary interface to gain information from LLMs, supporting various information tasks to enhance user engagement and understanding, such as exploring unfamiliar concepts by accessing more detailed explanations and examples, and removing irrelevant content to focus on the most pertinent information.

7.3.1 Continuous Exploration Through Explanation and Examples.

As a user receives the initial response from LLM for their query about an unfamiliar concept, they may have follow-up questions as more unfamiliar related concepts would be introduced in the explanation. Structuring the follow-up responses as diagrams and merging them in situ with the existing diagram allows contextual exploration and seamless integration of new information, which helps users understand the new concepts with the help of the existing

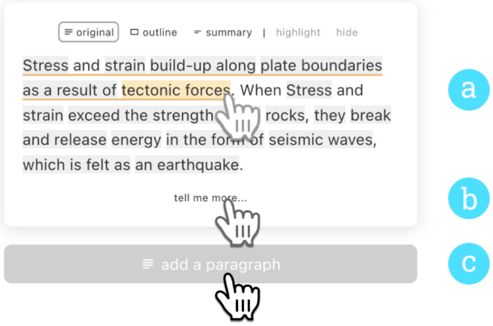


Artificial Intelligence (AI)

1. Branch of computer science
2. Focuses on creating intelligent machines
3. Machines exhibit capabilities:
   - Learning
   - Reasoning
   - Perception
   - Problem-solving

AI is a branch of computer science that focuses on creating intelligent machines.

Figure 5: The Outline (a) and Summary (b) views.



Stress and strain build-up along plate boundaries as a result of tectonic forces. When Stress and strain exceed the strength of rocks, they break and release energy in the form of seismic waves, which is felt as an earthquake.

Figure 6: Hovering on an entity and relationship text token highlights it and others co-referenced, and the corresponding nodes in the diagram (a). Hovering at the bottom of the text block reveals a "tell me more" button, which expands the paragraph (b). The "add a paragraph" bottom after the last text block prompts adding a new paragraph (c).

diagrams. Graphologue thus supports node-oriented exploration on top of the existing diagrams.

When a user encounters an unfamiliar concept in the diagram, they can select it and navigate to the *Explain* or *Examples* menu options to make a follow-up prompt for GPT-4 to generate an explanation or a few examples about the concept (Figure 7.b). The response text is directly appended to the end of the existing paragraph, and newly extracted entities and relationships are either co-referenced and assigned an existing identifier or added to the diagram as new nodes and links accumulatively.

7.3.2 Reduced Information Overload Through Node Collapsing. In addition to gaining more information about a specific node, a user may want to remove the uninterested information or reduce the complexity of the diagrams. To achieve this, Graphologue enables users to collapse and hide all leaf nodes of a diagram node, helping them concentrate on their areas of interest and proceed with their exploration in a less cluttered environment.

7.3.3 User-Initiated Correction Through Trimming and Merging. Even with the powerful GPT-4 and the complex prompts we use, incorrect labeling of entities and relationships still occurs. As the user explores, these errors accumulate and make the diagrams harder to follow and understand. To address this, GPT-4 supports the users to manually correct the errors if they find that necessary.

When a user finds a node is incorrectly extracted, e.g., a conjunctive adverb is identified as an entity, they can select the node and use the *Trim* menu option to remove the entity from the diagram. As a result, relationships associating it with others and their inline annotations in the original response are also removed. When they identify two entities referring to the same concept but are not co-referenced correctly and are rendered as two nodes in the diagram, they can easily merge them by dragging one node on top of the other (to be merged as), after which the connecting relationships and their annotations will also be changed.

7.4 Diagram-Level Information Managing

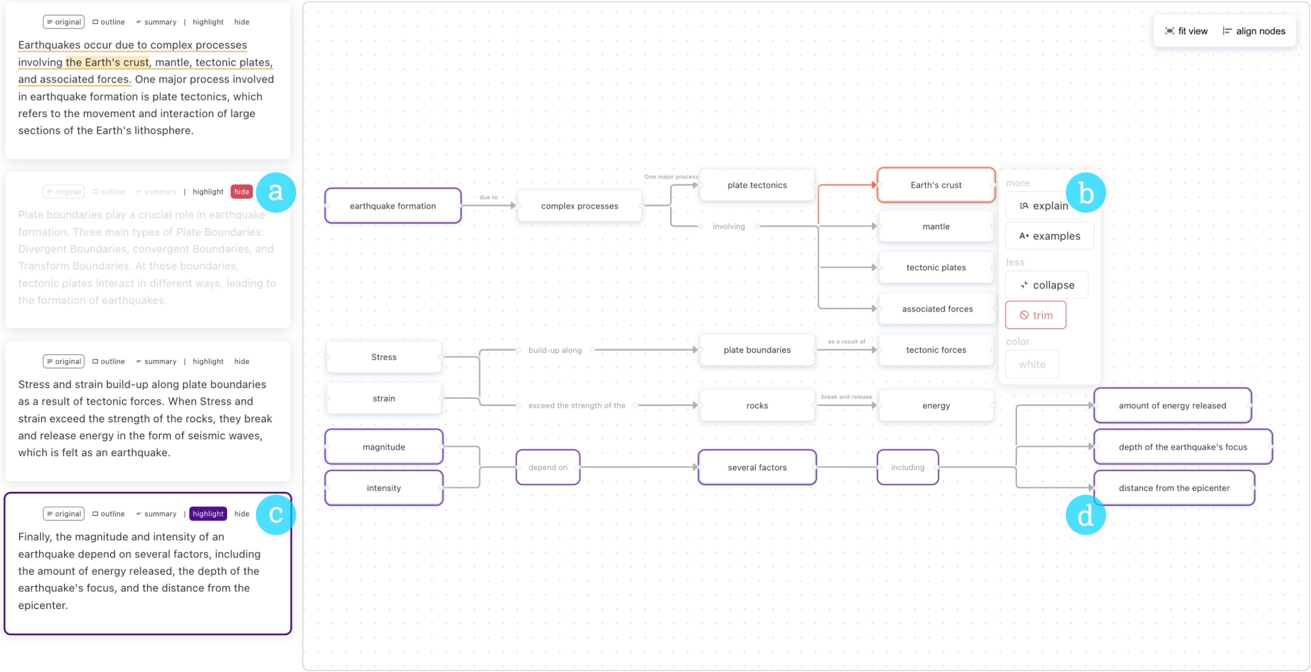
While the diagram representation and a wide range of interactions Graphologue support effectively help users view information from LLMs in a structured and progressive way, a large amount of information easily overwhelms the user eventually with long paragraphs with an exceeding number of entities. As introduced above, several diagram-level controls are introduced to help manage the information display complexity, including dividing responses into paragraphs, relationship saliency, and summary (Figure 4.b). The text blocks can also be switched to the *Outline* view that structures the plain text content with headings and bulleted lists (Figure 5.a). The underlining Markdown text is generated upon completion of each paragraph through prompt chaining, in parallel with self-correction for annotations and summarization (Appendix A.2–A.4).

Conversely, if users find the provided information to be insufficient, they have the option to request additional details for each aspect (Figure 6.b), or request the inclusion of more aspects related to the given topic (Figure 6.c).

