

IMPLICIT PEN ANNOTATION
ASSISTED BY LARGE LANGUAGE MODELS

BY
BENEDICT LEUNG

*A thesis submitted to the
School of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of*

Master of Science
in
Computer Science

Faculty of Science
University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada
August 2025

Benedict Leung: *Implicit Pen Annotation Assisted by Large Language Models* © August 2025

SUPERVISORS:

Dr. Mariana Shimabukuro

Dr. Christopher Collins

LOCATION:

Oshawa, Ontario, Canada

TIME FRAME:

August 2025

THESIS EXAMINATION INFORMATION

Submitted by: Benedict Leung

Master of Science in Computer Science

<p>Thesis Title: Implicit Pen Annotation Assisted by Large Language Models</p>

An oral defense of this thesis took place on August 6, 2025, in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Richard Pazzi
Research Supervisor	Dr. Christopher Collins
Research Co-supervisor	Dr. Mariana Shimabukuro
Examining Committee Member	Dr. Ali Neshati
Thesis Examiner	Dr. Loutfouz Zaman

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

In our modern society, integrating humans and computer systems has transformed everyday tasks, including reading, annotating, and reviewing documents. Annotating documents is an age-old practice that involves adding markings and notes to engage with the material. Although this process is crucial for understanding texts and facilitating collaborative communication, it has not evolved significantly over the years. Tedious and repetitive workflows in current digital annotation tools suggest missed opportunities for more intelligent, adaptive design. This work introduces *AnnotateGPT*, a document annotation tool with a digital pen. It leverages a Large Language Model (LLM) (1) to infer the underlying purposes of the user's annotations and (2) automatically generates annotations with the same purpose throughout the document. AnnotateGPT aims to alleviate the burdens of manual annotation, enabling users to focus on tasks that require critical expertise.

Keywords: annotation; digital pen; LLM

DECLARATION

I, Benedict Leung, hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

The research work in this thesis that was performed in compliance with the regulations of Research Ethics Board #18136

_____ Benedict Leung

STATEMENT OF CONTRIBUTIONS

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

ACKNOWLEDGMENTS

I am very glad to have this opportunity to do research and would like to thank a few people. First, my parents allowed me to freely learn whatever I wanted growing up, no matter how many failures I had. Although I wasn't always at my best in the past, they still allowed me the freedom to learn and grow in my own way. I hope to be able to give back to you in the future, as you made many sacrifices coming here as an immigrant.

Of course, I wasn't alone in my journey here. I would like to thank Chris Lam, my closest friend. He taught me not only that everyone can aim high at something but also that you should have some joy while doing it. I always see him smiling, radiating joy in everything you do, and everyone can see that too. I would also like to thank Ashley Lu, who has heavily shaped who I am today as a person. Before I met you, I didn't seek the company of others, but you are the first person with whom I want to be in company. From you, I have learned not to settle for "just enough" but to go as far as possible, no matter what. It is shameful that I have not properly thanked you, but let's hope I am not a coward once we meet again.

I want to thank all the people I have met up to this point. Yes, all. I am fortunate to be surrounded by many bright and aspiring individuals. They shaped who I am today, from being a decent human being to what I should aim for. Even if we had one little conversation in the hallway at high school, I was able to remember something from your voice. That was how bright all the people I have met have been and how they have passed on their "sparks" to me. I will continue to strive to keep my flame burning more brightly.

Finally, I would like to thank my supervisors, who are at the core of my research work. Dr. Mariana Shimabukuro was one of the first people I met in the research field. I always thought she was a robot who only does high-quality work, because I had a stereotype that CS researchers always code. But once I got to know her better, the stereotype quickly broke. She prioritizes her students' well-being over her own work. She is always available to answer questions and talk about anything. Lastly, Dr. Christopher Collins, without whom the lab environment would not exist. He creates a high standard in you, and you can tell just from his speaking. I am very glad he gave me the opportunity to join his lab during my undergrad. Although I still don't know why, as I never spoke to him when applying and simply wrote two words, 'eye tracking,' in my application. That is why I feel incredibly lucky to have him as my supervisor. Maybe one day, I can address my supervisors by their first names without using "Dr." as they usually smile at me when I do.

CONTENTS

THESIS EXAMINATION INFORMATION	iii
ABSTRACT	iv
DECLARATION	v
STATEMENT OF CONTRIBUTIONS	vi
ACKNOWLEDGMENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xi
LIST OF ALGORITHMS	xii
ACRONYMS	xiii
GLOSSARY	xiv
1 INTRODUCTION	1
1.1 Scope and Purpose	3
2 DOCUMENT ANNOTATION	5
2.1 Studies of Annotation	5
2.2 Annotation Tools	6
2.2.1 Paper and Digital Systems	6
2.2.2 Classification Mechanisms	7
2.2.3 Processing Annotations	9
2.3 Summary	10
3 LARGE LANGUAGE MODELS FOR ANNOTATION	11
3.1 Prompting Techniques	11
3.2 Human-AI Collaboration with LLMs	12
3.3 Vision	14
3.4 Retrieval Augmented Generation	15
3.5 Summary	16
4 ANNOTATEGPT	17
4.1 Challenges in Providing Feedback to Students	17
4.2 Design Guidelines	18
4.3 Framework	19
4.3.1 Stroke Clustering	21
4.3.2 Stroke Classification and Text Extraction	23
4.3.3 Annotation Classification and Purpose Inference	26
4.3.4 Generating Low-Level Context-Specific Annotations	29
4.4 Implementation	32
4.4.1 Document Parsing	32
4.4.2 Assistant Marker	33
4.4.3 Verifying AnnotateGPT's Annotations	34
4.4.4 Specialized Scrollbar	34
4.5 Summary	35
5 EVALUATION	36
5.1 User Study	36

5.1.1	English Tests Creation	36
5.1.2	Participants	37
5.1.3	Apparatus & Software	37
5.1.4	Study Design	37
5.1.5	Tasks & Procedure	38
5.2	Data Analysis	39
5.3	Results	39
5.4	Summary	52
6	DISCUSSION	53
6.1	Design Implications	53
6.1.1	LLMs as Cognitive Augmentation in Feedback Workflows	53
6.1.2	Support Equity in Education	54
6.1.3	Annotation as an Interaction Paradigm	54
6.2	Example Scenarios	55
6.2.1	Lightweight Annotation for High-Volume Feed- back	55
6.2.2	Leveraging Annotation History Across Subjects	56
6.3	Envisioned Applications	56
6.3.1	Editing Generative Content	58
6.3.2	Modality for Augmented Reality	58
6.4	Limitations & Future Work	58
6.5	Summary	59
7	CONCLUSION	60
A	APPENDIX	61
A.1	Annotation History	61
A.2	Additional Statistical Test Results	62
	BIBLIOGRAPHY	66

LIST OF FIGURES

Figure 1	Venetus A: Manuscript of the Iliad	2
Figure 2	Screenshots of different types of interfaces . .	8
Figure 3	Screenshot of the Metatation interface	9
Figure 4	Few-shot and zero-shot example	12
Figure 5	Few-shot and zero-shot CoT example	13
Figure 6	Multimodal prompt example	15
Figure 7	Interaction overview	17
Figure 8	Framework overview of AnnotateGPT	20
Figure 9	Sample clustering process	22
Figure 10	Classifying hand-drawn underlines/cross-outs	23
Figure 11	Classifying hand-drawn enclosures	24
Figure 12	Sample results for annotation classification and purpose inference	28
Figure 13	Comparing document vs. ChatGPT output . .	31
Figure 14	Prototype interface of AnnotateGPT	32
Figure 15	Assistant marker	33
Figure 16	Verifying AnnotateGPT’s annotations	33
Figure 17	Specialized scrollbar	34
Figure 18	Study setup	37
Figure 19	Example of followup annotations	40
Figure 20	Screenshots of the first page for P1–3	43
Figure 21	Screenshots of the first page for P4–6	44
Figure 22	Annotation density plots	45
Figure 23	Annotating duration plot	46
Figure 24	Heatmaps of each participant’s annotations . .	47
Figure 25	SUS scores plot	48
Figure 26	TLX scores plot	49
Figure 27	Questionnaire responses for AnnotateGPT’s features	50
Figure 28	Questionnaire responses on evaluation experience	50
Figure 29	Plot of the number of annotations with ratings	51
Figure 30	Generative content use cases	57
Figure 31	Augmented reality use cases	57

LIST OF TABLES

Table 1	Mapping from annotation form to function with anchor types	5
Table 2	Prompt for annotation classification and purpose inference	27
Table 3	Prompt for generating annotations	30
Table 4	Types of annotations	41
Table 5	Examples of purpose inferences	42
Table 6	Example feedback by AnnotateGPT	52
Table 7	Annotation history	61
Table 8	ART ANOVA results for strokes per annotation	62
Table 9	ART ANOVA results for number of annotations	62
Table 10	ART ANOVA Results for annotating duration	62
Table 11	<i>p</i> -values for number of annotations for AnnotateGPT	63
Table 12	<i>p</i> -values for number of annotations for baseline	63
Table 13	<i>p</i> -values for number of annotations between AnnotateGPT and baseline	63
Table 14	<i>p</i> -values for AnnotateGPT annotating duration	64
Table 15	<i>p</i> -values for baseline annotating duration . . .	64
Table 16	<i>p</i> -values for annotating duration between AnnotateGPT and baseline	64
Table 17	NASA-TLX scores	65

LIST OF ALGORITHMS

Algorithm 1	Horizontal Line Detection	23
Algorithm 2	Line Segment Intersection	24
Algorithm 3	Enclosure Detection	25

ACRONYMS

AI	Artificial intelligence
BERT	Bidirectional Encoder Representations from Transformers
CoT	Chain-of-Thought
GPT	Generative Pre-trained Transformer
ICL	In-context learning
LLM	Large language model
ML	Machine learning
TLX	NASA Task Load Index
RAG	Retrieval augmented generation
ThoT	Thread of Thought
SUS	System Usability Scale

GLOSSARY

ANNOTATION information associated with a piece of content in the document to explain or add information.

LARGE LANGUAGE MODEL foundation model that is trained on vast amounts of data, enabling them to understand and generate natural language and various types of content to perform a wide range of tasks.

INTRODUCTION

In our modern society, humans, machines, and computer systems often collaborate on various tasks. For example, booking appointments, managing home security, and communicating with others. Recently, artificial intelligence (AI) and large language models (LLMs) have become integral components of our computer systems. This powerful combination enables system designers to accomplish more complex tasks, particularly the generation of human-like reasoning and explanations using natural language, thereby increasing the accessibility of other computing techniques, such as machine learning (ML). This thesis leverages an AI agent, specifically OpenAI's GPT agent, to assist people in reading, annotating, reviewing, taking notes, and providing feedback on lengthy text documents.

There are various methods for engaging with reading materials. One of the primary and traditional approaches is close reading, which involves analyzing the interactions between syntactic, semantic, structural, rhetorical, and phonetic features within a text. On the other hand, distant reading [99], one of the more modern approaches to text analysis, uses computational tools to detect patterns in a text corpus. However, tools supporting only distant reading [1, 20, 28, 82] present information before the readers have done any work. While it provides a broad overview of text collections to support exploring trends across documents, it overlooks the reader's critical reading experience in order to gain a deeper understanding of the text. Making annotations is one of the common ways to engage with the materials to make sense of the text.

Annotation is almost as old as writing. Starting in the 8th Century BCE, the Greeks adopted alphabetic writing, initially considered secondary to singing and speaking [73]. For centuries following its creation, this writing system was vastly underused and even looked down upon by society [73]. Eventually, one of the earliest texts documented is Homer's oral poems, transcribed by Hellenistic scholars [9, 73], whose comments and explanations laid the foundation for the editions we have today, in addition to the advancements in manuscript technology that facilitated the meticulous copying of both the primary texts and their accompanying notes [9] (see Figure 1). To note is "a fundamental act of attention, of sharing, as basic as water or love." To note is an essential human act [9].

Readers annotate as part of their routine to understand and process the material. An annotation consists of information associated with a piece of content in the document to explain or add information [68, 81]. However, annotating documents can be time-consuming, repetitive and tedious. Traditionally, annotations have been made on paper documents, such as textbooks or printed paper [68]. Electronic documents have become increasingly popular, and digital annotation tools have

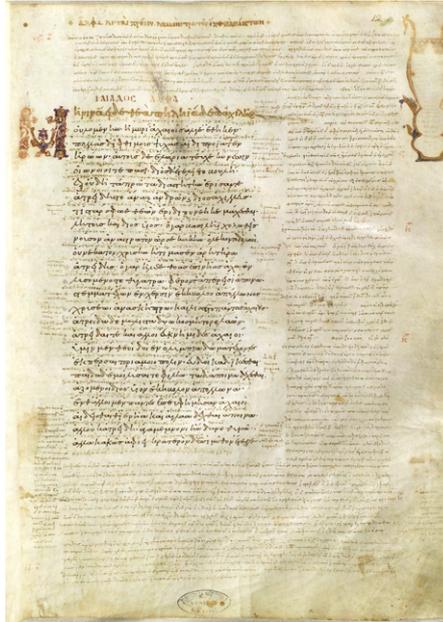


Figure 1: Folio 12r of Venetus A [9], where the bolded text is the poem and all other texts are annotations made by scholars.

been developed to facilitate the process [36]. Using digital annotation tools has enhanced the process by creating new mechanisms for annotation. For example, Adobe Acrobat [2] supports various types of notes, including highlighting and sticky notes. Annotation tools can also use semantic tags (e.g., question and discussion) to classify annotations, such as loomp [43]. It is important to remember that annotating a document is a personalized process. However, some aspects of a user's annotation workflow may follow certain patterns [68, 69]. For example, highlighting is used to signal future attention or aid memory. Although existing tools enable users to create annotations in digital documents, the process can still be burdensome.

Automating the process of annotating is not trivial, as annotations are idiosyncratic and personal to the user, making it difficult to identify patterns and regulations. Prior works aim to study annotation and determine their common characteristics and properties amongst individuals. Studies reveal that, naturally, every user has their own mode of annotating to meet their demands, but the form of the annotations remains consistent [69]. For example, they indicate a specific piece of text by underlining or highlighting it. Several attempts have been made to fasten the readers' process by detecting these common forms of annotations and using annotated words as queries to retrieve relevant information [18, 38, 70, 93]. However, they cannot annotate text within the same document, generalize the domain for which their application can operate or ignore certain forms of annotation (e.g., handwritten notes). This indicates the need for a system that can analyze all annotations and work flexibly within the user's workflow, while also being able to read the document for further annotations that align with the user's personalized annotation style.

The rise of transformer architectures [103] has enabled LLMs to have incredible language and vision capabilities and has been integrated into applications for processing various interactions [117]. The most common types of LLMs are Bidirectional Encoder Representations from Transformers (BERT) [29] and Generative Pre-trained Transformer (GPT) [16]. The BERT series typically requires training with a large dataset. However, this is not feasible in our case because there are no annotation datasets, and datasets also limit the domain of documents and can suffer from outdated knowledge. More promisingly, the GPT series are generative models that involve translating tasks into natural language to generate outputs without datasets. While working with text annotations along with the pattern-finding abilities of GPT and its ability to structure semantic inferences, it is possible to condition GPT by inputting instructions that describe the task to solve various tasks, also known as in-context learning and prompting techniques [64]. For example, OpenAI's GPT-4(V) [79, 115] is one of the state-of-the-art LLMs with robust language and vision capabilities trained on publicly available data (e.g., internet data) and performs well in text generation. This thesis explores leveraging GPT-4(V) for annotating documents by implicitly determining the purposes of annotations and further refining them. The goal is to enhance human abilities through user interaction and user experience design. The resulting tool, *AnnotateGPT*, enables human-AI collaboration in document annotation, resulting in high-quality annotations and better user experiences. This approach enables users to focus on tasks that require human expertise, such as applying domain knowledge, giving feedback, and conducting reviews.

1.1 SCOPE AND PURPOSE

According to Wolfe [112], annotation tools are diverse and can include a variety of models. It includes input methods such as keyboards, mice, digital pens, and voice recognition. Additionally, annotation tools can have different types of interfaces, such as hyperlinks that open in a separate window, highlighting, freehand marks directly on the text, interlinear text and markings inserted into the body of the primary text, columns aligned with the primary document, or animated callouts. Annotation tools can be used on various documents, including standalone text documents, web pages, sound files, videos, emails, and other types of annotations. Finally, annotation tools can have different anchors, such as a specific string or passage of text, a particular paragraph, an entire document, or an absolute pixel position.

This thesis focused on free-form annotation, as it is a priority to integrate annotation with reading smoothly [78]. Paper is considered an excellent medium for close reading, as it is easy to carry, versatile, easily shareable, and supports various activities, such as annotation. However, most keyboard and mouse interfaces for online annotation are often cumbersome, requiring users to select a command, point with

a mouse, and type on the keyboard. Hand-held reading devices that allow pen-based freehand input can replicate the ease and functionality of working with paper. Thus, AnnotateGPT uses digital ink to create freehand marks directly on a pen-based device, closely mimicking the functionality of paper. This thesis also focuses on stand-alone text documents that are widely used (e.g., PDFs).

As discussed, previous approaches do not further annotate within the same document or have a limited domain of documents to which they can be applied. One way to tackle both challenges is to use the general knowledge of LLMs through in-context learning or prompting techniques. However, LLMs have a limitation on the maximum number of input tokens they can handle and may not perform well when dealing with long texts [63, 128]. Thus, placing the entire document in the prompts or a significant portion of the document may not be practical and could impact performance. Retrieval augmented generation (RAG) addressed this issue, enabling the LLM to have an additional knowledge base in addition to its general knowledge base, by injecting the document contents into the LLM [131]. The last challenge is to identify patterns and nuances that are personal to an individual's editorial style. In AnnotateGPT, it leverages the vision capabilities of LLMs and RAG, which can be used to make inferences about the purposes of annotations and then inject correct inferences for future inferences. This leads to the following research questions, which this thesis addresses:

RQ1: How effectively can an LLM infer the purposes behind document annotations?

RQ2: How effectively can an LLM be used for comprehensive and continuous document annotation?