7.5 Merging Diagrams

Dividing GPT-4's raw response into different paragraphs and building a separate diagram for each of them effectively reduces information complexity and allows easy navigation and themed exploration. However, this structure is less helpful when users want to understand the relationships between aspects of the answer and get a holistic view of the topic, e.g., how a bird's lightweight body structure and respiratory system work cohesively to enable flight, or how Elon Musk's managing styles differ for his various companies.

Graphologue supports merging individual diagrams into one diagram for examination and integration. The merging and splitting transition is animated to help users gain context for the individual and merged diagrams. When the user hovers over each of the text blocks, the corresponding nodes in the merged diagram get highlighted (Figure 7.c). If the user only wants to examine a subset of paragraphs, they can hide the others from the merged diagram,



Earthquakes occur due to complex processes involving the Earth's crust, mantle, tectonic plates, and associated forces. One major process involved in earthquake formation is plate tectonics, which refers to the movement and interaction of large sections of the Earth's lithosphere.

Plate boundaries play a crucial role in earthquake formation. Three main types of Plate Boundaries: Divergent Boundaries, convergent Boundaries, and Transform Boundaries. At these boundaries, tectonic plates interact in different ways, leading to the formation of earthquakes.

Stress and strain build-up along plate boundaries as a result of tectonic forces. When Stress and strain exceed the strength of the rocks, they break and release energy in the form of seismic waves, which is felt as an earthquake.

Finally, the magnitude and intensity of an earthquake depend on several factors, including the amount of energy released, the depth of the earthquake's focus, and the distance from the epicenter.

Figure 7: A merged diagram with a hidden text block (a), whose extracted entities and relationships are removed from the diagram, and a highlighted text block (c), whose entity nodes are highlighted in the diagram (d). With either merged or split diagrams, a user can select any node in a diagram and select further exploration options in the popup menu (b).

allowing any combination of diagrams to be merged and viewed (Figure 7.a). After the part gets hidden, Graphologue automatically updates and animates the layout of the rest of the diagram to ensure an optimized and efficient view for the users.

8 TECHNICAL EVALUATION

To gain a preliminary understanding of GPT-4's inline entity and relationship annotation ability, we conducted a small-scale technical evaluation to assess the accuracy of GPT-4 inline annotations.

8.1 Setup

The goal of the evaluation is to assess the performance of our prompts with GPT-4 in terms of initial inline entity and relationship annotations, as well as subsequent corrections. To achieve this, we simulated interactions with the system by utilizing GPT-4 to respond to a wide range of topics, following our prompting strategy. We then manually examined all the entity and relationship annotations in both the initial and corrected responses separately, to gauge the system's performance.

8.1.1 Topics. We developed a corpus with GPT-4 using a two-step method. Initially, we requested GPT-4 to generate a list of 50 areas encapsulating various facets of human knowledge space (e.g., history, psychology, engineering, etc.). Subsequently, we asked it to provide a specific topic within each of these areas (e.g., The history of botanical gardens, Gamification). Following this, we amalgamated each topic with the Graphologue's prompt (Appendix A.1) to collect the annotated responses.

8.1.2 Initial Responses. We manually examined all the extracted entity phrases and relationships, identifying errors that appeared in these responses (column Before Correction in Table 1). We found that, given a sentence, GPT-4 could produce different annotations at different levels of granularity. For example, with this short phrase *"the goal of the evaluation,"* GPT-4 could produce two different annotations, such as `[the goal ($N1)] [of ($L, $N1, $N2)] [the evaluation ($N2)]` or `[the goal of the evaluation ($N1)]`. When examining the annotations, as long as they were semantically correct, regardless of the granularity, we accepted them as correct.

In response to all 50 queries, GPT-4 cumulatively returned responses encompassing 4360 words, 1086 annotated entity phrases, and 718 annotated relationships.

8.1.3 Responses After One Round of Correction. Graphologue is capable of detecting Orphan Nodes (i.e., entities without any associated relationships, usually resulting from incorrect entities or incomplete relationships) or Dead-end Links (i.e., relationships attempting to connect non-existent entities, typically due to dead-end relationships), as listed in rows Detectable Errors (errors that can be detected by Graphologue) in Table 1. GPT-4 can conduct a round of corrections using a correction prompt with GPT-4 (Appendix A.2).

We amalgamated the corrected responses with responses that didn't include these errors. The total dataset contained 4382 words

Table 1: Technical Evaluation Results[1]

		Before Correction	%	**After One Round of Correction**	%
	Word Count	4360		4382	
Node Annotation	Total Entity Phrases	1091		1103	
	Total Extracted Entity Phrases	1086		1106	
	Correct Extracted Entity Phrases	1043		1074	
	Erroneous Extracted Entity Phrases	43		32	
Performance	**Precision**		**96.04%**		**97.11%**
	Recall		**95.60%**		**97.37%**
Error Types	Missing Entity Phrase	48	4.40%	29	2.63%
	Incomplete Entity	22	2.02%	11	1.00%
	Incorrect Entity	12	1.10%	13	1.18%
	Incorrect Co-reference	9	0.82%	8	0.73%
Relationship Annotation	Total Relationships	825		843	
	Total Extracted Relationships	718		813	
	Correct Extracted Relationships	654		765	
	Erroneous Extracted Relationships	64		48	
Performance	**Precision**		**91.09%**		**94.10%**
	Recall		**79.27%**		**90.75%**
Error Types	Missing Relationship	171	20.73%	78	9.25%
	Incomplete Relationship	8	0.97%	7	0.83%
	Dead-end Relationship	37	4.48%	27	3.20%
	Reversed Relationship	12	1.45%	10	1.19%
	Incorrect Relationship	7	0.85%	4	0.47%
Detectable Errors	**Orphan Nodes**	133	**12.25%**	38	**3.44%**
	Dead-end Links	55	**7.66%**	23	**2.83%**

(the system is instructed to rewrite the sentence if necessary), 1106 annotated entity phrases, and 813 annotated relationships. We carefully analyzed all the extracted entity phrases and relationships, pinpointing errors that emerged in these responses (column After One Round of Correction in Table 1).

8.2 Key Findings

We consolidate our primary observations pertaining to the initial annotating performance and the improvements observed after a single round of correction.

8.2.1 Initial Annotation Performance (Before Correction). In entity annotation, we observe a precision of 96.04% and a recall of 95.60%. For relationship annotation, we note a precision of 91.09% and a recall of 79.27%. The majority of errors in entity annotation arise due to Missing Entity Phrases (n = 48) and Incomplete Entities (n = 22). Conversely, the bulk of errors in relationship annotations result from Missing Relationships (n = 171) and Dead-end Relationships (n = 37). The annotated responses include 133 Orphan Nodes (attributable to Incorrect Entities or Incorrect Relationships) and 55 Dead-end Links (resulting from Dead-end Relationships).

8.2.2 Annotation Performance with One Round of Correction. Upon correction, the precision for entity annotation improves to 97.11% (increases 1.07%), and the recall rises to 97.37% (increases 1.77%). For relationship annotations, we notice a precision of 94.10% (increases 3.01%) and a recall of 90.75% (increases 11.48%). The numbers of Orphan Nodes and Dead-end Links drop to 38 and 23, respectively. Most errors in entity annotations stem from Missing and Incorrect Entities. Similarly, for relationship annotations, most errors originate from Missing and Dead-end Relationships.