Human-computer interaction (HCI) focuses on designing with users in mind, emphasizing the importance of considering users' perceptions and thoughts when evaluating a system. To address these research questions, a user study was conducted to compare AnnotateGPT with a baseline (i.e., without assistance). The following two chapters will provide background information and a review of past works on annotation studies (Chapter 2) and large language models within the context of annotations (Chapter 3). The system design of AnnotateGPT is then described, detailing how LLMs are leveraged for annotation (Chapter 4). The study design for the user study is then outlined, along with the analysis of the results (Chapter 5). The next chapter then discussed design implications and possible future directions (Chapter 6). Finally, the thesis concludes with the main takeaways (Chapter 7).

DOCUMENT ANNOTATION

Annotating a document is a process that engages with reading material. It involves making markings associated with a piece of content to better understand the document. Annotations are specific to their process alone (idiosyncratic) and do not consistently serve the same purpose for an individual (polymorphic) by nature [38, 70]. Thus, studies have been conducted to better understand the purpose of these annotations.

2.1 STUDIES OF ANNOTATION

One of the well-known studies is Marshall's, in which she investigated annotations made by students in various textbooks [68]. The author provided observations of common annotation forms and determined their function. Later, Marshall and Brush observed the importance of the spatial arrangement of annotations [69]. They also provide a mapping of forms of annotations to their function, as shown in Table 1, along with their anchor types.

Form	Function
Within-text	
Underlining or highlighting higher level structure; cross-outs	Signal future attention
Short highlightings; circled words or phrases	Aiding memory
Words/phrases between lines	Interpretation
Extended highlighting or underlining	Trace progress through narrative
Marginal or blank space	
Asterisks	Signal future attention or aiding memory
Notation near figures or equations	Problem-solving
Notes, doodles, drawings	Interpretation or reflect material circumstances of reading

Table 1: Mapping from annotation form to function along with their anchor types (within-text and marginal or blank space) adapted from Marshall's work [68].

Annotations serve various purposes; they act as procedural signals, aiding text navigation by marking important content. Annotations also aid memory, helping readers retain crucial information by highlighting or underlining key points. Additionally, they facilitate problem-solving by allowing students to work through problems within the text in real-time. These annotations also record interpretive activity, reflecting readers' insights and reflections on the text's language, structure, and themes [68]. They further serve as visible traces of the reader's attention, particularly in challenging texts, and may reflect external circumstances.

Overall, annotations enhance the reading experience by guiding readers, aiding memory, facilitating engagement, and providing insights into readers' interaction with the text and their surroundings.

Mehta et al. [70] also investigated annotations made in the domain of poetry but also arrived at the same conclusion, noticing the need to support idiosyncratic annotations and the need to minimize distraction from the reading task [68, 70]. Based on these works and observations, Ovsianikov et al. [81] built upon Marshall’s initial research to formulate guidelines for an improved digital annotation system. They conducted an empirical study on paper annotations to gain a deeper understanding of the characteristics of an ideal annotation system. Their findings indicated that the most prevalent annotation method involved marking text sections with a marker, highlighting key terms, and making notes in the margins. Upon examining the specific annotations made by participants, it was observed that markings made with ink and highlighters facilitated quick text location for users. Concurrently, margin notes kept related thoughts and ideas close to the relevant text content. This followed Textension [13], a system which exemplifies this balance by supporting margin notes and line-level expansions while preserving the reading experience. The authors emphasized that pen interfaces would be more favoured if they provided extra features and capabilities. This insight complements Marshall’s work, suggesting a need for a delicate balance of introducing new features and capabilities while minimizing the cognitive effort required.

2.2 ANNOTATION TOOLS

Originally, annotations were made on paper. Digital annotation tools and digital documents have been developed to facilitate annotating. There are many different models of annotation tools, each offering many features and mechanisms to make the annotation process easier. Typical contexts for annotation include the following [81]:

- SELF-DIRECTED: Made by readers for personal use, like students studying or researchers noting references.
- AUTHOR-DIRECTED: Made by readers like teachers to share with the author, often for works in progress.
- READER-SHARED: Made by readers to share with others, like comments in library books, annotated editions or community highlights on Kobo eReaders.
- AUTHOR-SHARED: Made by authors to share with readers, like T. S. Eliot’s annotations on *The Wasteland*.

2.2.1 *Paper and Digital Systems*

Existing annotation tools address one of these contexts, tailoring their design features to fit the specific context. A critical consideration before designing is the choice of medium for the annotations. In a study conducted in 1999, O’Hara and Sellen [78] established that paper outperformed traditional vertical computer setups in facilitating annotations due to the enhanced affordances offered by paper. Ten years

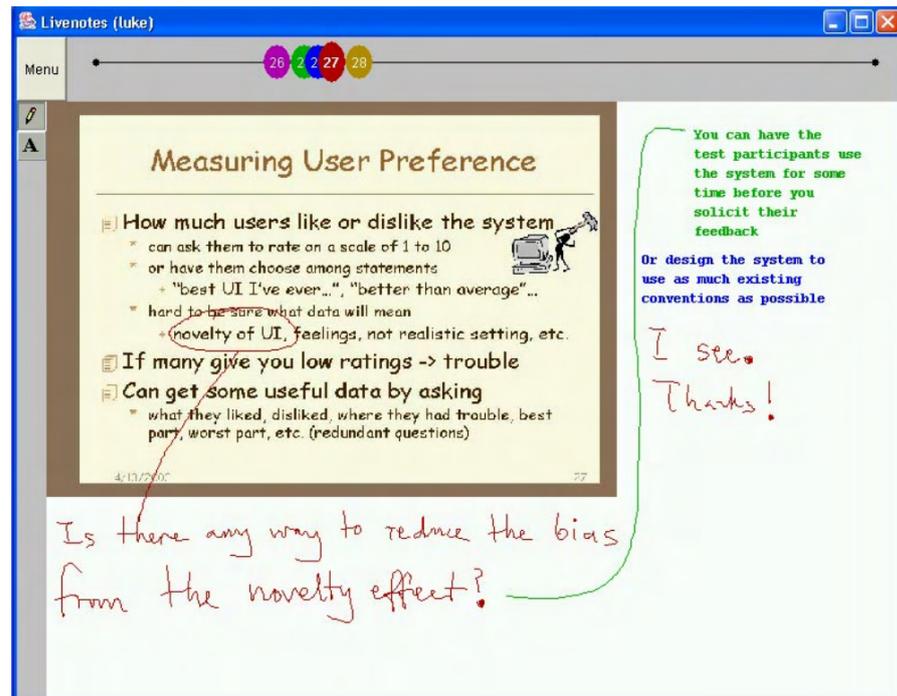
later, a follow-up study was conducted to also include tablets and horizontal setups. It was found that vertical setups were still perceived poorly, but tablets and horizontal setups provided a comparable annotation experience to paper [72]. Thus, previous works have used fully digital or hybrid mediums that coupled paper with digital systems to support digital document annotation. A hybrid medium will have greater interactivity and new ways of working with information [39]. This medium is implemented by enhancing paper as an input device and extending its output capabilities to a digital device. For example, IllumiPaper [51] integrates thin-film displays onto paper, enabling the paper to serve as a direct-touch display. However, enabling this medium requires additional hardware, such as specialized digital pens, vision or motion capture technologies, embedded sensing electronics, projectors, or custom-built devices [39]. Therefore, our work will focus on fully digital systems because they do not require additional hardware.

Digital annotation tools offer the advantage of facilitating collaboration and sharing annotated content [66, 101]. One way to implement fully digital systems is by using digital ink to support exclusively handwritten annotations [98]. For example, LiveNotes [48] supports annotating instructor slides by hand, placing digital ink (see Figure 2a). Alternatively, annotation tools offer highlighting, underlining, and adding sticky notes to classify annotations [112]. These digital features are standard in today's annotation interfaces, such as Adobe Acrobat [2].

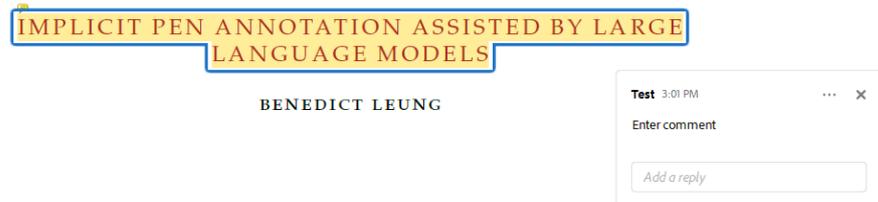
2.2.2 Classification Mechanisms

Transitioning to the digital domain allows us to predetermine tags and emphasize and classify annotations based on their style or semantic meaning. For example, Adobe Acrobat [2] (see Figure 2b) supports various annotation styles, such as highlighting, underlining, and sticky notes on PDFs, each presenting the annotation differently. MyNote [22] also presents a set of predefined tags but instead adds semantic meaning behind the annotations (normal, question, answer, and discussion). However, these predefined semantic tags are fixed and can vary across different annotation tools [12, 19, 46, 96], lacking flexibility and may not cover all scenarios.

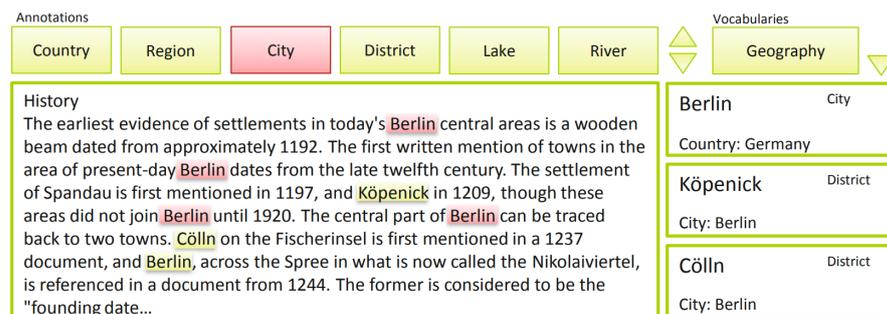
Instead of having fixed tags, annotation tools can opt for having the annotator define the tags themselves. For example, HyLighter [56] and loomp [43] (see Figure 2c) both provide the same annotation features as Adobe Acrobat, with the addition of organizing these annotations with user-defined tags. These vocabularies have a significant advantage because they are well-suited for annotation needs when created by skilled annotation experts to fit specific annotation processes [36]. However, updating or modifying this vocabulary to accommodate new concepts or changes can be time-consuming and require the continuous involvement of the annotators.



(a)



(b)



(c)

Figure 2: Screenshots of different types of interfaces where (a) is from LiveNotes [48] which supports freehand digital ink on instructor slides (fair use) (b) is from Adobe Acrobat [2] which supports highlighting, sticky notes, etc. and (c) is from loomp [43] which supports user-defined tags to classify annotations based on semantics (used with permission).

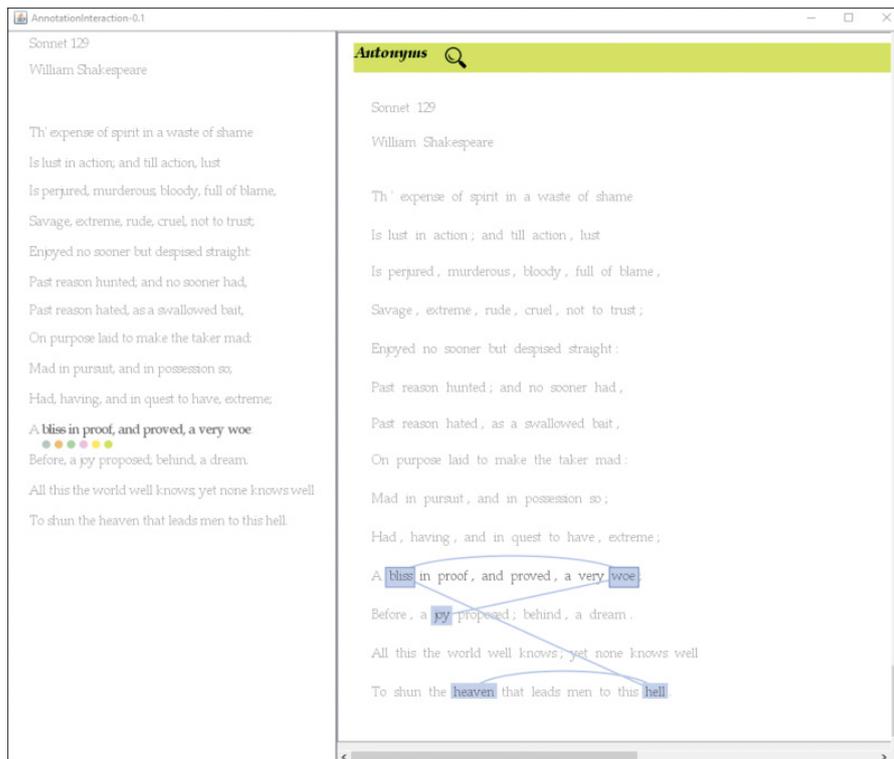


Figure 3: Screenshot of the Metatation interface where the reader underlines “bliss in proof, and proved, a very woe”. Metatation will search for and identify all antonyms due to identifying an antonym relation between “bliss” and “woe” [70].

2.2.3 Processing Annotations

There has been a lack of research on processing annotations in general. One notable example is XLibris [38], a pen-enabled tablet display that was the first to attempt using annotations to generate additional relevant content. Words annotated with a single pen stroke were used as keywords to find research papers related to the one being studied. These papers were presented to the reader as links in the margin near the corresponding pen stroke. Metatation [70] employs a similar approach in the domain of poetry. Annotations are analyzed as the reader processes the material to generate real-time, context-specific supplementary data. The difference was that Metatation focused on generating lower-level concepts, such as words or sentences, on the same document. For example, if the reader annotates words with an antonym relation, Metatation will retrieve and display all antonyms to the reader in real-time (see Figure 3). Another example is where Shipman et al. analyzed annotations by law students reading printed case law and writing Moot Court briefs [93]. The annotations are then processed and categorized to identify essential passages within the document, thereby automating the summary process. While these works demonstrate the potential for processing different types of annotations, they cannot fully implicitly interpret the purpose of the annotations beyond their domain, as the tool relies on the context of

the provided annotations (e.g., extracting relevant papers, visualizing key passages, and analyzing poetic elements). As previously discussed, annotations are both idiosyncratic and polymorphic in nature. Thus, assuming the annotation purpose greatly narrows the tool's potential and use cases. Ultimately, previous works cannot generalize the tool's domain or generate additional low-level, context-specific annotations.

2.3 SUMMARY

To sum up, annotations come in various forms and serve different purposes. Annotation tools have been developed to simplify the annotation process. This work will focus on fully digital systems, as they do not require additional hardware, unlike hybrid systems. Ovsiannikov et al. [81] emphasized that pen interfaces would be favoured if extra features were provided. Therefore, this work will also present new features to the pen interfaces that will provide a method to overcome all three limitations that previous methods cannot all address:

1. Inability to fully interpret and process all annotations, such as handwritten notes.
2. Difficulty in generating additional low-level context-specific annotations.
3. Lack of ability to generalize across different domains.

The next chapter will introduce large language models (LLMs) and how they can address these limitations.

In recent years, large language models (LLMs) have become increasingly popular for managing and creating new interactions with various systems [117]. LLMs are ML models and typically refer to Transformer-based [103] models that contain billions of parameters trained on massive text data [126]. They can understand and generate human language by predicting the likelihood of word sequences or generating new text based on a given input [21]. Mainstream transformer-based architecture generally falls into three categories [126]: (1) causal decoder, (2) prefix decoder and (3) encoder-decoder. This work will focus on the causal decoder, the most developed and researched architecture. Casual decoders include the unidirectional attention mask to ensure that each input token can only attend to the preceding tokens and itself [126]. The GPT-series is one of the most famous examples of causal decoders (e.g. GPT-4) and has been demonstrated to have human-like language understanding and generation [79]. One of their abilities is called in-context learning (ICL), where the model is trained to generate text based on a given context or prompt [21]. ICL enables LLMs to be interactive and conversational agents, generating more coherent and contextually relevant responses.

3.1 PROMPTING TECHNIQUES

Conversations with the LLM can typically accomplish tasks with prompting alone, even without any datasets, which is optimal since no datasets are available for document annotation. Prompts are instructions that describe the task to solve the said task [64]. One common prompting technique to enable ICL is called few-shot learning [17], where the prompt will contain exemplars of the solution before the test prompt is given (see Figure 4a). For example, to employ few-shot learning to sentiment analysis, the prompt can contain two exemplars (two-shot prompt) of the solution (e.g. I love ice cream! → positive) before giving the test sample. Few-shot learning has been demonstrated to improve overall performance on various tasks [17]. However, in some instances, it can hinder the inference process due to the exemplars introducing bias and semantic contamination [64, 88].

Another way to enable ICL is zero-shot learning, where the prompt contains no exemplars [64, 88] (see Figure 4b). Existing research has reported that causal decoders have better zero-shot performance than other architectures [126]. Zero-shot prompts are highly interpretable and more straightforward as they only need to handle task instructions. The prompt structure is also flexible, allowing us to insert our input wherever needed [64]. There are two approaches to creating effective zero-shot prompts: (1) heuristic-based manual design and (2) an optimization algorithm to find an optimal prompt. The second

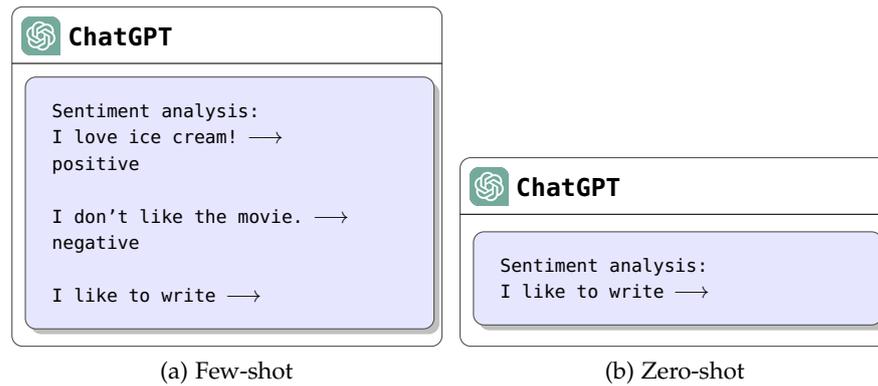


Figure 4: Sentiment analysis example using prompts where (a) gives two exemplars before the test sample is given (few-shot) while (b) gives no exemplar (zero-shot).

approach requires an evaluation set; again, it is unavailable for document annotation. Mishra et al. [71] proposed five principles for a good starting point in manual design: (1) using patterns about the expected output, (2) using bulleted lists and assertions, (3) breaking down tasks into multiple simpler ones, (4) adding explicit textual statements of output constraints, and (5) customizing the instructions so that the model can directly output the results.

Another prompting technique is improving the LLM’s reasoning capabilities, such as arithmetic reasoning and math word problems. Chain-of-Thought (CoT) [109] prompts can elicit reasoning in LLMs for few-shot learning by presenting reasoning steps in the exemplars (see Figure 5a). Later, a zero-shot CoT [52] was proposed by simply adding “Let’s think step by step” into the prompt (see Figure 5b) and has been optimized by an algorithm [129] to “Let’s work this out in a step by step way to be sure we have the right answer.” CoT significantly improves performance for problems that require reasoning rather than standard prompting. This also suggests that prompts need to condition or ask for good performance (e.g., “You are an expert on...”). This work was inspired by existing prompting techniques for interacting with LLMs and will be used to facilitate communication and collaboration with users.

3.2 HUMAN-AI COLLABORATION WITH LLMS

Artificial intelligence (AI) has been incorporated into systems due to its strong predictive performance [54, 92]. The aim is not to substitute human abilities but to enhance and complement them, fostering a collaboration that addresses each other’s weaknesses [122]. AI offers speed, precision, and the capability to process large datasets, while humans provide creativity, moral judgment, and rich contextual insight [54, 122]. This approach ensures efficient decisions while upholding societal values and ethical standards.

Additionally, the effectiveness of human-AI teamwork hinges on trust and transparency [122]. When AI systems are designed to be

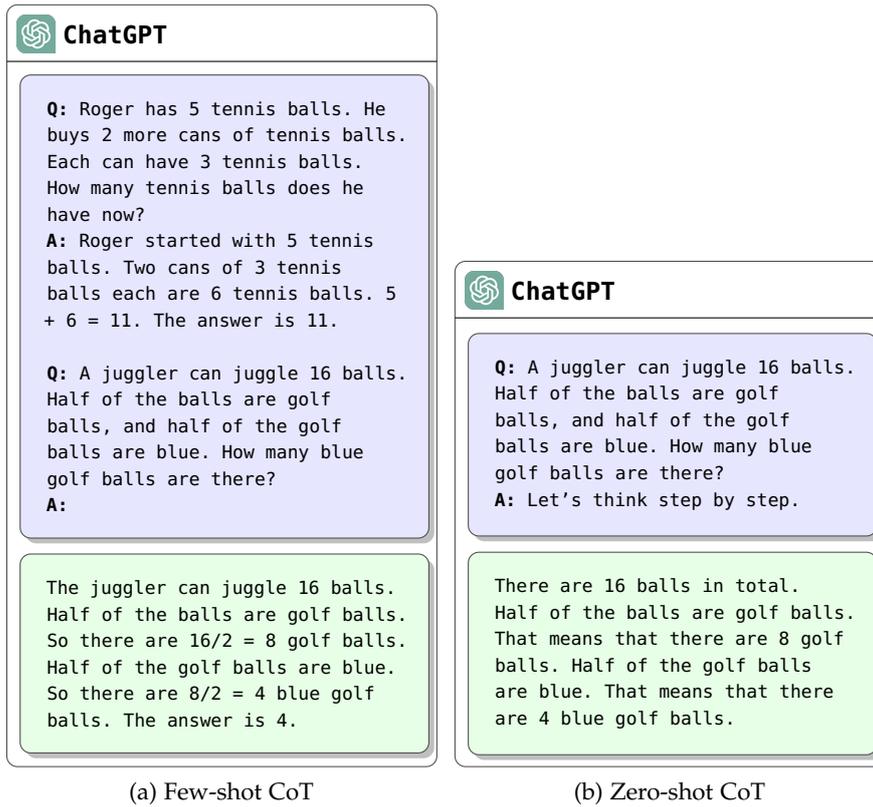


Figure 5: CoT prompts example on mathematical word problems adapted from previous works [52, 109], where (a) gives reasoning steps in the exemplar while (b) invokes reasoning steps by adding “Let’s think step by step” in zero-shot prompts. Blue boxes represent the user prompt, and green represents the output.

understandable, humans can grasp the rationale behind AI suggestions, facilitating responsible reliance on them [54]. This collaboration encourages mutual learning where humans direct AI with context and expertise, whereas AI helps humans undertake tasks that once seemed unattainable. Together, they achieve results neither could reach individually, creating a future where AI enhances human capabilities. For example, AI can make music and video recommendations to help people make subjective decisions [54]. Even in critical areas such as healthcare and criminal justice, AI tools offer suggestions and insights, but the final decisions rest with human experts [54, 92].

Many recent works have used LLMs as collaborative agents. One of the most popular AI agents used is OpenAI’s GPT series because it is available publicly via its API. These AI agents have demonstrated collaboration across various areas. For example, they can work as writing assistants, suggesting ideas for a story [26, 31, 35, 44, 91, 92, 97]. Users can explore themes and keywords generated by GPT to guide the creation of a creative story. Multimodal LLMs also expand the storytelling process to include AI-generated images combined with the generated story [31, 106]. They have demonstrated an original and effective way to build stories, providing controls supporting writers in their creative flow.

Another example is AI agents acting as teaching assistants for language learning [57, 90, 92, 127] or general education [24, 89]. Everyone’s learning process is unique, and the goal of the LLM is to adapt to each learner’s needs. AI agents interact with the learner and generate personalized questions [90, 127] or break down long content into multiple smaller lessons [24, 89]. It demonstrates that learners using these AI agents have a higher proficiency in the subject, making the learning process more engaging than traditional approaches.

Multiple AI agents can also collaborate, simulating real-world situations in a virtual environment, such as a small community [84], a company for literary translation [114], or a hospital [62]. These examples have used OpenAI’s agents, demonstrating that they can interact with the environment and other agents, and store previous experiences to guide future interactions. For example, Agent Hospital [62] uses OpenAI’s GPT agent to simulate doctors treating diseases. Starting with empty medical knowledge, they can self-evolve by storing past medical records and experiences in their medical knowledge, achieving a state-of-the-art accuracy of 93.06% on the MedQA dataset. It suggests they can reflect on past inferences and improve on future ones.

Recent works have also used LLMs to annotate text with humans but not to make sense of a text document, rather than parsing the text to extract critical information for information visualization [37, 47] or easing the process of annotating datasets [50, 76, 77, 108, 120].

In summary, current research on OpenAI’s GPT agents demonstrates their ability to collaborate effectively with humans by accommodating their creative processes and specific requirements. Inspired by how LLMs can work flexibly around users and reflect on their past experiences, this work will explore human-AI interaction with OpenAI’s GPT agent to enhance human abilities in co-annotating. To the best of our knowledge, this particular collaboration has not been explored before.

3.3 VISION

LLMs can be multimodal, meaning they can process different input types simultaneously. One possible modality is vision, wherein the input can contain images, and the task can be performed with the image [5, 124]. Prompting techniques can also be applied to multimodal LLMs to define and customize the desired output text for arbitrary vision-language use cases such as object localization, image captioning, visual question and answering, etc. [94, 107, 115]. As discussed, GPT-4 possesses human-like text understanding and generation abilities. Now, OpenAI has developed GPT-4o, which also excels in comprehending the image domain while having the same abilities as GPT-4 (see Figure 6). It can also be applied to many domains due to its vision and text capabilities. For example, risk perception in traffic [30], medical applications [113], even robotics control [105]. Given its generalizability across many domains and the idiosyncratic

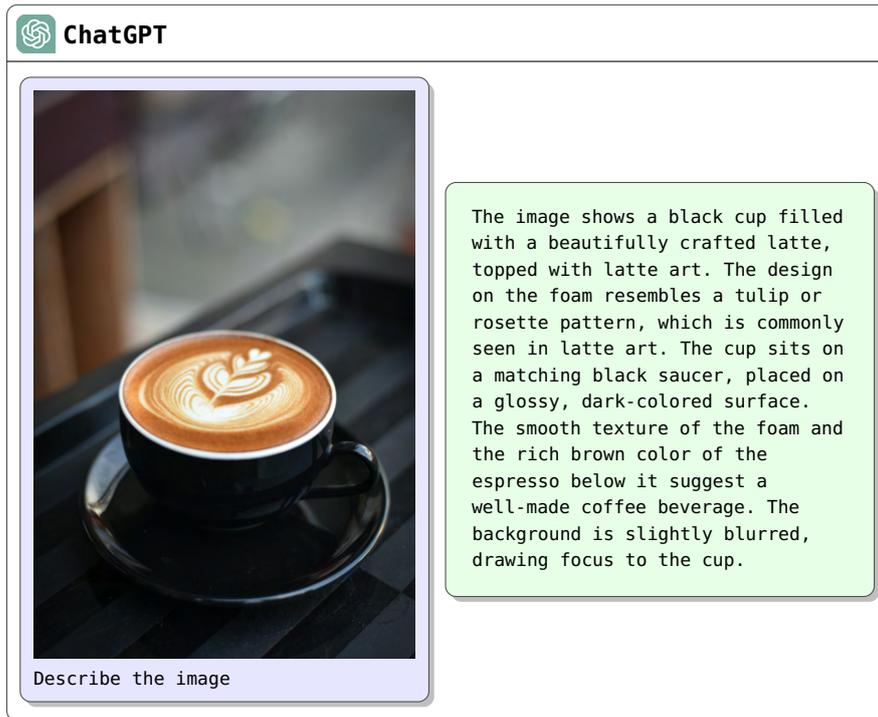


Figure 6: Multimodal prompt example on image captioning where ChatGPT describes a coffee cup with extreme detail. Blue boxes represent the user prompt, and green boxes represent the output.

and polymorphic nature of annotations, this work will use GPT-4o to interpret the purposes or functions of annotations.

3.4 RETRIEVAL AUGMENTED GENERATION

Stand-alone generation models have been shown to have a substantial amount of general knowledge without accessing any external memory. However, they cannot quickly expand and revise their memory and can't provide insight into their predictions [59]. They can also produce content inconsistent with factual statements or user inputs; this phenomenon is called "hallucination" [45]. LLMs also have a limitation on the maximum number of input tokens they can handle and may not perform well when dealing with long texts [63, 128]. To address these issues, a retrieval module is added to the LLM, which retrieves relevant documents and feeds them into the generation model, also known as retrieval-augmented generation (RAG) [23, 59, 60, 121, 125]. RAG enables the acquisition of knowledge in an explicit manner, allowing for plug-and-play functionality, which results in great scalability. Additionally, LLMs can also refer to human-written text [60]. An effective retrieval for LLMs has four components [125, 131]: (1) query rewriter, (2) retriever, (3) reranker and (4) reader. First, the query rewriter rephrases the original query to address vague queries or vocabulary mismatches between the query and target documents [131]. Then, the retriever retrieves the relevant documents based on the queries, and the reranker will rank these documents [125, 131]. Finally, the reader summarizes the documents [131]. It has achieved state-of-

the-art performance on NLP tasks [59] and exhibits a significantly higher correlation between queries and results [23].

Previous human-AI collaboration tools have used RAG to retrieve past interactions [62, 84]. Even a simple retriever, without the other three components, can guide future interactions based on past experiences, as discussed previously. OpenAI has developed a retriever¹ that parses and chunks documents, creates and stores embeddings, rewrites complex user queries into multiple searches that can run in parallel, conducts both keyword and semantic searches and reranks search results to select the most relevant ones before generating the final response. This work will use OpenAI’s retriever to help guide future inferences from past correct inferences, similar to previous applications. It can also be used to inject the document contents into the LLM, as putting the entire document or a significant portion of it in the prompts is not feasible.

3.5 SUMMARY

Based on past research and applications of LLMs, various tasks can be solved by translating them into natural language via prompts. Prompting techniques can be applied to the document annotation domain, which has been proven to condition the LLM for good performance. A popular example of a multimodal LLM is GPT-4, which is used extensively in this work to collaborate with humans to annotate documents. Leveraging its general knowledge and combined with RAG, it will help address the three limitations described in [Section 2.3](#):

1. *Inability to interpret and process all annotations, such as handwritten notes.*
Infer the purposes of annotations using GPT-4’s vision capabilities. With the help of RAG, past correct inferences can be incorporated into the LLM to aid in future inferences, providing insight into the user’s personalized annotation style.
2. *Difficulty in generating additional low-level context-specific annotations.*
RAG can also be used to inject the document contents into the LLM and extract any content that matches the purpose of the annotation
3. *Lack of ability to generalize across different domains.*
Naturally, the general knowledge the LLMs possess can be applied to many domains of documents.

¹ <https://platform.openai.com/docs/guides/tools-file-search>

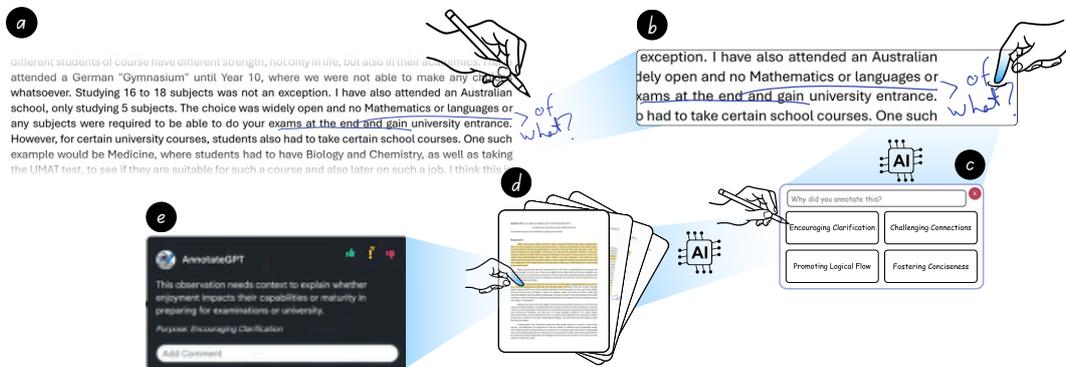


Figure 7: A high-level overview of the interaction design of AnnotateGPT. (a) The user manually annotates the document first. (b) Tapping on the annotation will activate an assistant. (c) Upon tapping, the assistant will guess the purpose of the annotation. (d) Selecting a purpose will prompt the assistant to provide further annotations (yellow highlights) based on the selected purpose. (e) Users can read, verify and continue the feedback.

As discussed, one of the typical contexts for annotation is author-directed, which is made by readers to share with the author [81]. For example, teachers give feedback to students on an assignment. Annotation enhances learning and assessment in student essays, where it is helpful in terms of providing specific and accurate feedback [7]. This work will center on this type of context, and the design of our proposed system will revolve around this context. This chapter introduces AnnotateGPT and its design guidelines. The following will detail methods used to employ implicit pen annotation assisted by LLMs, including (1) clustering pen strokes, (2) inferring annotation purpose, and (3) generating low-level context-specific annotations.

4.1 CHALLENGES IN PROVIDING FEEDBACK TO STUDENTS

As educational institutions shift from traditional to digital annotation methods, both teachers and students express a preference for handwritten comments. Both feel are more personalized and specific, hence more effective benefit [14]. However, teachers report that handwritten annotation practices are generally hindered by three issues: legibility, strict time limitations, and inconsistent feedback quality.

The degraded legibility of annotations is primarily due to physical and temporal constraints that affect the delivery of feedback to students and teachers. Students struggle to interpret the annotations due to their readability and unclearness [7, 27]. On the other hand, teachers are unable to provide detailed, thoughtful feedback due to time

constraints and the importance of prompt return to students [6, 14, 95], affecting the quality of feedback. Furthermore, the tone of feedback could be overly negative or opinionated, which could undermine the student’s confidence [7, 85, 95]. Beyond these immediate challenges, handwritten feedback often focuses too narrowly on surface-level errors (e.g., grammar) rather than providing balanced, constructive guidance [3, 85, 123].

This work will address these three issues by having an LLM collaborate in the annotation process, offering legible, timely, and constructive feedback. Annotations can be enhanced with AI-generated insights that preserve the personal tone of human feedback while improving clarity and depth. The LLM can support teachers by generating drafts of comments based on their annotations, reducing manual labour and enabling more consistent, high-quality responses. This hybrid approach aims to retain the personalized nature of handwritten feedback while mitigating its common pitfalls, ultimately enhancing both the efficiency and pedagogical impact of annotation practices.

4.2 DESIGN GUIDELINES

Based on the analysis of previous work, a system needed to be sensitive to how people create and use annotations. Therefore, the following goals were kept in mind when designing the interface:

- DG1. *Supports Pen-Based Devices.* The use of pen-based devices has grown in popularity due to the increased accessibility, decreased costs, and enhanced sensing capabilities of pen-enabled devices such as tablets and smart whiteboards [72]. Such devices afford users direct, natural interaction with interface elements and workspace content. However, they have also resulted in disjointed, fragmented user experiences. For example, many applications use hierarchical menus to organize supported functionality. Users often rely on hotkeys, key holding, and mouse clicking to access functionality. When using a pen within such applications, users cannot use shortcuts, so they are required to perform multiple “round trips” between a menu and their workspace, which leads to increased interaction costs and user frustration [33, 40, 87]. Instead, this work will use markers near areas of interest (e.g. annotated texts), reducing menu “round trips”, to support pen-based devices rather than traditional static menu bars and hierarchical menus.
- DG2. *Similar to Paper Experience.* Having the annotation experience be similar to the paper is crucial. For example, a freehand interface that supports multiple markings was perceived to have advantages that do not support many markings [75] as reviewers and editors rely on a wide range of markings when annotating paper texts [32, 81]. Compared to “sticky note” interfaces, in which reviewers insert textbox icons on top of text, it will be inferior because sticky notes often obscure the underlying text [81]. Thus,

this work will use a pen-based device with digital ink, allowing users to mark the document freely, much like on physical paper.