8.3 Summary

In summary, the technical evaluation results suggest that the annotation and correction prompts employed by Graphologue reliably execute entity and relationship annotation tasks. Specifically, they achieve 97.11% precision and 97.37% recall for entity annotation, and 94.10% precision and 90.75% recall for relationship annotation, with one round of self-correction.

9 USER EVALUATION

The primary goal of this project is to investigate the potential benefits of leveraging graphical representations for textual information from LLMs. This approach aims to address many of their limitations and enhance user interactions, especially during exploratory information-seeking tasks. In order to assess the efficacy of this novel approach, we conducted a user evaluation.

[1] Detailed explanation and examples of the error types can be found in Appendix B.

9.1 Participants

We recruited seven experienced users of ChatGPT to evaluate Graphologue. To ensure their level of experience, we requested participants to share screenshots of their ChatGPT history, enabling us to gauge the complexity of the prompts they typically used. Each participant was compensated with 30 USD for one hour of their participation.

9.2 Setup

We conducted all sessions remotely using Zoom, with Graphologue deployed on a cloud server for participants to access. We recorded the audio and screen activity during each session, subsequently transcribing these records to facilitate inductive qualitative analysis using the thematic analysis method [9].

9.3 Procedure

The study session consisted of four steps.

9.3.1 Introduction (5 minutes). The interviewer briefly introduced the goal of the project and study and presented four potential topics for exploration: Neuro-divergence, Supply and Demand, Northern Lights, and Inflation. The participant was then asked to select one of these subjects of interest.

9.3.2 System Training (10 minutes). The interviewer proceeded with a guided tour of all key features of Graphologue, with a standard example topic—Electric Vehicles. This step aimed to familiarize participants with Graphologue interface and functionalities.

9.3.3 System Interaction (20 minutes). Participants were guided to use Graphologue in the completion of their tasks. Specifically, they were tasked to gather information to prepare for a hypothetical lecture. They were given the following prompt: *"Imagine you are a professor at a public university, and you are scheduled to deliver a lecture on the topic of [selected topic]. Here is a list of concepts (three related concepts and one question were provided for each topic) that you aim to explain to your students by the end of the lecture."*

9.3.4 Survey and Interview (20 minutes). Following the completion of the tasks, participants were asked to respond to a survey. The survey consisted of questions designed to gauge their perceptions of the system's usefulness and the utility of the multiple representations of information that it offers. Each question was rated on a 5-point Likert scale, where 1 indicated Strong Disagreement and 5 represented Strong Agreement. Later, we conducted an interview wherein participants were prompted to draw comparisons between their current experience with Graphologue and their prior interactions with ChatGPT in the context of knowledge acquisition tasks. The purpose was to identify how our system addresses the challenges that typically arise when using ChatGPT.

9.4 Results

We share the results of our user evaluation, detailing the effectiveness of Graphologue in facilitating the understanding of information, managing the complexity of diagrams, and the distinct workflows when interacting with typical LLMs and Graphologue. We also discuss the identified limitations of the current system.

9.4.1 Graphologue Facilitates Information Comprehension. Participants found that the node-link diagrams enhanced their understanding of the diverse relationships inherent in the topic they explored (4 Strongly Agree, 3 Agree). For instance, P5 stated that the diagrams helped *"visualize the connections,"* and P4 suggested they *"aid in comprehending how different aspects connect."* Participants particularly praised the diagrams for providing an *"overall view of a topic"* and for *"understanding a set of instructions"* (P8), likening it to a *"mind map"* (P8), and stating that it provides an understanding of the *"organization of information"* (P1). Conversely, they noted that conventional text responses from LLMs can be *"wordy,"* and the graphs facilitate faster information consumption (P1).

The bidirectional mapping between the diagram and paragraph through highlighting *"is a good visual queue to get the users' attention to understand the various relationships"* (P5), which helped them locate the various terms in the explanation from the diagram conveniently (4 Strongly Agree, 1 Agree, 1 Neutral, and 1 Disagree).

9.4.2 Graphologue Provides Sufficient Control for Diagram Complexity. Most participants found the amount of information presented in the responses to be concise (3 Strongly Agree, 2 Agree, and 2 Neutral), and they appreciated the ability to control the level of detail they wished to see with Graphologue (3 Strongly Agree, 1 Agree, and 3 Neutral). For instance, the presentation of smaller diagrams for each individual paragraph was praised for its facilitation of understanding the content within a single paragraph (P7). P1 expressed satisfaction over the degree of control they had over the depth and breadth of information within each paragraph.

Participants responded favorably to the ability to split and merge diagrams, recognizing that this functionality well supports diverse information consumption goals (5 Strongly Agree, 2 Agree). P1 highlighted the utility of split diagrams when focusing on a specific paragraph, while appreciating the merged diagram when an overarching view of the topic was required. P2 suggested that the capacity to flexibly switch between these views could assist students in self-learning tasks, maintaining focus on both critical details and the broader picture. P5 highlighted the value of the merged view as it offers insights beyond *"standalone concepts."* For example, when exploring neuro-diversity, P7 used split diagrams to understand lower-level concepts in detail, such as strategies to identify neurodivergence in individuals. Then, they used the merged diagram to get an overview of how different strategies for identification can be integrated. The ability to highlight parts enabled them to examine how individual sections contributed to and were integrated within the larger diagram (P1, P2, P5).

9.4.3 Graphologue Reduces Prompting Effort and Facilitates Exploration. While using ChatGPT, which often generates verbose information, the burden is typically on the user to craft prompts that control the length of the responses, for instance, 'use less than 200 words.' With Graphologue, however, participants found the extensive controls for managing complexity effectively reduced the need for such directive prompts. As P1 noted, *"there's no need to [specify] 'make it brief,' cause you can just click on [the interface]."*

Interactions with the diagrams to extract new information from the LLM were considered convenient by the participants (5 Strongly Agree, 2 Neutral). Graphologue particularly eased the prompt construction of examples and explanations during learning activities, which can often be monotonous and time-consuming. As P1 noted, with ChatGPT, users have to prompt *"Please incorporate more examples about 'this thing' in your answer."* However, with Graphologue, acquiring context-specific examples is straightforward: *"clicking the 'Examples' button just does it, and you don't have to think about prompting"* (P5). Extending this point, P8 stated that *"following a chain"* becomes simple without the need to write a prompt. P4, for example, intended to understand the meaning of 'particles' in *"charged particles from the sun."* They selected the node 'particles' in the diagram and clicked the 'Explain' button. This action extended the paragraph and constructed a new part of the diagram stemming from the node, serving to clarify the context-specific meaning of 'particles.' P4 proceeded to follow this process for many other terms emerged in the explanation, such as protons and atomic nuclei.

9.4.4 Graphologue Combines the Strengths of Multiple Representations. While the generated diagrams serve as the primary representation of information, Graphologue also provides the original text, an outline, and a summary for each paragraph. Participants found these alternative representations to have complementary strengths. For instance, the outline view, formatted with headings and lists, effectively organized and highlighted the key ideas in each paragraph (6 Strongly Agree, 1 Agree), like *"a traditional lecture slide"* that significantly aids learning (P7). Conversely, the diagrams offset the limitations of the outline view, which simply enumerates different points without showing their interconnections. The diagrams, however, clearly illustrate *"how the bulleted points interact with each other, or how they relate back to the main idea"* (P4). P8 explored the concept of 'marginal utility' and found themselves trying to synthesize a variety of encompassed economics jargon such as 'total utility,' 'tax policies,' and 'welfare programs.' They switched the text block to the outline view, which, coupled with the diagram, provided hierarchical and relational information detailing how these terms function within the concept of 'marginal utility.'

Participants also found that the original text could fill in the details missing from the diagrams. For instance, P1 enjoyed highlighting specific parts of the text they were interested in to get more details, which then formed the basis when seeking additional information. P5 reported that at times, it can be *"a little intimidating if you just see the whole diagram,"* and the synchronization between the outline view and the diagram allowed them to use the outline view as a way to filter out unnecessary nodes in the diagrams. These examples not only illustrate the strengths and weaknesses of each representation, but also show that participants could adaptively employ, switch, and combine them for different use cases.

9.4.5 Limitations. Graphologue separates the response into smaller paragraphs to reduce the complexity of the diagrams. However, this can lead to extensive use of screen space, and P5 suggested that they need to do *"a lot of scrolling"* to view all the diagrams. While more complex diagrams can be space-efficient, they compromise the ease of understanding. To address this inherent trade-off between diagram complexity and spatial efficiency, in addition to merging all small diagrams into one diagram, future iterations of the system need to provide more control for users to flexibly merge any selection of diagrams.

Some participants reported difficulties in understanding the diagrams when they did not match their mental models or included an overwhelming amount of details. On the other hand, while GPT-4's text responses tend to be verbose, unlike diagrams, the text representation does not immediately impose a mental structure, and allows users to gradually construct their own mental models while reading the text. Graphologue alleviates this problem by providing both the generated diagrams and the original text. This issue can be further mitigated by enabling users to manipulate the diagrams during generation to align the diagrams with their mental models.

Other limitations associated with the current interface design and the annotation performance were also identified. For example, P1 found the animations used to progressively augment the diagrams distracting. Some participants found the latency of generation, largely due to GPT-4's performance, negatively impacted the experience. These limitations can be addressed by employing smoother animation effects, prompt engineering to minimize annotation errors, and more responsive generative models.

9.4.6 Summary. The study findings demonstrate that the prompting techniques and the interface designs leveraged by Graphologue offer a more direct representation of the concepts and their relationships, enhancing the comprehension of information from LLMs. The rich and flexible control over the complexity of the diagrams, along with the combination of the strengths of various representations provided by Graphologue, enabled participants to maintain control over the amount of information they wished to consume for diverse information-seeking goals and tasks.

10 DISCUSSION

Our user study findings highlight benefits, limitations, and opportunities of Graphologue, which employs graphical representation and enhanced interaction with LLM-generated information. We discuss these aspects in detail below.

10.1 Improving Annotation Performance

While our technical evaluation demonstrates the advanced inline annotation capabilities of GPT-4, especially with self-correction, the final annotations still contain errors, such as relationships that point to non-existent entities. These annotation errors can produce misleading diagrams, for which the users need to cross-reference the original text to alleviate their confusion. Many approaches can be utilized to improve the performance. For example, incorporating domain knowledge as references for corresponding annotation tasks. Improved prompt designs informed by new knowledge of LLMs could also help improve the overall performance, e.g., assigning roles to the model for different domain-specific tasks or providing more training examples. A more advanced and fine-tuned model could also lead to improved annotation performance.

10.2 Prompting LLMs via Graphical Interfaces

The graphical user interface of Graphologue enables users to employ direct manipulation with the diagram to request explanations and examples from LLMs, saving users' efforts to manually craft textual prompts. Future work could provide more options for interacting with LLMs graphically. For example, the system can allow users to select multiple disconnected nodes and build a new diagram illustrating their connections, or summarize a branch of a diagram with one higher-level concept. Moreover, a text input box can be provided to allow customized requests. As the user explores the knowledge space through the graphical interface, their prior actions and the current diagram can be leveraged as context to construct prompts for LLMs to get responses that are better aligned with the user's needs. For instance, when a user collapses a branch, the subsequent prompts can incorporate text that indicates the collapsed aspect is less relevant to the user's present goal.

10.3 Applications Across Diverse Domains

The node-link diagrams constructed by Graphologue, utilizing LLM responses, are particularly helpful for tasks requiring a comprehensive understanding of diverse concepts and the interconnections among them, such as exploratory information seeking. Participants from our user study identified several immediate applications of Graphologue, including education and professional training. Future research could explore the application of Graphologue's real-time diagram generation capabilities to other contexts where node-link diagrams serve as a powerful visual facilitator. For instance, these diagrams could be used to visualize character relationships in fictional writing or to depict connections among academic papers in literature. In the context of group discussion, diagrams can be generated based on conversations to visualize the discussed concepts and their connections, acting as a collective knowledge map to ground and facilitate the discussions.

10.4 Beyond Node-Link Diagrams

A theme surfaced in the user study is that although node-link diagrams can serve as a suitable representation for understanding interconnected concepts and relationships within LLM responses, they might not always be optimal for a wide range of information tasks. Other formats, such as tables, storyboards, animations, and flowcharts, can be more suitable representations for different aspects of the information. For instance, a table can be clearer when comparing different aspects of multiple concepts, and animations can better illustrate dynamic processes. Future research could investigate how to annotate and construct such representation formats and intelligently select the most suitable one according to the context. Other systems could also explore creating connections among these representations and offering ways for users to switch between them as needed and desired.

11 CONCLUSION

LLMs such as GPT-4 have swiftly gained recognition and popularity due to their unprecedented intelligence and potential for a wide range of applications. Existing interfaces for LLMs, like ChatGPT, employ linear and text-based interfaces, often generating an abundance of information. Our formative study identified three challenges related to the limited usability, readability, and interactivity of textual LLM responses. Graphologue, in contrast, leverages dynamic and interactive diagrams as the primary interface for interacting with LLMs. Our user study indicated that Graphologue effectively addressed many limitations inherent in conversational interfaces by offering flexible graphical representations that facilitated direct and adaptable graphical dialogue with LLMs.

ACKNOWLEDGMENTS

This work would not be possible without the selfless and heartwarming support from all the members of the Creativity Lab at UC San Diego. Our deepest gratitude extends to Fuling Sun, who was crucial in facilitating a productive drive toward the successful completion of this project. We would also like to thank Sangho Suh, Bryan Min, Matthew Beaudouin-Lafon, and Jane E for helping with the video figure production, William Duan, Vidya Madhavan, Tony Meng, Xiaoshuo Yao, and Juliet (Lingye) Zhuang for assistance with the technical evaluation, as well as Brian Hempel and Devamardeep Hayatpur for proofreading the paper draft.

REFERENCES

[1] Maneesh Agrawala, Wilmot Li, and Floraine Berthouzoz. 2011. Design principles for visual communication. *Commun. ACM* 54, 4 (2011), 60–69.