- DG3. *Limiting System Complexity.* Past works would use gestures to implement enhanced features for annotation [118, 119]. For example, RichReview [119] uses a pigtail gesture to anchor annotations to the gesture. However, errors can arise when using such gestures. Furthermore, this work evaluates the LLM’s effectiveness in annotation, and such gestures could affect the results. Thus, the proposed system would use common gestures, such as tapping, to reduce training time and mental demand, while focusing on evaluating the performance of the LLM. The simplicity also helps non-technical users, such as teachers, easily integrate it into their workflow, given their prior experience with annotation interfaces [92].
- DG4. *Maximum Workspace Size.* There are significant effects of interface layout where past works compare different types of layouts [111, 112]. Researchers have found that a split-screen interface, presenting the main text and annotations in separate panes, was less likely to produce and respond to annotations than an interface that directly embeds annotations into the document. Furthermore, documents with a large “virtual” margin area had advantages over the traditional logical page regarding legibility and time [86]. These results suggest that reviewers require a workspace integrated into the document with sufficient space to produce annotations. Following these results, this work will initially hide all menus and markers, allowing users to display a menu or marker with a hover gesture.
- DG5. *Give Users Agency.* Taken from past works on systems incorporating AI, it should enhance **human** capabilities on tasks, not replace them. This approach not only provides more creative and contextual insight but also upholds societal values and ethical standards [54, 122]. For example, users can verify AI suggestions, provide suggestions when asked, or select options to pick from [92].

4.3 FRAMEWORK

This work introduces AnnotateGPT, a digital pen annotation tool that leverages an LLM to create two AI agents: (1) to infer the annotation form and purpose, and (2) to generate additional low-level context-specific annotations. It first clusters pen strokes to classify which pen strokes correspond to which annotations. A marker is assigned to each cluster, and upon activation, it prompts agent (1) to generate four potential annotation purposes. Finally, a user selects one of the purposes to assign to agent (2). The framework overview is shown in [Figure 8](#).

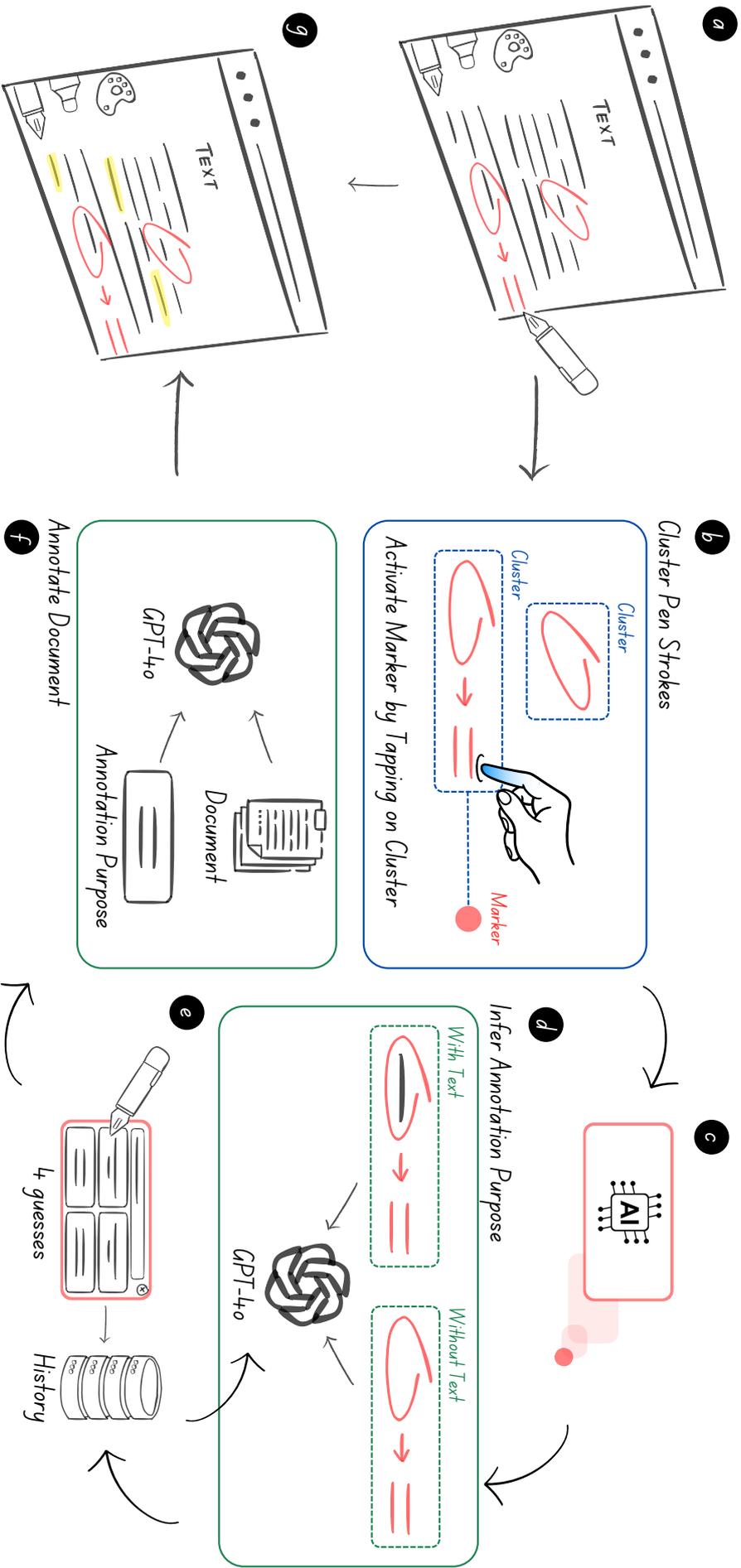


Figure 8: An overview of AnnotateGPT’s framework: (a) The user first annotates the document. (b) AnnotateGPT then clusters the pen strokes based on spatiotemporal distance, representing an annotation. (c) The user taps on the cluster/annotation to activate and open the assistant. (d) The assistant captures two images from the cluster, one with the underlying text and one without, and makes four guesses about the annotation’s purpose. (e) The user then selects a purpose, which AnnotateGPT will remember for future inferences. (f) Finally, AnnotateGPT generates annotations based on the selected purpose and (g) highlights them on the document.

4.3.1 Stroke Clustering

A common challenge in marking-based interactions is interpreting pen strokes. Annotations are often idiosyncratic, informal, and ambiguous. A marking-based system must handle these variations for practicality to interpret the user’s visual intent. A key challenge is determining where one annotation ends and another begins. Existing systems require users to press a button on the digital pen or pause between symbols [34, 74]. Some systems even mandate that each object be drawn in a single stroke [55]. These constraints compromise the natural paper experience (DG2) and increase the system’s complexity (DG3), conflicting with the design guidelines.

Alternatively, previous works employ a clustering algorithm to group pen strokes into distinct clusters, allowing symbols to be naturally drawn with an arbitrary number of strokes [25, 49, 70]. This method uses a general-purpose clustering algorithm to process unprocessed pen strokes and group them into distinct clusters, each representing a single symbol, in our case, a single annotation. Note that the parser has no prior information about the sizes and shapes of the clusters, the number of strokes they may include, or even the total number of clusters to be identified.

This work will use hierarchical agglomerative clustering, as proposed by Chiu and Wilcox [25], with single linkage, which has been found to outperform mean and double linkage [70]. Starting with each stroke as its cluster and repeatedly merging the two closest clusters until a stopping criterion is met. The single linkage is the minimum pairwise distance between pen strokes in the clusters, used to determine the distance between clusters. This work defines the distance between two pen strokes as a weighted function of its spatial and temporal distance (spatiotemporal distance) proposed from Metatation [70]. The temporal distance between two pen strokes, s_1 and s_2 , is defined as:

$$\Delta t = \begin{cases} \frac{t_0(s_2) - t_f(s_1)}{30} & \text{if } t_0(s_2) - t_f(s_1) < 30 \text{ seconds,} \\ 1 & \text{otherwise,} \end{cases}$$

where $t_0(s_2)$ is the start time of s_2 and $t_f(s_1)$ is the end time of s_1 . The temporal distance has an upper bound of 30 seconds to ensure normalization across all pen stroke pairs. The spatial distance between two pen strokes is the minimum Euclidean distance between two bounding boxes of s_1 and s_2 . The size of the page then normalizes the spatial distance. Thus, the spatiotemporal distance is defined as:

$$d_{st} = \sqrt{\Delta s^2 + \Delta t^2}$$

According to this definition, hierarchical agglomerative clustering merges the closest pen strokes. As mentioned, the challenge is not knowing the number of clusters in advance. If this number were known, the clustering algorithm could stop once the target clusters were identified. Fortunately, a heuristic can automatically identify

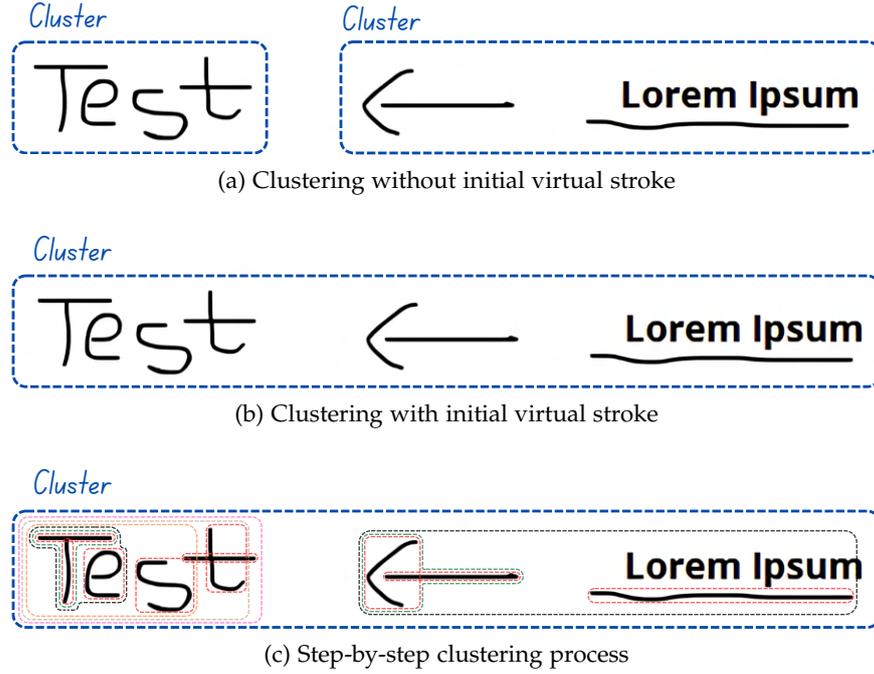


Figure 9: Sample clustering process with and without a virtual stroke. (a) In the absence of the virtual stroke, the clustering fragments the annotation because it is the sole annotation. (b) The virtual stroke helps to resolve the issue. (c) The steps of the clustering algorithm are visualized, with each bounding box representing one step.

this number [49]. At each iteration or merge, the pairwise distance between clusters increases. If there is a sharp increase in the closest pairwise distance, it is indicative of a “forced merge,” a merge that combines two distant clusters [49]. Thus, identifying the optimal stopping iteration, i^* .

$$i^* = \arg \max_i \left[\frac{d_{i+1} - d_i}{d_i - d_{i-1}} \times (d_{i+1} - d_i) \right]$$

The ratio in this expression compares the increase between consecutive iterations to detect sharp increases. However, if the previous increase was tiny, even a minor rise can exaggerate the ratio, especially in early iterations. The method prioritizes larger global leaps by using the absolute increase $(d_{i+1} - d_i)$ as a scaling factor to address this.

However, through empirical observation, if the user produces a complex annotation (i.e., an annotation consisting of more than 3 strokes) and it is the only annotation on the page, the clustering algorithm fragments it (see Figure 9). By adding another annotation, the algorithm clusters correctly. The stopping criterion is the maximum distance change between consecutive iterations, forcing the sole annotation to fragment. Using this behaviour, a “virtual” stroke with a timestamp of 0 and a size of 1×1 pixel is added to the page. As seen in Figure 9a, the annotation consists of underlining, an arrow and a handwritten note. The clustering cannot recognize the annotation as a whole based on the stopping criterion. When inserting an initial virtual stroke, it recognizes the complex annotation as a whole and the virtual stroke

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In efficitur, felis id cursus cursus. ligula lacus auctor ipsum, ut faucibus sapien~~tem~~ nec massa. Suspendisse quis lorem sed est blandit

Figure 10: A hand-drawn underline with significant jitters, where segments of the stroke are classified as cross-outs (blue) and underlining (red). In this case, the stroke is classified as underlining, as most points are. Also, only the first line is extracted.

as a separate cluster (see Figure 9b). Note that this faulty behaviour only occurs when a complex annotation is the only one on the page. A sample clustering process is shown in Figure 9c, where it first clusters the arrow and then the word ‘test.’ Afterward, it groups the arrow and the underline and then groups all strokes into a valid cluster. The cluster is assumed to be an annotation for further processing (Figure 8b), which will be described in future sections.

4.3.2 Stroke Classification and Text Extraction

A heuristic classification is used for every stroke to classify strokes and extract annotated text, emphasizing annotated text extraction. This work classifies three types of annotations: (1) HIGHLIGHTING, CROSSING OUT, and UNDERLINING (horizontal lines), (2) CIRCLING, and (3) ANNOTATED. Text is extracted based on type for LLM input, described in future sections. A stroke is defined by a list of xy coordinates representing the digital pen’s position over time.

Algorithm 1 Horizontal Line Detection

```

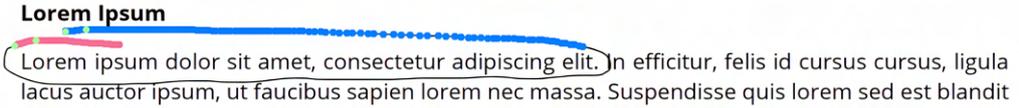
function ISHORIZONTALLINE ( $P$ )
  if  $length(P) < 2$  then
    return false
  end if

   $y_0 \leftarrow P[0].y$ 
  for  $i \leftarrow 1$  to  $length(P) - 1$  do
    if  $P[i].y < y_0 - 20$  or  $P[i].y > y_0 + 20$  then
      return false
    end if
  end for

  if  $\sigma_y > \sigma_{maxY}$  then
    return false
  end if
  return true
end function

```

Two heuristics detect horizontal lines: (1) all y -coordinates must be within 20 pixels of the first y -coordinate, and (2) y -variance, σ_y , is less than $\sigma_{maxY} = 5$. These heuristics account for human error, as humans

Lorem Ipsum


>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In efficitur, felis id cursus cursus, ligula lacus auctor ipsum, ut faucibus sapien lorem nec massa. Suspendisse quis lorem sed est blandit

Figure 11: A hand-drawn enclosure, where red points represent past sliding windows starting from one end and blue points on the other. Green points represent the sliding windows that are “loosely” intersected, classifying the stroke as a successful enclosure.

do not naturally draw perfect horizontal lines. Pseudocode is provided in [Algorithm 1](#). When a horizontal line is detected, it is assigned one of three sub-types: HIGHLIGHTING, CROSSING OUT, or UNDERLINING. To assign sub-types, most stroke points must be near (within 10 pixels) the center of the word for CROSSING OUT and HIGHLIGHTING, or below the word for UNDERLINING. This process is not trivial due to human error, as shown in [Figure 10](#), where stroke segments are inconsistent. The proposed solution identifies the majority of sub-types to handle jitter errors, classifying this stroke as “underlining the first line” and ignoring the second line. The leftmost and rightmost words affected are extracted to capture all text in between, ensuring no short words, like “a”, are missed due to the sampling rate. HIGHLIGHTING follows the same criteria as CROSSING OUT, with additional checks for highlighter use. If no points are near text or a sub-type cannot be assigned, it defaults to ANNOTATED.

To detect CIRCLING, the beginning and end of the stroke must be close or intersecting. This method detects enclosures, not just circles, such as circling a sentence that extends to the following line. A sliding window is used at each stroke end, starting from one end to one-third of the stroke’s length, with a minimum window size of 30 pixels. Each iteration checks if two sliding windows are “loosely” intersected using line segment intersections, where the intersection point falls within the first line segment if $0 \leq \lambda \leq 1$ and within the second if $0 \leq \gamma \leq 1$. To account for human error, the range is adjusted to $-1 \leq \lambda, \gamma \leq 2$, allowing for an intersection within two times the segment length. [Figure 11](#) illustrates the enclosure detection, showing

Algorithm 2 Line Segment Intersection

```

function INTERSECTS( $a, b, c, d, p, q, r, s$ )
   $det \leftarrow (c - a) \times (s - q) - (r - p) \times (d - b)$ 

  if  $det = 0$  then
    return false
  else
     $\lambda \leftarrow \frac{(s - q) \times (r - a) + (p - r) \times (s - b)}{det}$ 
     $\gamma \leftarrow \frac{(b - d) \times (r - a) + (c - a) \times (s - b)}{det}$ 
    return  $-1 \leq \lambda \leq 2$  and  $-1 \leq \gamma \leq 2$ 
  end if
end function

```

Algorithm 3 Enclosure Detection

```

function CHECKENCLOSED( $P$ )
  slidingWindow1:
  for  $i_1 \leftarrow 0$  to  $\frac{\text{length}(P)-1}{3}$  do ▷ Sliding window
     $\sigma_1 \leftarrow 1$ 
     $x_1 \leftarrow P[i_1].x$ 
     $y_1 \leftarrow P[i_1].y$ 
     $x_2 \leftarrow P[i_1 + \sigma_1].x$ 
     $y_2 \leftarrow P[i_1 + \sigma_1].y$ 

    while  $\text{dist}(x_1, y_1, x_2, y_2) < 900$  do ▷ Check line distance
       $\sigma_1 \leftarrow \sigma_1 + 1$ 

      if  $i_1 + \sigma_1 \geq \frac{\text{length}(P)}{3}$  then
        break slidingWindow1
      end if
       $x_2 \leftarrow P[i_1 + \sigma_1].x$ 
       $y_2 \leftarrow P[i_1 + \sigma_1].y$ 
    end while

  slidingWindow2:
  for  $i_2 \leftarrow \text{length}(P) - 1$  to  $\frac{2 \times \text{length}(P)}{3}$  do ▷ Sliding window
     $\sigma_2 \leftarrow 1$ 
     $x_3 \leftarrow P[i_2].x$ 
     $y_3 \leftarrow P[i_2].y$ 
     $x_4 \leftarrow P[i_2 - \sigma_2].x$ 
     $y_4 \leftarrow P[i_2 - \sigma_2].y$ 

    while  $\text{dist}(x_3, y_3, x_4, y_4) < 900$  do
       $\sigma_2 \leftarrow \sigma_2 + 1$ 

      if  $i_2 - \sigma_2 \leq \frac{2 \times \text{length}(P)}{3}$  then
        break slidingWindow2
      end if
       $x_4 \leftarrow P[i_2 - \sigma_2].x$ 
       $y_4 \leftarrow P[i_2 - \sigma_2].y$ 
    end while

    if INTERSECTS( $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ ) then
      return true
    end if
  end for
end for
return false
end function

```

blue and red points for past sliding windows and green points for loosely intersected windows. The pseudocode is in Algorithms 2 and 3. Once CIRCLING is detected, the underlying text is extracted if at least 50% of its bounding box is within the enclosure. The type changes to ANNOTATED if no underlying text exists.