[2] Shaaron Ainsworth and Andrea Th Loizou. 2003. The effects of self-explaining when learning with text or diagrams. *Cognitive science* 27, 4 (2003), 669–681.

[3] Sriram Karthik Badam, Zhicheng Liu, and Niklas Elmqvist. 2018. Elastic documents: Coupling text and tables through contextual visualizations for enhanced document reading. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 661–671.

[4] David Baidoo-Anu and Leticia Owusu Ansah. 2023. Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *Available at SSRN 4337484* (2023).

[5] Jeff Baker, Donald Jones, and Jim Burkman. 2009. Using visual representations of data to enhance sensemaking in data exploration tasks. *Journal of the Association for Information Systems* 10, 7 (2009), 2.

[6] Benjamin B. Bederson and James D. Hollan. 1994. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (Marina del Rey, California, USA) *(UIST '94)*. Association for Computing Machinery, New York, NY, USA, 17–26. https://doi.org/10.1145/192426.192435

[7] Richard A. Bolt. 1980. "Put-That-There": Voice and Gesture at the Graphics Interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques* (Seattle, Washington, USA) *(SIGGRAPH '80)*. Association for Computing Machinery, New York, NY, USA, 262–270. https://doi.org/10.1145/800250.807503

[8] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2022. AudioLM: a Language Modeling Approach to Audio Generation. arXiv:2209.03143 [cs.SD]

[9] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis.* American Psychological Association.

[10] Peter Brusilovsky. 2001. Adaptive hypermedia. *User modeling and user-adapted interaction* 11 (2001), 87–110.

[11] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. arXiv:2303.12712 [cs.CL]

[12] Alberto J. Cañas, Roger Carff, Greg Hill, Marco Carvalho, Marco Arguedas, Thomas C. Eskridge, James Lott, and Rodrigo Carvajal. 2005. *Concept Maps: Integrating Knowledge and Information Visualization.* Springer Berlin Heidelberg, Berlin, Heidelberg, 205–219. https://doi.org/10.1007/11510154_11

[13] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman (Eds.). 1999. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[14] Angel Chang, Will Monroe, Manolis Savva, Christopher Potts, and Christopher D. Manning. 2015. Text to 3D Scene Generation with Rich Lexical Grounding. arXiv:1505.06289 [cs.CL]

[15] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. 2023. Muse: Text-To-Image Generation via Masked Generative Transformers. arXiv:2301.00704 [cs.CV]

[16] Kuo-En Chang, Yao-Ting Sung, and Ine-Dai Chen. 2002. The effect of concept mapping to enhance text comprehension and summarization. *The Journal of Experimental Education* 71, 1 (2002), 5–23.

[17] Chaomei Chen. 2010. Information visualization. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, 4 (2010), 387–403.

[18] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail

Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374 [cs.LG]

[19] Zhutian Chen and Haijun Xia. 2022. CrossData: Leveraging Text-Data Connections for Authoring Data Documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 95, 15 pages. https://doi.org/10.1145/3491102.3517485

[20] Peter C.-H. Cheng, Ric K. Lowe, and Mike Scaife. 2001. *Cognitive Science Approaches To Understanding Diagrammatic Representations*. Springer Netherlands, Dordrecht, 79–94. https://doi.org/10.1007/978-94-017-3524-7_5

[21] Peggy Chi, Nathan Frey, Katrina Panovich, and Irfan Essa. 2021. Automatic Instructional Video Creation from a Markdown-Formatted Tutorial. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 677–690. https://doi.org/10.1145/3472749.3474778

[22] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Visual Sketching of Story Generation with Pretrained Language Models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 172, 4 pages. https://doi.org/10.1145/3491101.3519873

[23] Antoine Clarinval, Isabelle Linden, Anne Wallemacq, and Bruno Dumas. 2018. Evoq: A Visualization Tool to Support Structural Analysis of Text Documents. In *Proceedings of the ACM Symposium on Document Engineering 2018* (Halifax, NS, Canada) *(DocEng '18)*. Association for Computing Machinery, New York, NY, USA, Article 27, 10 pages. https://doi.org/10.1145/3209280.3209533

[24] Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. 1997. QuickSet: Multimodal Interaction for Distributed Applications. In *Proceedings of the Fifth ACM International Conference on Multimedia* (Seattle, Washington, USA) *(MULTIMEDIA '97)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/266180.266328

[25] Weiwei Cui, Xiaoyu Zhang, Yun Wang, He Huang, Bei Chen, Lei Fang, Haidong Zhang, Jian-Guan Lou, and Dongmei Zhang. 2019. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 906–916.

[26] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and Content-Guided Video Synthesis with Diffusion Models. arXiv:2302.03011 [cs.CV]

[27] Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S. Bernstein. 2018. Iris: A Conversational Agent for Complex Tasks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174047

[28] Hendijanifard Fatemeh, Kardan Ahmad, and Dibay Moghadam Mohammad. 2011. ICMAP: An interactive tool for concept map generation to facilitate learning process. *Procedia Computer Science* 3 (2011), 524–529.

[29] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G. Karahalios. 2015. DataTone: Managing Ambiguity in Natural Language Interfaces for Data Visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 489–500. https://doi.org/10.1145/2807442.2807478

[30] Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for Science Writing Using Language Models. In *Designing Interactive Systems Conference* (Virtual Event, Australia) *(DIS '22)*. Association for Computing Machinery, New York, NY, USA, 1002–1019. https://doi.org/10.1145/3532106.3533533

[31] Charles Goodwin. 2015. *Professional Vision*. Springer Fachmedien Wiesbaden, Wiesbaden, 387–425. https://doi.org/10.1007/978-3-531-19381-6_20

[32] Stephen J Guastello, Mary Traut, and Gene Korienek. 1989. Verbal versus pictorial representations of objects in a human-computer interface. *International journal of man-machine studies* 31, 1 (1989), 99–120.

[33] Tessai Hayama and Shuma Sato. 2020. Supporting Online Video e-Learning with Semi-automatic Concept-Map Generation. In *Learning and Collaboration Technologies. Designing, Developing and Deploying Learning Experiences*. Springer International Publishing, Cham, 64–76.

[34] Devamardeep Hayatpur, Daniel Wigdor, and Haijun Xia. 2023. CrossCode: Multi-Level Visualization of Program Execution. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 593, 13 pages. https://doi.org/10.1145/3544548.3581390

[35] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. 2022. AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars. arXiv:2205.08535 [cs.CV]

[36] Mengdie Hu, Krist Wongsuphasawat, and John Stasko. 2017. Visualizing Social Media Content with SentenTree. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 621–630. https://doi.org/10.1109/TVCG.2016.2598590

[37] Gwo-Jen Hwang, Po-Han Wu, and Hui-Ru Ke. 2011. An interactive concept map approach to supporting mobile learning activities for natural science courses. *Computers & education* 57, 4 (2011), 2272–2280.

[38] Gwo-Jen Hwang, Li-Hsueh Yang, and Sheng-Yuan Wang. 2013. A concept map-embedded educational computer game for improving students' learning performance in natural science courses. *Computers & Education* 69 (2013), 121–130.

[39] Peiling Jiang, Fuling Sun, and Haijun Xia. 2023. Log-It: Supporting Programming with Interactive, Contextual, Structured, and Visual Logs. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 594, 16 pages. https://doi.org/10.1145/3544548.3581403

[40] Dae Hyun Kim, Enamul Hoque, Juho Kim, and Maneesh Agrawala. 2018. Facilitating Document Reading by Linking Text and Tables. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 423–434. https://doi.org/10.1145/3242587.3242617

[41] Yea-Seul Kim, Mira Dontcheva, Eytan Adar, and Jessica Hullman. 2019. Vocal Shortcuts for Creative Experts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300562

[42] Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. *PLoS digital health* 2, 2 (2023), e0000198.

[43] Jill H Larkin and Herbert A Simon. 1987. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science* 11, 1 (1987), 65–100.

[44] Ching Liu, Juho Kim, and Hao-Chuan Wang. 2018. ConceptScape: Collaborative Concept Mapping for Video Learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173961

[45] Vivian Liu and Lydia B Chilton. 2022. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 384, 23 pages. https://doi.org/10.1145/3491102.3501825

[46] Stephen MacNeil, Andrew Tran, Juho Leinonen, Paul Denny, Joanne Kim, Arto Hellas, Seth Bernstein, and Sami Sarsa. 2023. Automatically Generating CS Learning Materials with Large Language Models. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1176. https://doi.org/10.1145/3545947.3569630

[47] J.H. McClellan, L.D. Harvel, R. Velmurugan, M. Borkar, and C. Scheibe. 2004. CNT: concept-map based navigation and discovery in a repository of learning content. In *34th Annual Frontiers in Education, 2004. FIE 2004*. IEEE, F1F–13. https://doi.org/10.1109/FIE.2004.1408581

[48] Ronald Metoyer, Qiyu Zhi, Bart Janczuk, and Walter Scheirer. 2018. Coupling Story to Visualization: Using Textual Analysis as a Bridge Between Data and Interpretation. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. Association for Computing Machinery, New York, NY, USA, 503–507. https://doi.org/10.1145/3172944.3173007

[49] Priyanka More and Rashmi Phalnikar. 2012. Generating UML diagrams from natural language specifications. *International Journal of Applied Information Systems* 1, 8 (2012), 19–23.

[50] Raphael Moura, Michael Beer, Edoardo Patelli, and John Lewis. 2017. Learning from major accidents: Graphical representation and analysis of multi-attribute events to enhance risk communication. *Safety science* 99 (2017), 58–70.

[51] Arpit Narechania, Arjun Srinivasan, and John Stasko. 2020. NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 369–379.

[52] Donald A. Norman and Stephen W. Draper. 1986. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., USA.

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

[54] Hammond Pearce, Benjamin Tan, Baleegh Ahmad, Ramesh Karri, and Brendan Dolan-Gavitt. 2022. Examining Zero-Shot Vulnerability Repair with Large Language Models. arXiv:2112.02125 [cs.CR]

[55] Jakub Piskorski and Roman Yangarber. 2013. *Information Extraction: Past, Present and Future*. Springer Berlin Heidelberg, Berlin, Heidelberg, 23–49. https://doi.

org/10.1007/978-3-642-28569-1_2

[56] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? arXiv:2302.06476 [cs.CL]

[57] Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. arXiv:2102.07350 [cs.CL]

[58] Michail Schwab, Hendrik Strobelt, James Tompkin, Colin Fredericks, Connor Huff, Dana Higgins, Anton Strezhnev, Mayya Komisarchik, Gary King, and Hanspeter Pfister. 2017. booc.io: An Education System with Hierarchical Concept Maps and Dynamic Non-linear Learning Plans. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 571–580. https://doi.org/10.1109/TVCG.2016.2598518

[59] Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. arXiv:2303.11366 [cs.AI]

[60] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2022. Make-A-Video: Text-to-Video Generation without Text-Video Data. arXiv:2209.14792 [cs.CV]

[61] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. 2022. ProgPrompt: Generating Situated Robot Task Plans using Large Language Models. arXiv:2209.11302 [cs.RO]

[62] Tamara Sumner, Faisal Ahmad, Sonal Bhushan, Qianyi Gu, Francis Molina, Stedman Willard, Michael Wright, Lynne Davis, and Greg Janée. 2005. Linking learning goals and educational resources through interactive concept map visualizations. *International Journal on Digital Libraries* 5 (2005), 18–24.

[63] Ivan E. Sutherland. 1964. Sketch Pad a Man-Machine Graphical Communication System. In *Proceedings of the SHARE Design Automation Workshop (DAC '64)*. Association for Computing Machinery, New York, NY, USA, 6.329–6.346. https://doi.org/10.1145/800265.810742

[64] Chien-Lin Tang, Jingxian Liao, Hao-Chuan Wang, Ching-Ying Sung, and Wen-Chieh Lin. 2021. ConceptGuide: Supporting Online Video Learning with Concept Map-Based Recommendation of Learning Path. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2757–2768. https://doi.org/10.1145/3442381.3449808

[65] Priyan Vaithilingam, Tianyi Zhang, and Elena L. Glassman. 2022. Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 332, 7 pages. https://doi.org/10.1145/3491101.3519665

[66] Bret Victor. 2011. Up and Down the Ladder of Abstraction: A Systematic Approach to Interactive Visualization. http://worrydream.com/LadderOfAbstraction/

[67] Johannes Wheeldon. 2011. Is a Picture Worth a Thousand Words? Using Mind Maps to Facilitate Participant Recall in Qualitative Research. *Qualitative Report* 16, 2 (2011), 509–522.

[68] Terry Winograd et al. 1972. Shrdlu: A system for dialog.

[69] Po-Han Wu, Gwo-Jen Hwang, Marcelo Milrad, Hui-Ru Ke, and Yueh-Min Huang. 2012. An innovative concept map approach for improving students' learning performance with an instant feedback mechanism. *British Journal of Educational Technology* 43, 2 (2012), 217–232.

[70] Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. 2022. PromptChainer: Chaining Large Language Model Prompts through Visual Programming. arXiv:2203.06566 [cs.HC]

[71] Haijun Xia. 2020. Crosspower: Bridging Graphics and Linguistics. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 722–734. https://doi.org/10.1145/3379337.3415845

[72] Haijun Xia, Ken Hinckley, Michel Pahud, Xiao Tu, and Bill Buxton. 2017. WritLarge: Ink Unleashed by Unified Scope, Action, & Zoom. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3227–3240. https://doi.org/10.1145/3025453.3025664

[73] Haijun Xia, Jennifer Jacobs, and Maneesh Agrawala. 2020. Crosscast: Adding Visuals to Audio Travel Podcasts. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 735–746. https://doi.org/10.1145/3379337.3415882

[74] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story Writing With Large Language Models. In *27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 841–852. https://doi.org/10.1145/3490099.3511105

[75] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision* 130, 9 (2022), 2337–2348.

A PROMPTS

We provide all the prompts that we use for the prototype environment of Graphologue with OpenAI GPT-4 API. **System**, **User**, and **Assistant** (i.e. GPT-4) are pre-defined roles for querying the API[2].