The final type, ANNOTATED, generally handles all other types of annotation described in Chapter 2. Note ANNOTATED is assigned if no underlying text can be extracted from the other types. Instead, ANNOTATED will look for any text intersecting the stroke or nearby text. Other annotations are assumed to be within the text or in the side margins, as these are the typical cases [69]. Thus, any lines within the range of the stroke’s bounding box on the y-axis are extracted. If no lines are extracted, the closest paragraph is extracted instead.

4.3.3 Annotation Classification and Purpose Inference

All stroke classifications and extracted text, along with images of the stroke cluster, will be fed into the LLM to classify and describe the overall cluster/annotation. Due to the non-deterministic nature of annotations described in Chapter 2, the LLM will output four different guesses on the purpose/function of the annotation (Figure 8d). The prompt consists of three manually designed parts: the problem statement, input, and parsing instructions.

First, a system prompt describes the problem to ensure the LLM understands what to expect from the input, consisting of two images of annotation, stroke type, and annotated text. It lists various types of annotations found in the wild [68, 69] and outlines the image processing steps. Here is a brief overview of the steps:

- (1). Describe the annotation using the types listed.
- (2). Guess the purpose of the annotation based on the context.
- (3). Use multiple branches of thinking.
- (4). Review the history of annotation styles for personalization.
- (5). Summarize and relate findings to the annotation.
- (6). Make potential guesses about the annotation’s purpose by adopting different personas.
- (7). For each guess, give two specific levels of detail: specific and broad.

Step (4) initiates RAG to retrieve previously picked purposes based on the annotated text, initially starting with an empty history. This step injects the user’s personalized annotation style into the LLM. Step (6) leverages this history to offer diverse interpretations by using multiple thinking branches (step (3)) and adopting different personas [10]. Once the user selects a purpose, the history updates to inform future inferences in step (4). To better accommodate the user’s unique annotation styles, step (7) aims to understand how the user engages with content

Prompt Role	Prompt
System	You are an expert in describing annotations and determining their purpose. You will be shown two images of annotations from a document a user has personally annotated. The first image shows the annotation with the document, and the second shows the annotation without the document. Do not give vague answers, such as whether the user is interested in or emphasizing the text; be very specific. Describe your steps first.
User	<p>Types of annotation:</p> <ul style="list-style-type: none"> - circles or boxes - underlining - highlighting - crossing out - handwritten notes/text - punctuation marks (e.g., commas, periods, question marks, asterisks, etc.), choose which one - arrows - brackets, angle brackets, or braces <p>Here are the steps:</p> <ol style="list-style-type: none"> 1. Describe the annotation by reviewing the list of annotation types for possibilities. 2. Guess the purpose of the annotation based on the context. 3. Use [four / two] branches of thinking such as backtracking to check for any other possibilities. 4. Look at past annotation history in your knowledge base 5. Summarize your past findings and relate them to the annotation 6. Give [four / two] different guesses of the purpose using different personas and past annotation history. The purposes should have different themes and relate to the context. 7. For each guess, give two levels of detail: specific and broad. <p>When describing with specific, describe the purpose so it is specific to the words of the annotated text. When describing with broad, use umbrella terms without using the annotated text.</p> <p>[<i>annotation image</i>] [<i>annotation image without underlying text</i>]</p> <p>Context: The user is [<i>context</i>] and has [<i>annotation type and annotated text</i>].</p>
User	[<i>Parsing instructions</i> (format shown in Figure 12)]
User	Let's work this out in a step by step way to be sure we have the right answer.

Table 2: The prompt for annotation classification and purpose inference. Blue text is only added when no stroke annotates more than two words to understand further how the user engages with content.

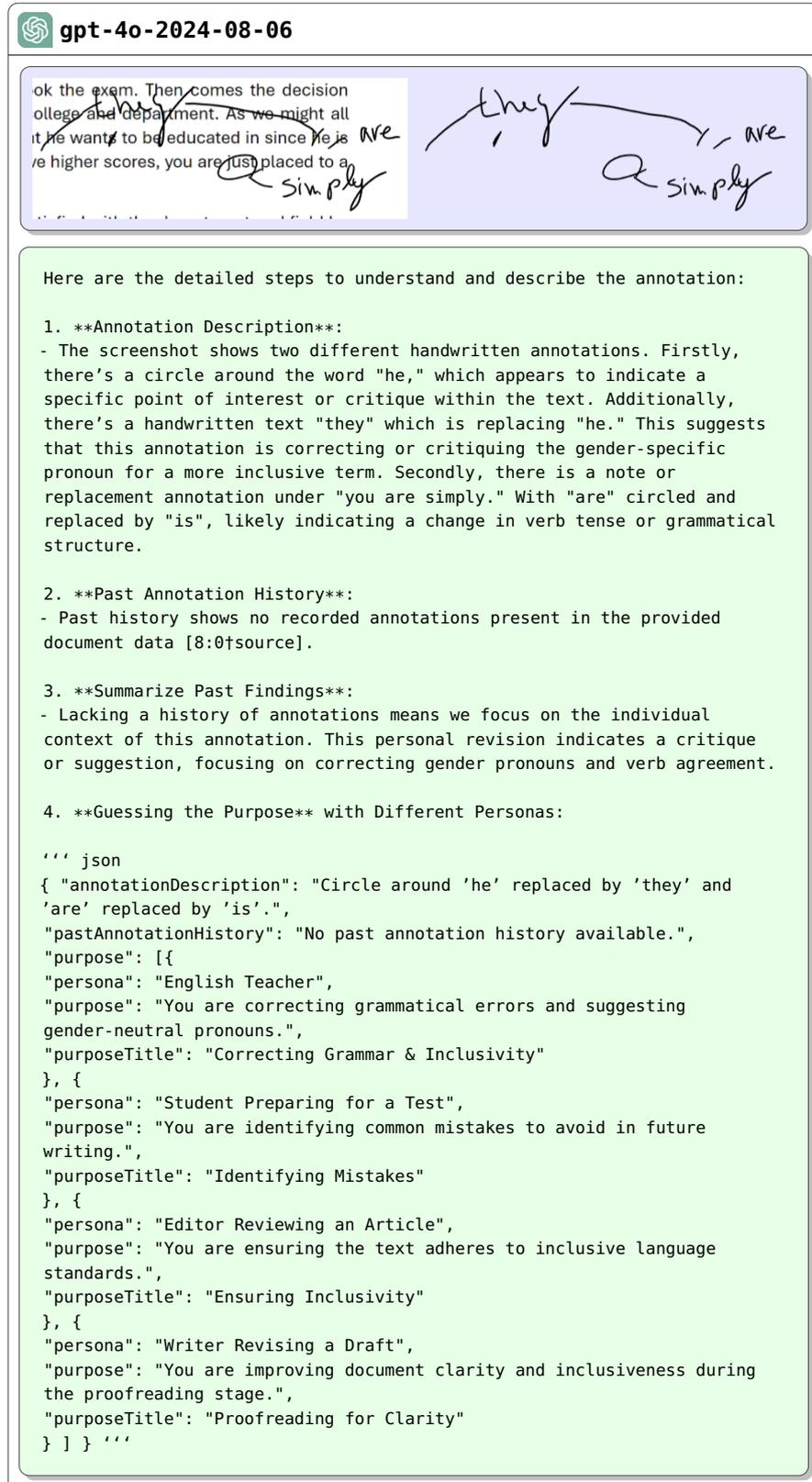


Figure 12: Sample results for annotation classification and purpose inference

on a word level, determining whether their intent is specific to the text or more general. This allows for flexibility across various contexts and preferences. Step (7) is triggered only when no stroke annotates more than two words. The exact prompt is detailed in [Table 2](#).

Secondly, a user prompt feeds two images of the annotation, its context (provided by the user once), the annotation type, and the annotated text. One image includes the underlying text, while the other does not, ensuring clarity without unnecessary noise. For example, isolating the handwritten notes that may overlap with the document text or be embedded within the text. To give further hints, the annotation type and annotated text are extracted as described in [Section 4.3.2](#). Note that stroke classification operates at the level of individual strokes. Therefore, an annotation may include multiple stroke types aggregated into the prompt. ANNOTATED is only considered if it is the sole stroke type among all strokes. Providing “hints” will guide a black-box LLM toward a specific output and has been shown to improve reasoning accuracy [65].

Finally, parsing instructions are provided to structure the output into a standardized format, such as JSON, XML, or a list, also known as structured outputs. However, requiring an LLM to produce only structured output instead of a free-form response may impair its reasoning abilities [100]. Therefore, the parsing instructions will be included as part of the free-form response at the end. This work uses a JSON object to store the annotation description, related history, and four guesses on the annotation purpose, each made by different personas. Sample results from a pilot study are shown in [Figure 12](#). In this example, it can decipher the handwritten notes and their purpose with grammatical structure and gender-neutral language.

4.3.4 *Generating Low-Level Context-Specific Annotations*

After a user selects a purpose, the LLM generates annotations accordingly. The prompt outlines a seven-step annotation process:

- (1). Specify details to look for based on the purpose.
- (2). Offer feedback on why the sentence needs annotation.
- (3). Offer improvement suggestions.
- (4). Feedback is kept under 20 words for clarity.
- (5). Avoid annotating sentences that need no modifications.
- (6). Sentences and feedback are wrapped in delimiters: triple asterisks and double curly brackets, respectively (single tokens in gpt-4’s vocabulary).
- (7). Optionally, target specific words within the sentence, delimited by triple quotation marks.

Step (1) ensures annotations are not too broad. For instance, if the user annotates a comma for “punctuation usage”, the LLM won’t

Prompt Role	Prompt
System	<p>You are an expert at annotating documents. The user has annotated the document I have given you. Given the purpose of an annotation, you will find all sentences in the document that could be annotated for the same purpose. Do not change the sentence from the document in any way. Give one sentence in one annotation. Describe your steps first. Do not ask follow-up questions.</p>
User	<p>[<i>context</i>]. Read every page and find sentences that could be annotated with:</p> <p>[<i>purpose</i>]</p> <p>Here is a step-by-step list for annotating a document:</p> <ol style="list-style-type: none"> 1. Describe what details in sentences to look for in the document. Be specific. Do not change the original purpose in any way. 2. Explain why you annotated the sentence. 3. Suggest fixes for the sentence by describing the fix without giving the answer. 4. Combine the explanation and suggestion without quoting the sentence using less than 20 words. 5. Do not include any sentences that need no modification. 6. Make a list of sentences for each response using triple asterisks for sentences and double curly braces for the explanation and suggestion. For example: <pre>## Response <number> *** <sentence> *** {{ <explanation and suggestion> }}</pre> <p>...</p> <ol style="list-style-type: none"> 7. For each sentence, you can optionally target words in the sentence to annotate. If you do, list the words or phrases to look for in the sentence, separated by commas and enclosed by triple quotation marks. For example: <pre>## Response <number> *** <sentence> *** """ <words or phrase to look for (e.g. <word/phrase 1>, <word/phrase 2>)> """ {{ <explanation and suggestion> }}</pre> <p>...</p> <p>Make sure you have all the sentences needed to be annotated in the format above.</p>
User	<p>Walk me through one question at a time in manageable parts step by step, summarizing and analyzing as we go to make sure we have all the sentences needed to be annotated</p>

Table 3: The prompt for generating annotations by extracting the sentence and its associated feedback and targeted words.

Document:	... a student will be able to get more of the grades and could get more attention if he has the collection of various innovative ideas accompanied with the facts told by him
vs.	
ChatGPT:	A student will be able to get more attention if he has various innovative ideas accompanied with the facts told by him.

Figure 13: Illustration of the difference in the document and ChatGPT output. Therefore, a more rigorous approach is necessary to accurately match sentences from ChatGPT to the document.

search for other punctuation errors like periods, closely aligning results with human expertise. Steps (2) and (3) provide an initial AI review that encourages user comments about the addressed issues [111]. The LLM can provide unnecessary modifications, so step (5) is provided to avoid this. Step (7) lets the agent optionally focus on specific words, allowing the user to identify the issue within the sentence quickly. The detailed prompt is shown in Table 3. Thread of Thought (ThoT) [130] prompts handle chaotic contexts (e.g., long documents with distractors) better than CoT by systematically breaking down and analyzing information, plugging in “Walk me through this context in manageable parts step by step, summarizing and analyzing as we go.”

The purpose of these delimiters is to easily extract each annotated sentence along with its associated feedback and target words. Once extracted, sentences will be searched for and highlighted in the document. However, the solution is not trivial, as sentences generated by the LLM may not precisely match those in the document. Through testing, it was revealed that ChatGPT omitted words in the output, yet still provided a valid sentence. Thus, a more rigorous approach is needed (see Figure 13). The search algorithm performs in two scopes (*single-page* and *cross-page*) at two levels (*exact* and *fuzzy*):

SINGLE-PAGE Extracts all text from one page simultaneously.

CROSS-PAGE Sentences can cross over to the next page. Therefore, this scope extracts N characters from the end of one page and the start of the next, where N is the length of the targeted sentence.

EXACT Searches for an exact match within the document. Non-alphanumeric characters are filtered out and removed (e.g., commas and periods) to optimize computation.

FUZZY In some cases, ChatGPT can provide a valid sentence with words omitted where *exact* will not detect, as discussed previously. Thus, this scope does a fuzzy search using the Levenshtein distance [58]. Levenshtein distance measures the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another, typically used for string similarity. This metric will match the closest sentence with a minimum of $N/2$. Non-alphanumeric characters are also not considered. At this level, the sentence from Figure 13 would match.

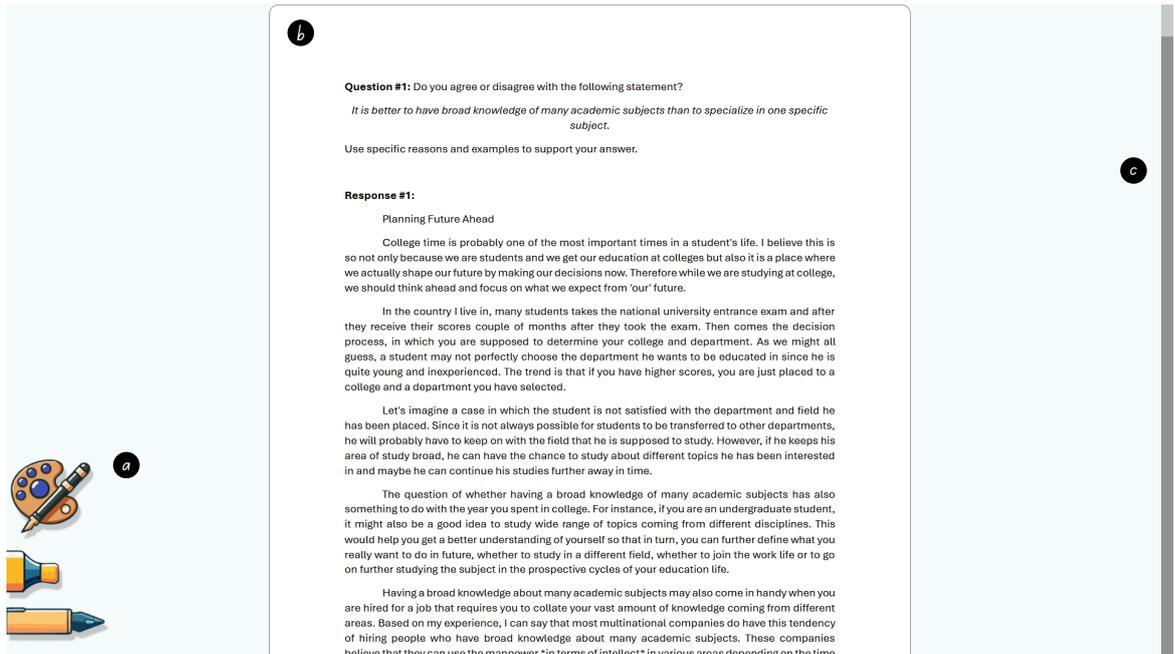


Figure 14: Prototype interface of AnnotateGPT comprises three components: (a) the toolbar (from top to bottom: colour palette, highlighter, and pen), (b) the document, and (c) the specialized scrollbar.

The order of scopes starts at single-page, then moves to cross-page, while the order of levels starts at exact and then shifts to fuzzy. The scope/level proceeds if the text cannot be found using that scope/level. The same approach is applied to previously found texts to avoid duplicates that the assistant may output.

4.4 IMPLEMENTATION

All methods have been implemented in a web application using Next.js [104]. The official OpenAI API library [80] was used for executing prompts on gpt-4o for generating annotations and gpt-4o-mini for purpose inference. The interface comprises three components: the toolbar (which includes a colour palette, highlighter, and pen), the document, and a specialized scrollbar (see Figure 14). The toolbar is located on the non-dominant side of the user for simultaneous thumb and pen interaction for effective mode switching (DGI) [87].