A.1 Initial Query

System Please provide a well-structured response to the user's question in multiple paragraphs. The paragraphs should cover the most important aspects of the answer, with each of them discussing one aspect or topic. Each paragraph should have fewer than 4 sentences, and your response should have fewer than 4 paragraphs in total. The user's goal is to construct a concept map to visually explain your response. To achieve this, annotate the key entities and relationships inline for each sentence in the paragraphs.

Entities are usually noun phrases and should be annotated with [entity ($N1)], for example, [Artificial Intelligence ($N1)]. Do not annotate conjunctive adverbs, such as "since then" or "therefore", as entities in the map.

A relationship is usually a word or a phrase that consists of verbs, adjectives, adverbs, or prepositions, e.g., "contribute to", "by", "is", and "such as". Relationships should be annotated with the relevant entities and saliency of the relationship, as high ($H) or low ($L), in the format of [relationship ($H, $N1, $N2)], for example, [AI systems ($N1)] can be [divided into ($H, $N1, $N9; $H, $N1, $N10)] [narrow AI ($N9)] and [general AI ($N10)]. Relationships of high saliency are those included in summaries. Relationships of low saliency are often omitted in summaries. It's important to choose relationships that accurately reflect the nature of the connection between the entities in text, and to use a consistent annotation format throughout the paragraphs.

You should try to annotate at least one relationship for each entity. Relationships should only connect entities that appear in the response. You can arrange the sentences in a way that facilitates the annotation of entities and relationships, but the arrangement should not alter their meaning, and they should still flow naturally in language.

Example paragraph A: [Artificial Intelligence (AI) ($N1)] [is a ($H, $N1, $N2)] [field of computer science ($N2)] that [creates ($H, $N1, $N3)] [intelligent machines ($N3)]. [These machines ($N3)] [possess ($H, $N3, $N4)] [capabilities ($N4)] [such as ($L, $N4, $N5; $L, $N4, $N6; $L, $N4, $N7; $L, $N4, $N8)] [learning ($N5)], [reasoning ($N6)], [perception ($N7)], and [problem-solving ($N8)]. [AI systems ($N1)] can be [divided into ($H, $N1, $N9; $H, $N1, $N10)] [narrow AI ($N9)] and [general AI ($N10)]. [Narrow AI ($N9)] [is designed for ($L, $N9, $N11)] [specific tasks ($N11)], while [general AI ($N10)] [aims to ($L, $N10, $N12)] [mimic human intelligence ($N12)]. [It ($N1)] [has grown across ($H, $N1, $N13)] [multiple industries ($N13)], [leading to ($L, $N1, $N14; $L, $N1, $N15; $L, $N1, $N16)] [improved efficiency ($N14)], [enhanced decision-making ($N15)], and [better user experiences ($N16)].

Example paragraph B: [Human-Computer Interaction ($N1)] [is a ($H, $N1, $N2)] [multidisciplinary field ($N2)] that [focuses on ($H, $N1, $N3)] [the design and use of computer technology ($N3)], [centered around ($H, $N1, $N4)] [the interfaces ($N4)] [between ($H, $N4, $N5; $H, $N4, $N6)] [people (users) ($N5)]

[2] https://platform.openai.com/docs/api-reference/chat

and [computers ($N6)]. [Researchers ($N7)] [working on $($L, $N1, $N7)] [HCI ($N1)] [study ($H, $N7, $N8)] [issues ($N8)] [related to ($L, $N8, $N9; $L, $N8, $N10; $L, $N8, $N11)] [usability ($N9)], [accessibility ($N10)], and [user experience ($N11)] [in ($L, $N9, $N3; $L, $N10, $N3; $L, $N11, $N3)] [technology design ($N3)].

Example paragraph C: [Birds ($N1)] [can ($H, $N1, $N2)] [fly ($N2)] [due to ($H, $N2, $N3)] [a combination of physiological adaptations ($N3)]. [One key ($H, $N3, $N4)] [adaptation ($N4)] [is ($H, $N4, $N5)] the [presence of lightweight bones ($N5)] that [reduce ($H, $N5, $N6)] [their body weight ($N6)], [making ($L, $N5, $N7)] it [easier for them to fly ($N7)]. [Another ($H, $N3, $N8)] [adaptation ($N8)] [is ($H, $N8, $N9)] the [structure of their wings ($N9)] which [are designed for ($H, $N9, $N2)] [flight ($N2)].

Your response should have multiple paragraphs.

User *[The query provided by the user.]*

A.2 Self-Correction

*[The initial prompts and responses from the **System**, **User**, and **Assistant**.]*

System In the following sentence of your original response, there are some issues that need to be fixed.

The entities *[list the annotated entities, separate by commas]* were mentioned but not connected by any relationships. *[Add this paragraph only if orphan nodes were detected.]*

One or more relationships annotated by relationship annotations *[list the annotated relationships, separate by commas]* were trying to connect entities with ids that are not mentioned in the response. *[Add this paragraph only if dead-end relationships were detected.]*

In your corrected response, please make sure that all entities and relationships are extracted correctly. Relationships should only connect existing entities, and entities should be connected by at least one relationship. Please try to fix these issues in your response by annotating the same sentence again. You may arrange the sentences in a way that facilitates the annotation of entities and relationships, but the arrangement should not alter their meaning and they should still flow naturally in language.

When annotating a new entity that was not mentioned in the previous response, please make sure that they are annotated with a new entity id (for example, if the previous annotation has reached id “$N102”, then the new annotation id should start at “$N103”). However, if the same entity has appeared in the original response, please match their id.

Please only include the re-annotated sentence in your response.

A.3 Summary

System You are a professional writer specializing in text summarization. Make a short, one-sentence summary of the chunk of the text provided by the user. The summary should reflect the main idea and the most important relationships of the text. Notice that the user has annotated the text with entities and relationships. Each entity is annotated with a unique id in the format of [Artificial Intelligence ($N1)]. Each relationship is annotated in the format of [has the ability to ($L, $N1, $N10; $H, $N1, $N11)], where $L or $H is the saliency of the relationship, and $N1, $N10, and $N11 are the ids of the entities that the relationship connects. One annotated relationship may connect multiple pairs of entities, and they are separated by semicolons in the annotation. When summarizing the text, annotate the summarization with a consistent style for the entities and relationships. Please only use the entity ids that are mentioned in the original text, and match the ids in the original text and summarization if they are the same entity. Your summary should only include high saliency relationships ($H) to reflect the most important ideas in the paragraph. You can arrange the sentences in the summarization in a way that facilitates the annotation of entities and relationships, but the arrangement should not alter their meaning and they should still flow naturally in language. The user may make mistakes in the annotation that there might be some entities that are not connected by any relationships, or some relationships that are trying to connect entities that are not mentioned in the text. Please avoid these mistakes when annotating the summary. Your summary should have only one short sentence.

Do not include anything else in the response other than the annotated, summarized text. For example, for paragraph: [Human-Computer Interaction ($N1)] [is a ($H, $N1, $N2)] [multidisciplinary field ($N2)] that [focuses on ($H, $N1, $N3)] [the design and use of computer technology ($N3)], [centered around ($H, $N1, $N4)] [the interfaces ($N4)] [between ($H, $N4, $N5; $H, $N4, $N6)] [people (users) ($N5)] and [computers ($N6)]. [Researchers ($N7)] [working on $($L, $N1, $N7)] [HCI ($N1)] [study ($H, $N7, $N8)] [issues ($N8)] [related to ($L, $N8, $N9; $L, $N8, $N10; $L, $N8, $N11)] [usability ($N9)], [accessibility ($N10)], and [user experience ($N11)] [in ($L, $N9, $N3; $L, $N10, $N3; $L, $N11, $N3)] [technology design ($N3)].