4.4.1 Document Parsing

The implementation uses PDF (Portable Document Format), which is widely supported across platforms and devices. They also embed annotations such as highlighting and freehand markups. The interface parses the PDF into two layers: (1) the image layer, where it displays an image for each document page, and (2) the text layer, where it displays text that is not visible to the user. The text layer aligns the position of the text with the image. It also parses the text into words where each word and character is marked by `` tags in HTML for

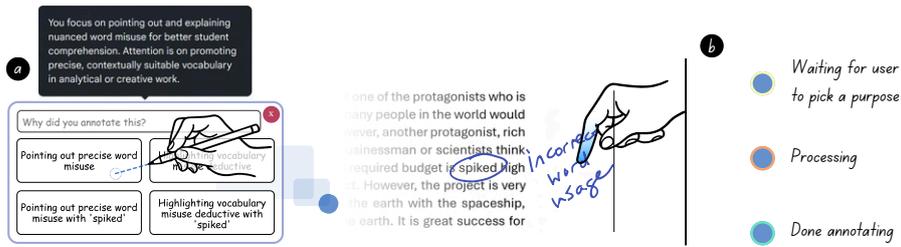


Figure 15: (a) Upon tapping an annotation, an assistant marker will appear and open on the left side of the document, displaying the potential purposes of said annotation. An input box is also provided for the user to type a purpose. Hovering over the option will provide more details about the purpose. (b) The assistant marker has three possible states: *waiting for purpose* (yellow), *processing* (orange), and *done* (green).

text extraction (Section 4.3.2) and displays AnnotateGPT’s annotations (Section 4.3.4).

4.4.2 Assistant Marker

Using the clustering algorithm (Section 4.3.1), an assistant marker is assigned to each cluster, initially hidden from the user (DG4). The assistant marker enables users to interact with AnnotateGPT. To activate the assistant, the user must tap on the cluster representing the annotation with their finger. Once tapped, the marker appears on the user’s non-dominant side of the document (see Figure 15a). Meanwhile, the cluster/annotation is processed as described in Section 4.3.3, and the marker’s border turns orange.

Once the images are processed, the marker displays the four guesses, and its border turns yellow to inform the user to pick a purpose. An input box is also provided in the marker if none of the options satisfy the user’s expectations. Once the user picks a purpose, the assistant will further annotate the document as described in Section 4.3.4. While the assistant is processing, the user can continue to annotate the document.

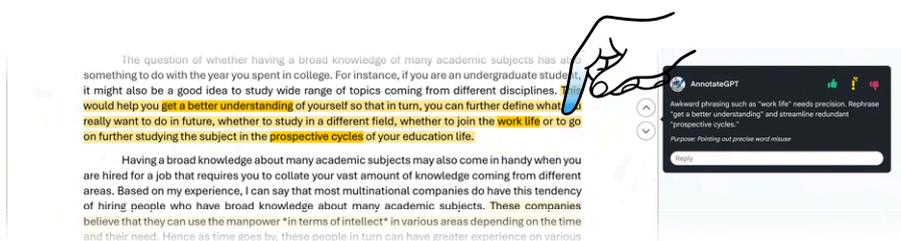


Figure 16: Tapping on the highlights will display a tooltip, where users can read the associated feedback by AnnotateGPT. Additionally, they can reply to continue the feedback. To verify the annotation, three ratings are given (left to right): *accept*, *helpful*, and *reject*.

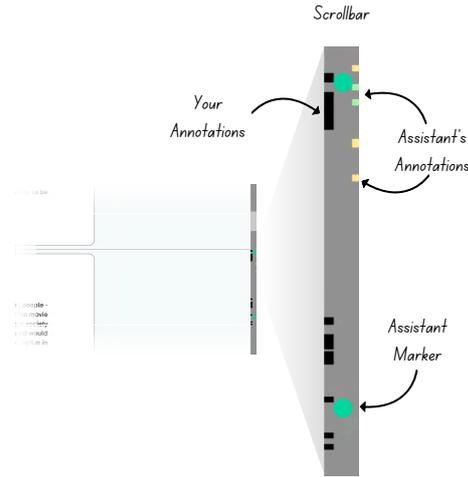


Figure 17: A specialized scrollbar to display the location and state of the annotations and assistant markers.

4.4.3 Verifying AnnotateGPT's Annotations

Lastly, the user will review the annotations made by AnnotateGPT. AnnotateGPT's annotations appear in **yellow** highlights, whereas the darker **yellow** highlights are the ones being targeted, as seen in [Figure 16](#). Upon tapping, a tooltip will display the associated feedback for the sentence. Sometimes, a sentence can be annotated multiple times from separate assistant invocations, in which the tooltip will display all feedback.

The user can give one of three ratings inside the tooltip (DG5). (1) *accept* if the annotation matches the purpose. Otherwise, (2) *reject*. (3) If the annotation gave helpful feedback but does not match the purpose, they can rate it *helpful*. Rating it *accept* or *helpful* will change the highlights to **green** (and dark **green**), while *reject* will remove the annotation entirely.

A reply box is also provided in the tooltip to continue annotating AnnotateGPT's comments and feedback. The arrow buttons beside the tooltip will navigate to the previous or next annotation. Alternatively, the user can tap another annotation.

4.4.4 Specialized Scrollbar

A specialized scrollbar is implemented to further assist in navigating the annotations, showing the locations and states of annotations and assistant markers. The scrollbar is separated into three columns: left, middle and right (see [Figure 17](#)).

LEFT Shows the location and colour of the user's annotations.

MIDDLE Shows the state (based on colour, see [Figure 15b](#)) and location of the assistant markers.

RIGHT Shows the location and state (*accept*, *helpful*, or *unrated*) of the assistant's annotations.

4.5 SUMMARY

This chapter introduces *AnnotateGPT*, a document annotation tool with a digital pen. It leverages an LLM (1) to infer the purposes of annotations and (2) automatically makes annotations with the same purpose across the document. It also focuses on a specific context, marking English essays, as it enhances a student-centred approach to learning and is important for students, but differs from other feedback forms [7]. However, annotations must be used and received appropriately [7]. This approach can be adapted to different contexts by changing the prompts. Overall, *AnnotateGPT* aims to alleviate the burdens of manual annotation, allowing users to focus on tasks that require critical expertise. The next chapter will detail how *AnnotateGPT* will be evaluated.

5

EVALUATION

Human-computer interaction (HCI), the field to which this thesis belongs, emphasizes the importance of designing with people in mind. Users' perceptions and thoughts regarding the system should be taken into consideration when evaluating it. Through a participant study, this work addresses the latter two research questions posed in [Chapter 1](#). In particular, (RQ1) whether LLMs can infer the purposes behind annotations and (RQ2) whether they can be used for comprehensive and continuous document annotation to address these questions thoroughly, various studies or longitudinal research must be conducted to test different annotation contexts and to analyze whether they can adapt to diverse annotation styles, as described in [Chapter 2](#). However, this falls outside the scope of this work. Instead, a smaller-scale user study was conducted to test the preference between traditional annotation interfaces and AnnotateGPT, as well as to gather self-reported ratings to evaluate its performance in marking English essays.

5.1 USER STUDY

To assess whether AnnotateGPT can answer the research questions, a user study was conducted to evaluate its effectiveness compared to traditional digital annotation interfaces. The baseline of comparison against AnnotateGPT was free-form, digital ink annotations (i.e., AnnotateGPT without the assistant marker). The study evaluated the usability, workload, and self-reported ratings for providing feedback on English essays using free-form annotations. The user study complies with the regulations of the Research Ethics Board #18136.

5.1.1 *English Tests Creation*

The data comes from essays written in response to two standardized English tests, created using the ETS Corpus of Non-Native Written English [11]. This dataset comprises 12,100 essays written by non-native English speakers as part of the TOEFL (Test of English as a Foreign Language) international exam for academic English proficiency, which was conducted in 2006 and 2007. Each essay is categorized into one of eight questions (e.g., "Young people enjoy life more than older people do") associated with a low, medium, or high score level.

Essays with a high score level were randomly selected from each question and compiled into a Microsoft Word document. A spell and grammar checker was applied to reduce the number of superficial annotations. The content is then exported into a PDF file. This process was repeated for the second document curation and the training

document, which had only one question. The resulting PDFs were used to create the final study documents.

5.1.2 Participants

Six pre-service teachers and teachers (1 left-handed) were recruited, of whom four are aged 18–24 and 2 are aged 25–34 with a minimum proficiency level of upper-intermediate English (B2 on the CEFR). Participants were recruited using recruitment posters on campus and a mass email sent to students in the Faculty of Education. The participation criteria included full mobility of the hand and wrist to allow for handwriting with a pen, normal or corrected-to-normal vision (e.g., glasses or contact lenses), and the ability to read and comprehend English. Recruitment was conducted on a first-come, first-served basis. Participants received the equivalent of \$40 CAD for appreciation of their time.

Based on the screening questionnaire, participants had 1 to 4 years of teaching experience. Among them, four individuals teach in STEM education (two at the high school level), one teaches English and history, and one focuses on educational studies. Furthermore, participants shared their experiences with text annotation: for paper, 1 does it daily, 3 weekly, 1 yearly, and 1 never; for digital annotation, 1 does it daily, 2 weekly, 1 monthly, 1 yearly, and 1 never. All but one annotate texts as part of their occupation.

5.1.3 Apparatus & Software

Using the Surface Pen, participants sat before the Microsoft Surface Studio 2 (28" touchscreen). They adjusted the screen until they felt comfortable with its position. The web/study application ran locally, and video recordings were made of the screen and the back of their shoulders to address usability concerns and capture quotes (see [Figure 18](#)).

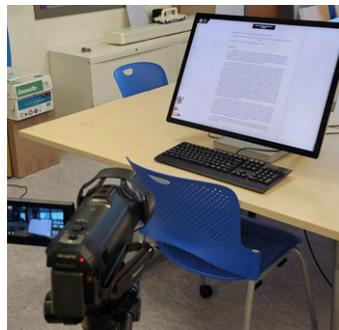


Figure 18: The setup used for the study

5.1.4 Study Design

The study followed a within-subjects study design with one primary independent variable, *TECHNIQUE*, with two levels (*ANNOTATEGPT* and *BASELINE*). *QUESTIONS* form secondary independent variables with 8 levels/questions. Each participant will mark an English test consisting of 8 *QUESTIONS* with each *TECHNIQUE*. The order of the *TECHNIQUE* is counterbalanced to minimize learning, where performance improves through practice with the task, and fatigue effects, where performance declines over time — half start with *BASELINE*.

The primary measures taken included the strokes made, annotation ratings (*accept*, *reject*, *helpful*) and interactions from the assistant

marker (purpose inference and generated annotations). Additionally, questionnaires and interviews provided subjective measures.

5.1.5 *Tasks & Procedure*

PRE-STUDY & INSTRUCTIONS Upon arriving, participants were first briefed on the work’s objective. They read and signed a consent form, confirming their understanding and agreement to participate in the study. Participants were then instructed to adjust the screen until comfortable and completed the demographics questionnaire (on Google Forms).

TRAINING Participants watched a video tutorial (~2 minutes for the *BASELINE*, and another ~2 minutes for *ANNOTATEGPT*) on the features of each technique. They then practised the technique on the training document. The following tasks must be completed before proceeding (*BASELINE* only has the first task): (1) Make an annotation. (2) Activate the assistant marker. (3) Generate additional annotations using the marker. (4) Navigate the annotations. (5) Verify the annotations. Participants can continue to practice until they are comfortable.

ANNOTATING Participants had 30 minutes to mark and annotate the English test by giving feedback, ensuring realistic working conditions, consistent time constraints across techniques, and reduced fatigue effects. For *ANNOTATEGPT*, digital ink is disabled after 25 minutes to allow them to finalize their ratings (*accept, reject, helpful*). A 5-minute warning was given before it was disabled. Another warning was given at the 30-minute mark to wrap things up. The same warnings were given for the *BASELINE*, but at the 25- and 30-minute mark to finalize their annotations. Participants were not required to finish annotating all questions, as the priority is to evaluate the overall experience and performance rather than task completeness.

QUESTIONNAIRES After each technique, participants completed questionnaires, including NASA Task Load Index (*TLX*) [41, 42] and System Usability Scale (*SUS*) [15]. *TLX* measures workload and the cost of accomplishing tasks across six dimensions: mental, physical, and temporal demands, frustration, effort, and performance. Participants answered six questions based on their self-assessment. On the other hand, *SUS* measure the system’s usability by covering its effectiveness, efficiency and satisfaction across ten questions. The study used a 7-point and 5-point Likert scale for *TLX* and *SUS*, respectively. Additional questionnaires asked about the participants’ self-perceived performance in evaluating the test and the features of *AnnotateGPT*.

INTERVIEW A post-study semi-structured interview was conducted to gather information about (1) which system they prefer and to provide examples of why and (2) what issues they sought during the test. Additional questions were asked based on observed participants’

trends, such as annotation behaviours and comments on feedback quality. The study took approximately 90 minutes to complete.

5.2 DATA ANALYSIS

The analysis employed an inductive thematic analysis to examine all collected data, including annotations, system logs, questionnaires, observational notes, screen recordings, and audio recordings of interviews. To complement the qualitative analysis, statistical analyses were conducted using the Wilcoxon signed-rank test (SciPy v1.15.3), and ART ANOVA [110] with its post-hoc tests were conducted using R with Benjamini–Hochberg correction. Non-parametric tests are used due to violation of normality and small sample size. Confidence intervals were calculated using bootstrapping with 1000 resamples.

5.3 RESULTS

WORKFLOWS OBSERVED There were three common workflows regarding how participants manually annotate to collaborate with AnnotateGPT:

1. **ANNOTATE-INTERPRET**: Participant makes one annotation, then has the assistant interpret it and waits for the result.
2. **ANNOTATE- N -INTERPRET- N** : Participant makes N annotations, then has the assistant interpret them concurrently.
3. **ANNOTATE-FOLLOWUP**: Participant makes an annotation to fill in the gaps for the assistant.

Participants ($N = 3$) will follow **ANNOTATE- N -INTERPRET- N** to initially focus on making annotations as the assistant processes previous ones, or they first annotate the entire question and then allow the assistant to interpret multiple annotations afterward. The other participants initially followed the **ANNOTATE-INTERPRET** workflow. Participants ($N = 3$) have also used **ANNOTATE-FOLLOWUP** to address anything AnnotateGPT has missed, where they have manually annotated anything AnnotateGPT did not highlight, as seen in [Figure 19](#).

In addition to the annotation workflow, participants have three common workflows on how they interact with AnnotateGPT:

1. **GENERATE-VERIFY**: Participant generates additional annotations based on one of their annotations and verifies afterwards.
2. **GENERATE- N -VERIFY**: Participant generates additional annotations based on N of their annotations and verifies all at once.
3. **VERIFY-COMMENT**: Participant will verify and continue to comment on the annotation.

Participants used a mix of **GENERATE-VERIFY** and **GENERATE- N -VERIFY** to generate and verify annotations made by AnnotateGPT. Participants

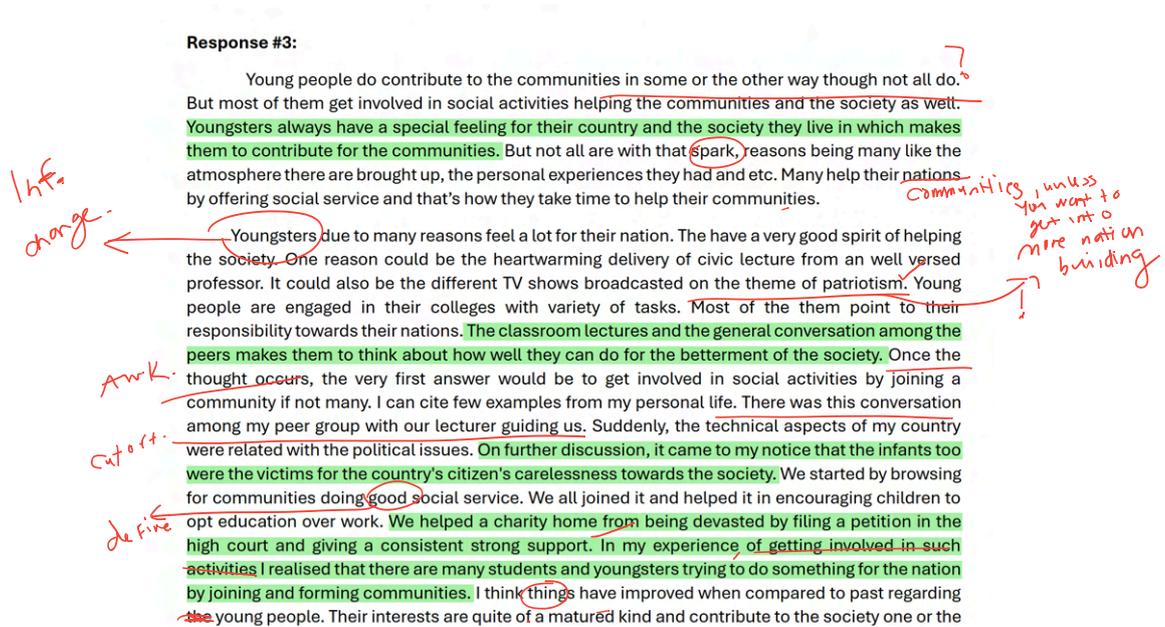


Figure 19: Example of how a participant fills in the gaps of the automated annotations, where the annotations were filling around the highlights.

($N = 3$) used GENERATE-VERIFY to provide feedback on the first question and then propagate the same feedback to the other questions using AnnotateGPT one at a time. Participants ($N = 5$) who used GENERATE- N -VERIFY to first manually annotate the first question without using AnnotateGPT, then interpret and generate at the end. Participants ($N = 4$) used VERIFY-COMMENT to continue or fix AnnotateGPT's feedback during the verification process.

TYPES OF ANNOTATION The types of annotations can be placed into two dimensions: *form*, which ranges TELEGRAPHIC-EXPLICIT, and *purpose*, which ranges MICRO-MACRO (see Table 4). The TELEGRAPHIC-EXPLICIT range describes whether the annotations are personal opaque codings versus explicitly textual feedback. The MICRO-MACRO range describes whether the annotation targets fine-grained textual features (such as spelling) versus broader, structural aspects of the text (such as logical flow). Participants have used a combination of these types of annotations to give feedback on the English tests. In total, 310 and 73 annotations were manually made for the BASELINE and ANNOTATEGPT, respectively.

ANNOTATION BEHAVIOURS Participants were also asked about their annotation behaviours, as it was observed that telegraphic annotations were used more for ANNOTATEGPT. Three participants (P1, P4 and P5) exhibited drastic differences in annotation behaviours between the two techniques (see Figure 20 and Figure 21), where they were significantly more telegraphic in their annotations with AnnotateGPT compared to their baseline. For those participants, there was minimal manual textual feedback. They noted that they "felt more responsible"

Category	Subcategory	Description	N_{Base}	N_{GPT}	N_{Gen}	Example
Grammar	Tense	Correct and consistent use of verb forms.				
	Preposition	Proper use of prepositions in phrases and expressions.	78	13	4	'S IN JUL
	Punctuation	Accurate use of commas, periods, etc.				the
	Capitalization	Correct use of uppercase or lowercase letters.				
Vocabulary	Word Choice	Suggesting appropriate, precise, and context-fitting words.				at a 20 years older me I would g itself just as the young me dt
	Spelling	Correct spelling of words.	52	13	6	myself travels in 20 years
	Collocation	Suggesting a better natural combination of words.				
Sentence Structure	Clarity	Suggesting sentences that are not easy to understand and are confusing.				
	Run-ons	Avoid long, improperly joined sentences.	107	29	8	13 reviews
	Fragment	Avoids incomplete sentence fragments.				
Organization & Coherence	Logical Flow	Ideas are not connected smoothly and lack a clear sequence.				
	Paragraphing	Proper use of paragraphs to group related ideas.	55	15	10	Conclusion?
Task Achievement	Completeness	Fully responds to the question or task requirements.	18	3	1	Appreciate the Small case study
	Encouragement	Provide supportive and motivating feedback where appropriate.				
Type	Telegraphic	A personal opaque coding.	70	31	—	see Figure 20 and 21
	Explicit	Clear and explicit meaning, usually textual.	240	42	—	

Table 4: Types of annotations observed in the study are placed into two dimensions: form and purpose. Each type has the number of counts observed for the baseline (N_{Base}), AnnotateGPT (N_{GPT}) and purpose inference (N_{Gen}).

Annotation	Description	Listed Purposes
	<p>The annotation consists of a pen crossing out the word 'now', indicating that the user intends to delete or correct this word. This action suggests that it is deemed unnecessary or incorrect in the context.</p>	<ol style="list-style-type: none"> 1. Language Correction 2. Writing Improvement Guidance 3. Language Correction ("now") 4. Writing Improvement Guidance ("now")
<p>vast amount of knowledge coming from different</p>	<p>The annotation shows a crossing out of the phrase 'coming from different', indicating a need for clarity and precision in language use.</p>	<ol style="list-style-type: none"> 1. Promoting Clear Language 2. Fostering Coherence in Writing 3. Promoting Clear Language (word-specific) 4. Fostering Coherence in Writing (word-specific)
<p>Break your essay into paragraphs to separate your thoughts</p>	<p>The annotation is a handwritten note in red pen stating, 'Break your essay into paragraphs to separate your thoughts.' This instruction emphasizes the importance of organizing the essay clearly.</p>	<ol style="list-style-type: none"> 1. Improving Essay Structure 2. Constructive Criticism 3. Emphasizing Clarity 4. Fostering Skill Development
<p>about sports - sports is a rather popular pastime activity for the younger generation. And it's also an important part of older people's lives, a point that is supported by the fact that certain activities that have disappeared the older generation first but have now moved on to include the younger generation as well. (Without of course, losing any importance, to the older generation, it's like a safety-side activity) to kids. Sports can therefore be enjoyed by both, the young and the old, and does not leave</p>	<p>The annotation consists of a circle around the phrase [...] This indicates that the user is focusing on this passage for scrutiny during the grading process.</p>	<ol style="list-style-type: none"> 1. Encouraging Critical Analysis 2. Highlighting Key Content for Revision 3. Promoting Clarity in Expression 4. Identifying Areas for Conceptual Clarity 5. Run On Sentence
<p>those who by new things and take risks for their future.</p> <p>The general that currently play even shows that it is good to have risky guests for greater rewards. Important fact is that trying the risky way for more experience and greater success rather</p>	<p>The annotation consists of a line circling the phrase [...] The use of a circle emphasizes significance, indicating it's noteworthy in the context of the English test.</p>	<ol style="list-style-type: none"> 1. Emphasizing Life Lessons 2. Focus on Clarity 3. Inspiring Courage 4. Critical Analysis 5. Not Connected to the Answer

Table 5: Examples of purpose inference along with its annotation description. Green text indicates AnnotateGPT correctly inferred the purpose, while red text indicates participants manually typed the purpose.

P1

Question #1: Do you agree or disagree with the following statement?
It is better to have broad knowledge of many academic subjects than to specialize in one specific subject.

Use specific reasons and examples to support your answer.

Response #1:

I think that the question to either have a broad knowledge of a variety of subjects or to have a good and specialized understanding in one subject is a difficult one. I agree with both of the points to some extent. First of all, it is important for a younger student to have a broad knowledge of many subjects. Schools, in particular primary and secondary schools should try and give their students a broader overview of many subjects. This is also important in the younger years of a student's life, because most of the students have not decided what they will do in their future life. They are not really sure what they are really interested in and in what area they want to specialize. When it comes to secondary schools, students should be given their first choices, based on the knowledge they have already obtained. They should be able to limit their subject choice and cut down on the number of areas where they have to study in. Of course, this would also result in deepening the knowledge in the subjects chosen. When it comes to the last two to three years of school for students, they should be given lessons in only the six to seven areas they are really interested in. Whereas, I think that certain subjects do not seem to have to be taken. Currently, Mathematics, German, a second language, History or Politics and Biology, Chemistry or Physics have to be studied. I think this is not needed, as different students of course have different strengths, not only in the, but also in their academics. I have attended a German "Gymnasium" until Year 10, where we were not able to make any choices whatsoever. Studying 16 to 18 subjects was not an exception. I have also attended an Australian school, only studying 5 subjects. The choice was widely open and no Mathematics or language or any subjects were required to be able to do your exams at the end and gain university entrance. However, for certain university courses, students also had to take certain school courses. One such example would be Medicine, where students had to have Biology and Chemistry, as well as taking the UMAT Test, to see if they are suitable for such a course and also later on such a job. I think this is a successful approach for students, which not only gives them the motivation they need in school, but also the requirements to successful results at the end of their school time. On the side, we also have university students who have to be taken under consideration. Of course they are a different matter altogether. I hold the opinion that the less subjects they take and the deeper they go into details in their study, the better it is. At the beginning of a three to four year study period for a degree in that area, university students should get a bit of a broader overview of the area for one to two semesters. In that period of time, they get the time needed to see what specifically they want to do and with what exactly they want to continue. The rest of the remaining study period, university students should be prepared for exactly the field they are interested in and want to get a working place later on. This is necessary to make sure that every one of the students is really good at what they are doing and can prepare well for their choice of job. This would give the university students not only more choices from which they can choose their courses, but it would also prepare them better for what they will do later in their life. It would also be a positive criteria for employers when they take people, because then they can be sure that they know what the person is doing and that the person

P2

Question #1: Do you agree or disagree with the following statement?
It is better to have broad knowledge of many academic subjects than to specialize in one specific subject.

Use specific reasons and examples to support your answer.

Response #1:

Planning Future Ahead

College time is probably one of the most important times in a student's life. I believe this is so not only because we are students and we get our education at colleges but also it is a place where we actually shape our future by making our decisions now. Therefore while we are studying at college, we should think ahead and focus on what we expect from 'our' future.

In the country I live in, many students takes the national university entrance exam and after they receive their scores couple of months after they took the exam. Then comes the decision process, in which you are supposed to determine your college and department. As we might all guess, a student may not perfectly choose the department he wants to be educated in since he is quite young and inexperienced. The trend is that if you have higher scores, you are just placed in a college and a department you have selected.

Let's imagine a case in which the student is not satisfied with the department and field he has been placed. Since it is not always possible for students to be transferred to other departments, he will probably have to keep on with the field that he is supposed to study. However, if he keeps his area of study broad, he can have the chance to study about different topics he has been interested in and maybe he can continue his studies further away in time.

The question of whether having a broad knowledge of many academic subjects has also something to do with the area you spent your college. For instance, if you are an undergraduate student, it might also be a good idea to study wide range of topics coming from different disciplines. This would help you get a better understanding of yourself so that in turn, you can further define what you really want to do in future, whether to study in a different field, whether to join the work life or to go on further studying the subject in the prospective cycles of your education life.

Having a broad knowledge about many academic subjects may also come in handy when you are hired for a job that requires you to collate your vast amount of knowledge coming from different areas. Based on my experience, I can say that most multinational companies do have this tendency of hiring people who have broad knowledge about many academic subjects. These companies believe that they can use the manpower "in terms of intellect" in various areas depending on the time and their need. Hence as time goes by, these people in turn can have greater experience on various work-related subjects than their colleagues who have less broad knowledge about these academic topics and become more qualified and desired employees.

Nevertheless, studying wider range of topics and having a broader knowledge of many academic subjects might not be a good idea if you are a graduate student. This is because the aim of

P3

Question #1: Do you agree or disagree with the following statement?
It is better to have broad knowledge of many academic subjects than to specialize in one specific subject.

Use specific reasons and examples to support your answer.

Response #1:

Planning Future Ahead

College time is probably one of the most important times in a student's life. I believe this is so not only because we are students and we get our education at colleges but also it is a place where we actually shape our future by making our decisions now. Therefore while we are studying at college, we should think ahead and focus on what we expect from 'our' future.

In the country I live in, many students takes the national university entrance exam and after they receive their scores couple of months after they took the exam. Then comes the decision process, in which you are supposed to determine your college and department. As we might all guess, a student may not perfectly choose the department he wants to be educated in since he is quite young and inexperienced. The trend is that if you have higher scores, you are just placed in a college and a department you have selected.

Let's imagine a case in which the student is not satisfied with the department and field he has been placed. Since it is not always possible for students to be transferred to other departments, he will probably have to keep on with the field that he is supposed to study. However, if he keeps his area of study broad, he can have the chance to study about different topics he has been interested in and maybe he can continue his studies further away in time.

The question of whether having a broad knowledge of many academic subjects has also something to do with the area you spent your college. For instance, if you are an undergraduate student, it might also be a good idea to study wide range of topics coming from different disciplines. This would help you get a better understanding of yourself so that in turn, you can further define what you really want to do in future, whether to study in a different field, whether to join the work life or to go on further studying the subject in the prospective cycles of your education life.

Having a broad knowledge about many academic subjects may also come in handy when you are hired for a job that requires you to collate your vast amount of knowledge coming from different areas. Based on my experience, I can say that most multinational companies do have this tendency of hiring people who have broad knowledge about many academic subjects. These companies believe that they can use the manpower "in terms of intellect" in various areas depending on the time and their need. Hence as time goes by, these people in turn can have greater experience on various work-related subjects than their colleagues who have less broad knowledge about these academic topics and become more qualified and desired employees.

Nevertheless, studying wider range of topics and having a broader knowledge of many academic subjects might not be a good idea if you are a graduate student. This is because the aim of

Figure 20: Screenshots of the first page for P1–3. The left side is with the baseline, and the right side is with AnnotateGPT. It shows different annotation approaches where P1 was more focused on identifying issues rather than providing textual feedback.

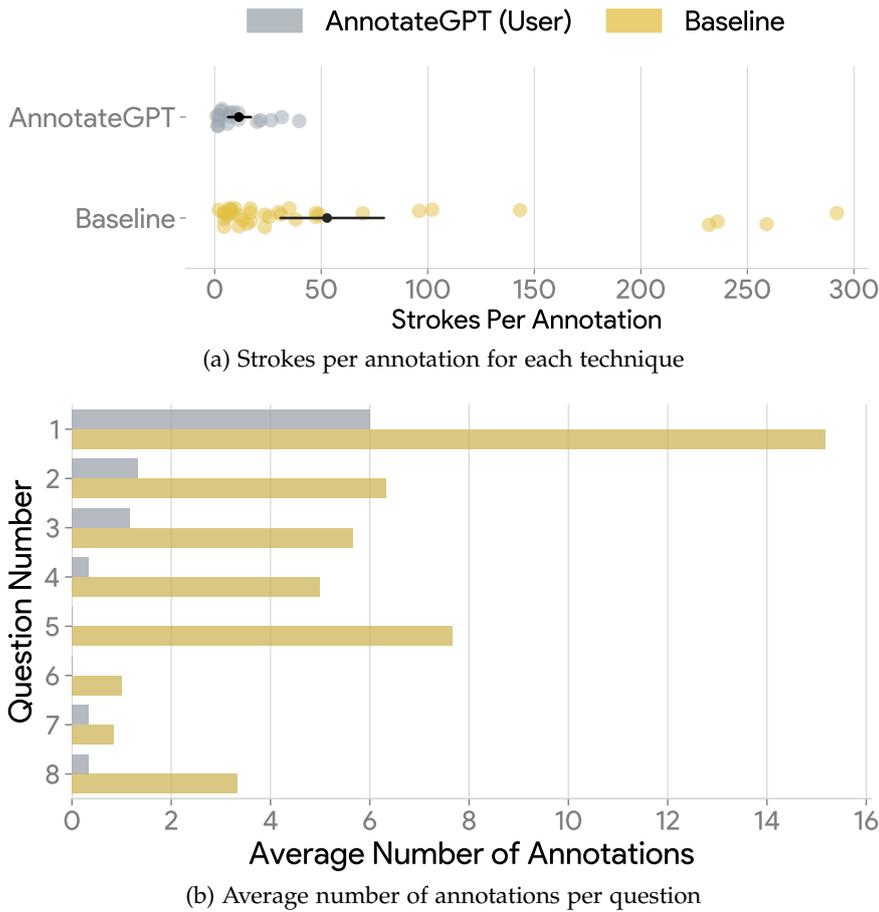


Figure 22: Plots representing the inter- and intra-density of annotations. Only user-authored annotations are included. It shows participants used more explicit annotations for the baseline and focused more on the first question.

[P1] when using the baseline and “letting [AnnotateGPT] to come up with the comments” [P4], suggesting AnnotateGPT offloaded some of the responsibility. This suggests participants were relying on AnnotateGPT to determine the purpose of the annotation without the “need to add [explicit] comments because [AnnotateGPT] would write it for me” [P5], using the underlying text as context. For example, a participant noted, “I like how it can detect highlights [...] If I select the highlight, it will recognize what I highlighted and give me feedback” [P2]. Despite the annotations being telegraphic, AnnotateGPT was able to pinpoint the purposes of the annotations (see Table 5).

MISCLASSIFICATIONS There were instances where AnnotateGPT failed to infer the purpose of the annotations accurately. Out of 29 inferences, 9 were manually typed by the participants. All of the failed cases involved telegraphic annotations, causing the LLM to produce vague and general descriptions of the annotations. This guided the LLM to make very broad guesses, which would not suffice to pinpoint the purpose (see Table 5).

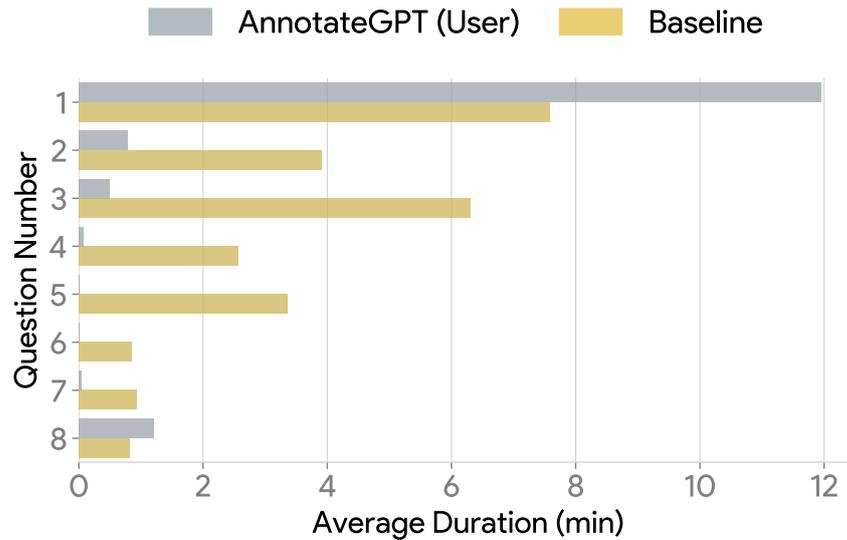


Figure 23: Plot for annotating duration for AnnotateGPT and baseline for each question. Only user-authored annotations are included. It shows participants spent the most time on the first question.

DENSITY OF ANNOTATIONS The inter- and intra-density of annotations significantly differ between **TECHNIQUE** and **QUESTIONS** (see [Figure 24](#)). ART ANOVA revealed a significant main effect of **TECHNIQUE**, with an average of 53 strokes (95% CI: [30, 80]) and 11 strokes (95% CI: [6, 17]) per annotation for **BASELINE** and **ANNOTATEGPT**, respectively (see [Figure 22a](#)). This suggests telegraphic annotations were used significantly more for **ANNOTATEGPT**.

Regarding the number of annotations per question, ART ANOVA revealed significant main and interaction effects between **TECHNIQUE** and **QUESTIONS**. Participants focused more on the first question than on any other question (see [Figure 22b](#)). This suggests that fatigue effects were present after the first question. However, there were significant effects only for question 4. The same trend is observed for **ANNOTATEGPT**, but this is due to the first question being the primary interaction space of the assistant, with subsequent questions serving as follow-ups. More details on the statistical tests can be found in the Appendix (see [Section A.2](#)).

ANNOTATING DURATION The annotating duration for each question is the time elapsed from the first to the last stroke being drawn on said question. ART ANOVA revealed a significant main effect and interaction effects between **TECHNIQUE** and **QUESTIONS**. Again, only significant effects are for question 4. However, considering the number of annotations with the annotating duration, participants were faster in making annotations on the first question, further suggesting that fatigue effects were present after the first question. Additionally, participants spent more time on the first question for **ANNOTATEGPT** than for **BASELINE** due to the diverse workflows described earlier, where participants verify suggestions made by AnnotateGPT and then continue to annotate the first question. More details on the statistical tests can be found in the Appendix (see [Section A.2](#)).