You may summarize it as: [HCI ($N1)] [is a ($H, $N1, $N2)] [multidisciplinary field ($N2)] that [centered around ($H, $N1, $N4)] [the interfaces ($N4)] [between ($H, $N4, $N5; $H, $N4, $N6)] [users ($N5)] and [computers ($N6)].

User *[The paragraph to be summarized, from the original response.]*

A.4 Outline

System You are a professional presentation slide builder. Structure the following text provided by the user into a presentation slide, in markdown format. If you need to use a list, use a numbered list. Do not include anything else in the response other than the markdown text.

User *[The paragraph to create the outline, from the original response.]*

A.5 Node Explanation

*[The initial prompts and responses from the **System**, **User**, and **Assistant**.]*

User In the sentence *[the sentence containing the node from the original response]*, you mentioned the entity *[node label]*. Can you explain this entity in 1 to 2 sentences? Please refer to the original response as the context of your explanation. Your explanation should be concise, one paragraph, and follow the same annotation format as the original response. You should try to annotate at least one relationship for each entity. Relationships should only connect entities that appear in the response. When annotating a new entity that was not mentioned in the previous response, please make sure that they are annotated with a new entity id (for example, if the previous annotation has reached id “$N102”, then the new annotation id should start at “$N103”). However, if the same entity has appeared in the original response, please match their id.

For example, for “[general AI ($N10)]” in the sentence “[AI systems ($N1)] can be [divided into ($H, $N1, $N9; $H, $N1, $N10)] [narrow AI ($N9)] and [general AI ($N10)].”: [General AI ($N10)] refers to a [type of ($L, $N1, $N10)] [artificial intelligence ($N1)] that [has the ability to ($L, $N10, $N14; $L, $N10, $N5; $L, $N10, $N15)] [understand ($N14)], [learn ($N5)], and [apply knowledge across a wide range of tasks ($N15)].

A.6 Node Examples

[The initial prompts and responses from the ***System****,* ***User****, and* ***Assistant****.]*

User In the sentence *[the sentence containing the node from the original response]*, you mentioned the entity *[node label]*. Can you give a few examples of it? Your response should follow the same annotation format as the original response, as shown in the following example. When annotating a new entity that was not mentioned in the previous response, please make sure that they are annotated with a new entity id (for example, if the previous annotation has reached id “$N102”, then the new annotation id should start at “$N103”). However, if the same entity has appeared in the original response, please match their id. You don’t need to further explain the examples you give.

For example, for “[Fruits ($N1)]” in the sentence “[Fruits ($N1)] can [help with ($H, $N1, $N2)] [health ($N2)].”, your response could be: “[Fruits ($N1)], for example, [includes ($H, $N1, $N3; $H, $N1, $N4; $H, $N1, $N5)], [apples ($N3)], [oranges ($N4)], and [watermelons ($N5)].”

A.7 Tell Me More

[The initial prompts and responses from the ***System****,* ***User****, and* ***Assistant****.]*

User For the paragraph *[the paragraph to be extended]*, can you continue writing one or two more sentences at the end of the paragraph? When continue writing this paragraph, please refer to the original response as the context of your writing. Your response should be about the same topic and aspect of the original paragraph and could add more details. Your response should follow the same annotation format as the original response. When annotating a new entity that was not mentioned in the previous response, please make sure that they are annotated with a new entity id (for example, if the previous annotation has reached id “$N102”, then the new annotation id should start at “$N103”). However, if the same entity has appeared in the original response, please match their id. Your response should only have the new content.

A.8 Add a Paragraph

[The initial prompts and responses from the ***System****,* ***User****, and* ***Assistant****.]*

User Can you continue writing one paragraph after the end of your original response? When writing the new paragraph, please refer to the original response as the context of your writing. Your response should still try to answer the user’s original question and could add more details or provide a new aspect. Your response should follow the same annotation format as the original response. When annotating a new entity that was not mentioned in the previous response, please make sure that they are annotated with a new entity id (for example, if the previous annotation has reached id “$N102”, then the new annotation id should start at “$N103”). However, if the same entity has appeared in the original response, please match their id. Your response should only have the new content.

B TECHNICAL EVALUATION EXPLANATION AND EXAMPLES

Table 2: Annotation Error Explanation and Examples

Category	Error	Content
Node Annotation Error Types	**Missing Entity Phrase** Example	An entity that is not annotated when should. ... [is driven by ($H, $N8, $N9)] the [need to reduce ($H, $N7, $N10)] [greenhouse gas emissions ($N10)] from **traditional internal combustion engines**.
	Incomplete Entity Example	Entity annotations lacking necessary label words or splitting a single entity phrase. [Impressionist painters ($N1)] [belong to ($H, $N1, $N2)] **[the Impressionism ($N2)] [art movement ($N3)]**.
	Incorrect Entity Example	An entity annotation that includes words or phrases that do not belong to the entity. **[However ($N13)]**, [the Industrial Revolution ($N13)] [led to ($H, $N13, $N14)] [mass production ($N14)] of [clothing ($N3)].
	Incorrect Co-reference Example	Labeling co-referenced entities with different identifiers. **[Apple ($N1)]** [is ($H, $N1, $N2)] [good ($N2)]. **[It ($N3)]** [helps ($H, $N3, $N4)] [health ($N4)]. *(Count as 1 incorrect.)*
Relationship Annotation Error Types	**Missing Relationship** Example	A relationship that is not annotated when should. ... **[establishing ($H, $N1, $N5)]** [political systems ($N5)], [economies ($N6)], and [cultural practices ($N7)]. *(Missing relationships for economies and cultural practices.)*
	Incomplete Relationship Example	A relationship annotation that does not include the whole phrase as the label. [should be ($H, $N12, $N13)] [based on ($H, $N13, $N14)]
	Dead-end Relationship Example	A relationship annotation that includes entity identifiers that do not exist. [These philosophers ($N9)] **[emphasized ($H, $N9, $N13)] [the importance of ($L, $N13, $N14; $L, $N13, $N15)]** [subjectivity ($N14)] and [individual freedom ($N15)].
	Reversed Relationship Example	A relationship that has been annotated with reversed direction. [Apple ($N1)] **[is ($H, $N2, $N1)]** [good ($N2)].
	Incorrect Relationship Example	A relationship that is annotated with the wrong pair of entities. [Gods ($N4)] in [Greek mythology ($N1)] [play ($H, **$N1, $N4**)] [central roles ($N10)] in [the narratives ($N11)]. *(Should be $N4, $N10.)*
Detectable Errors	**Orphan Nodes**	Nodes that are not involved in any relationships, might be caused by Incorrect Entity or Incomplete Relationship. (Percentage divided by Total Extracted Entity Phrase.)
	Dead-end Links	Links that try to connect non-existent nodes, caused by Dead-end Relationship. (Percentage divided by Total Extracted Relationship.)