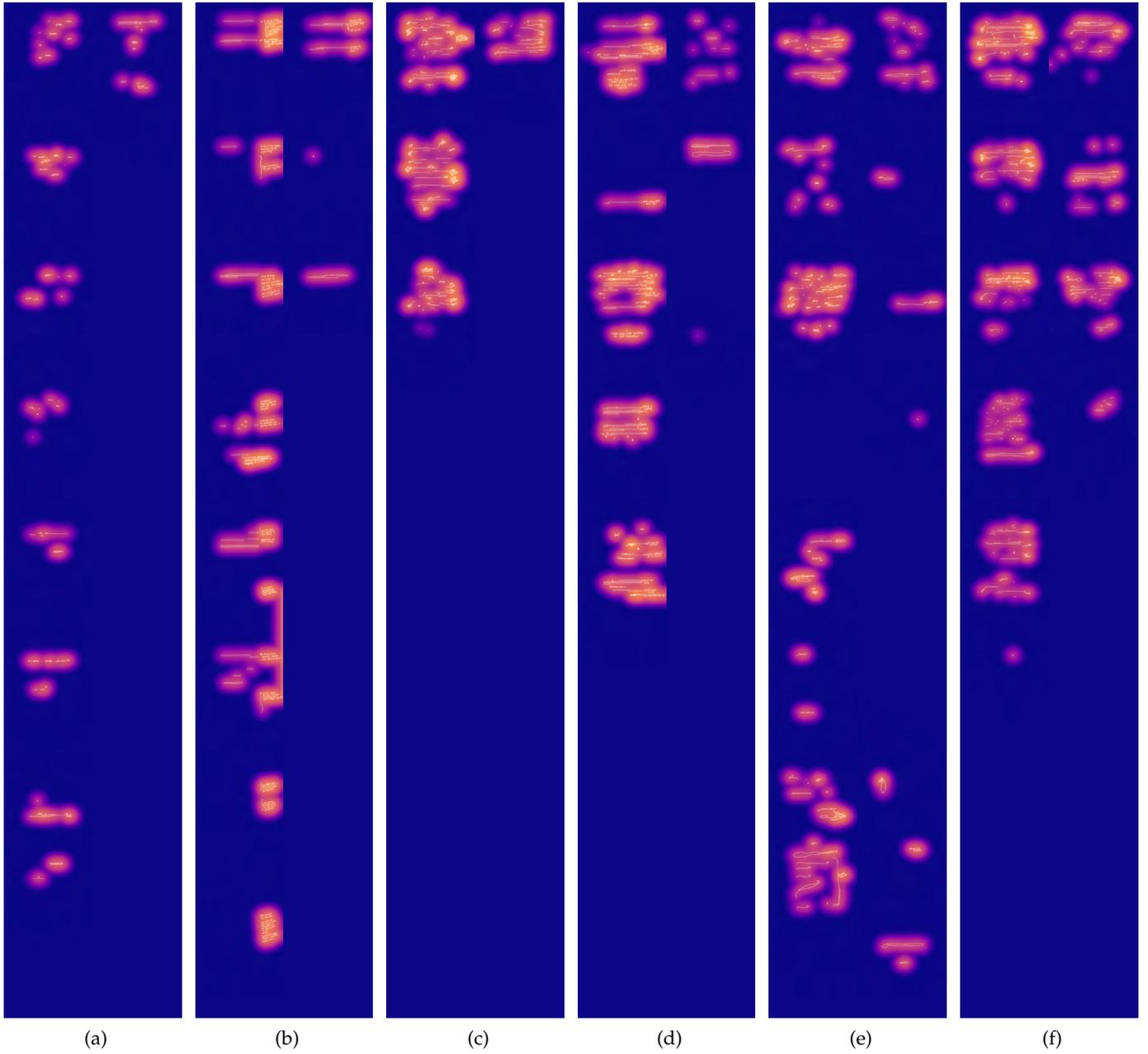


Figure 24: Heatmaps of each participant’s annotations, where the left heatmap of each figure is with the baseline and the right is with AnnotateGPT. It shows that significantly more user annotations are made for the baseline.

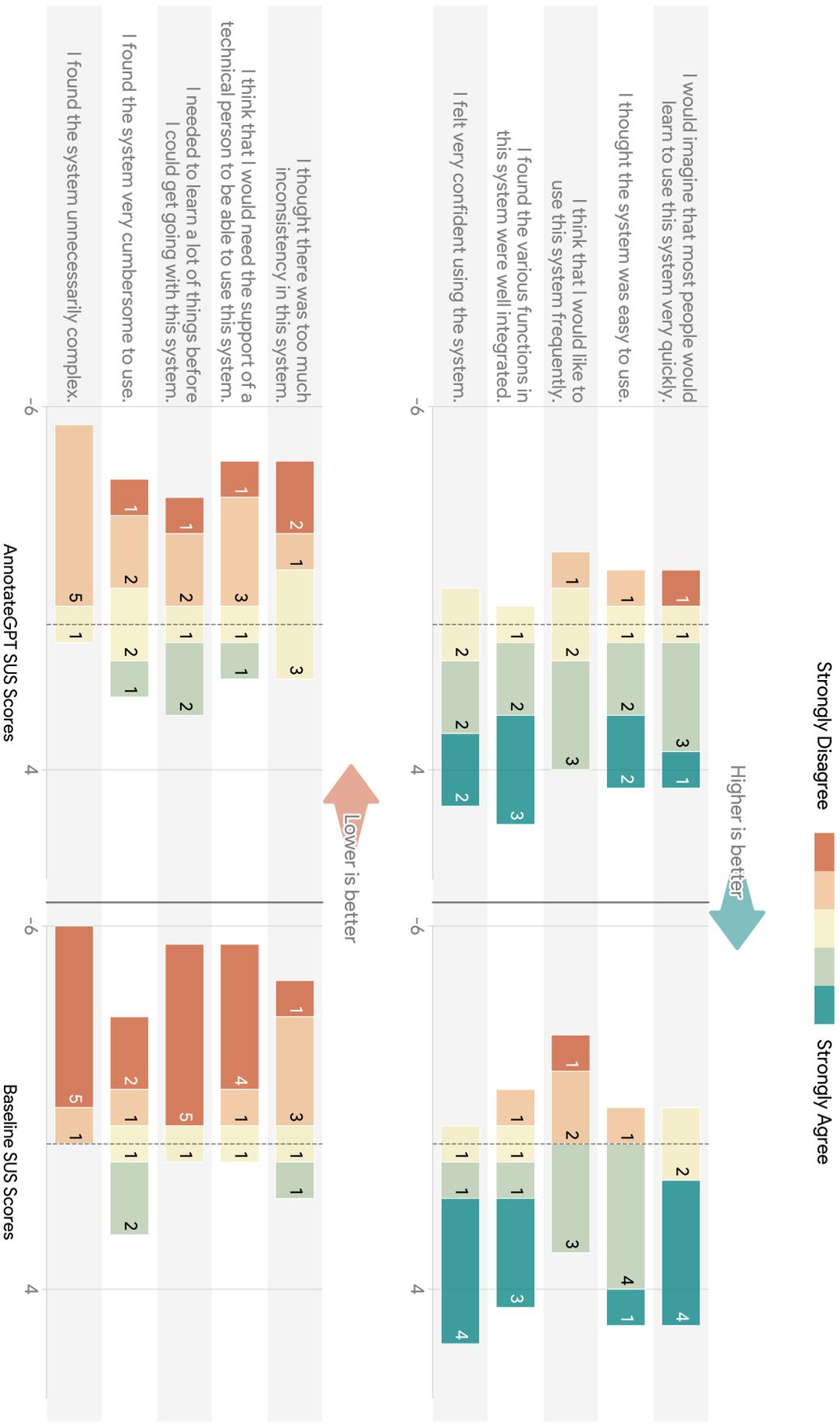


Figure 25: Plot representing the SUS scores across AnnotateGPT and baseline. Both techniques had similar usability.

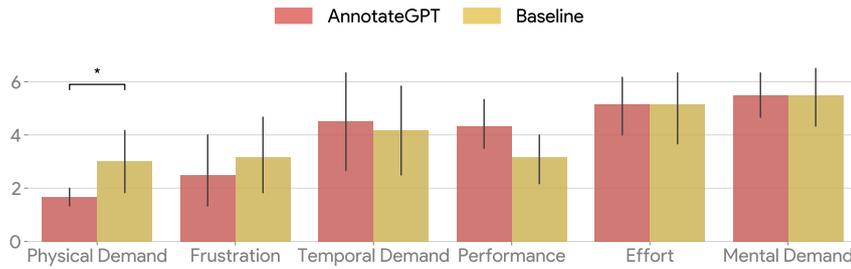


Figure 26: Plot representing the TLX scores across AnnotateGPT and baseline. It shows that AnnotateGPT required a significant reduction ($p < 0.05$) in physical demand, while other subscales exhibit similar demands.

USABILITY On the System Usability Scale, ANNOTATEGPT was rated 68 (95% CI: [56, 79]), and the BASELINE was rated 77 (95% CI: [67, 85]), both typically considered to be rated “Good” [8] (see Figure 25). TECHNIQUE did not significantly affect usability ($Z = 2.0$, $p = 0.09$), suggesting ANNOTATEGPT has similar usability as BASELINE. Due to the small sample size, these findings are only indicative trends.

TASK LOAD INDEX NASA-TLX scores were collected to assess the perceived workload across six subscales on a 7-point Likert scale (Figure 26). Among the individual subscales, physical demand showed a statistically significant reduction ($Z = 0.0$, $p < 0.05$, $r = 0.62$) for ANNOTATEGPT ($Mdn = 2.0$, $IQR = 0.75$) over BASELINE ($Mdn = 2.5$, $IQR = 2.5$). No other subscales demonstrated statistically significant differences. These findings suggest that ANNOTATEGPT can meaningfully reduce physical workload due to participants creating more telegraphic annotations. Again, due to the small sample size, these findings should be interpreted as potential trends. More details on the statistical tests for each subscale can be found in the Appendix (Section A.2).

QUESTIONNAIRES Participants also rated AnnotateGPT’s features (Figure 27) and their overall experience in annotating the tests (Figure 28) on a 5-point Likert scale. Participants agreed that the automated annotations ($Mdn = 4.5$, $IQR = 1.0$) and feedback were very helpful ($Mdn = 4.0$, $IQR = 0.0$) and easy to understand ($Mdn = 4.0$, $IQR = 0.0$). Participants also agreed that AnnotateGPT was able to provide helpful guesses on the purpose of their annotations ($Mdn = 4.0$, $IQR = 0.75$). However, participants found that AnnotateGPT was unable to find all the issues ($Mdn = 2.5$, $IQR = 1.0$). These findings suggest that AnnotateGPT can accurately infer the purpose of its annotations and provide meaningful feedback to the identified sentences. However, there were some inconsistencies in targeting all issues.

Regarding their overall annotating experience, there were no statistical differences between TECHNIQUE. Participants found grading the test (ANNOTATEGPT: $Mdn = 3.5$, $IQR = 2.5$; BASELINE: $Mdn = 3.5$, $IQR = 1.75$) and providing feedback to be easy (ANNOTATEGPT: $Mdn = 3.5$, $IQR = 1.75$; BASELINE: $Mdn = 3.5$, $IQR = 1.00$). They

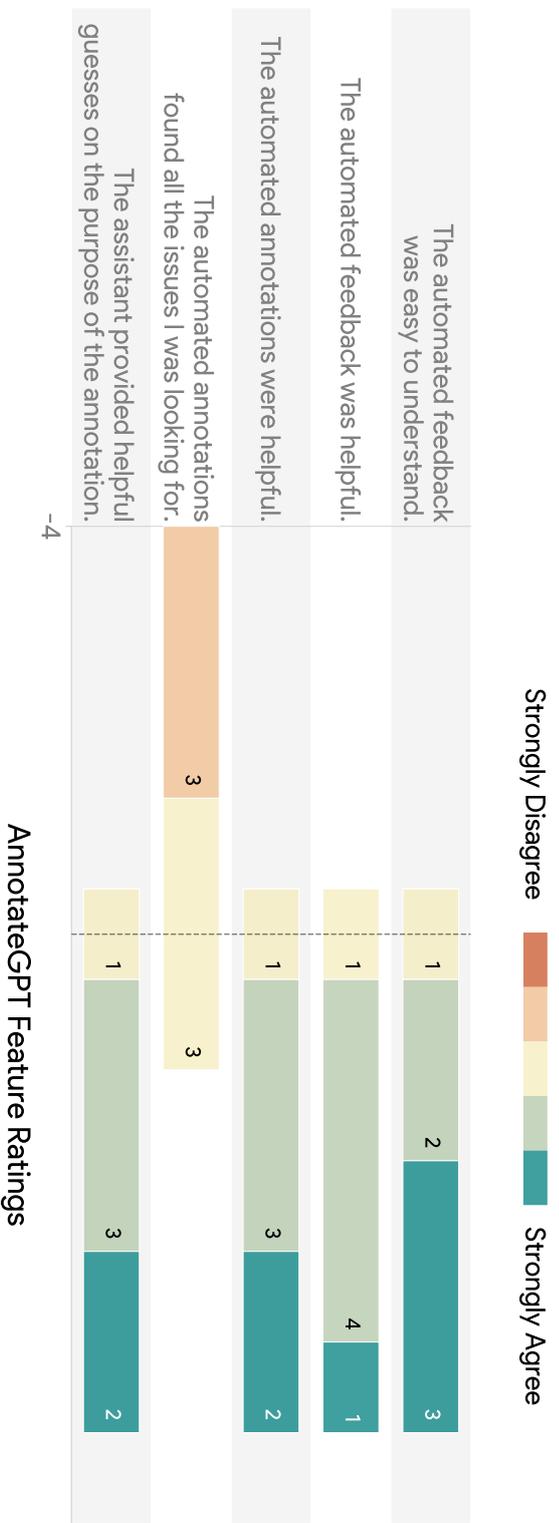


Figure 27: Questionnaire responses for AnnotateGPT’s features on a 5-point Likert scale. Participants agreed that AnnotateGPT accurately inferred the purpose of their annotations and provided helpful feedback. However, it was unable to identify all issues for a given purpose.

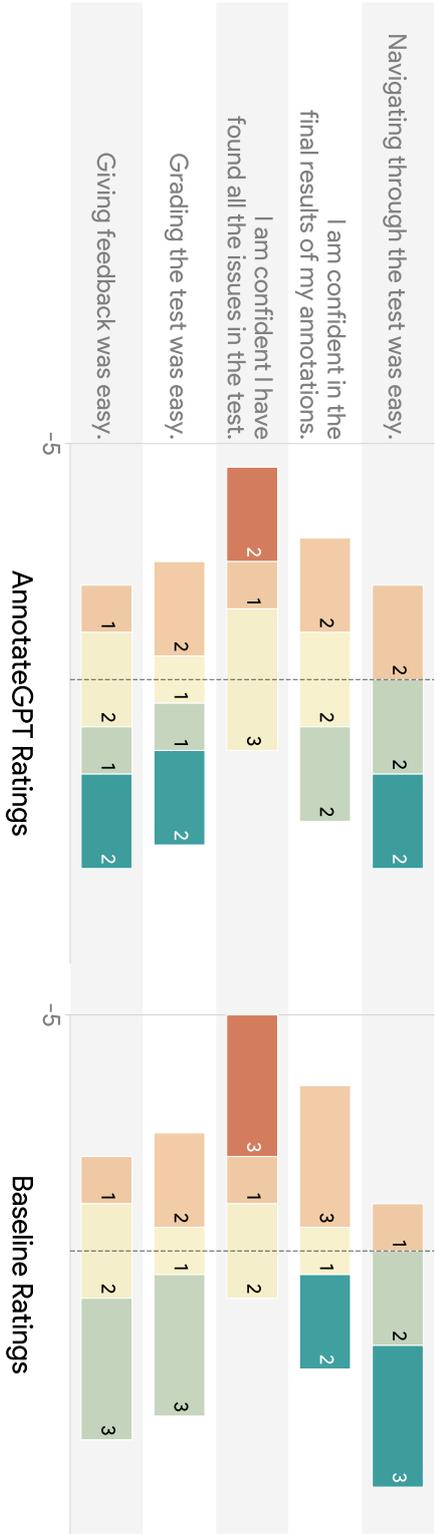


Figure 28: Questionnaire responses for the evaluation experience on a 5-point Likert scale. Participants found it easy to grade and provide feedback on the essay, but they are not confident that they found all the issues for both techniques.

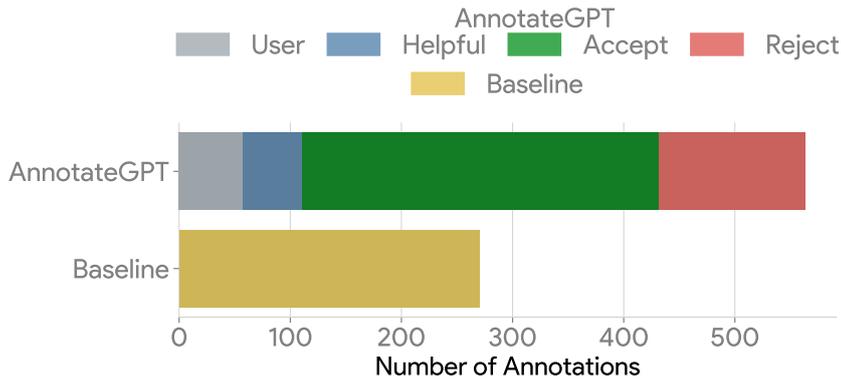


Figure 29: Number of annotations between AnnotateGPT and baseline, along with human-rated annotations for AnnotateGPT.

also reported comparable confidence in the final results of their annotations (ANNOTATEGPT: $Mdn = 3.0$, $IQR = 1.5$; BASELINE: $Mdn = 2.5$, $IQR = 2.5$) and in identifying all issues in the test (ANNOTATEGPT: $Mdn = 2.5$, $IQR = 1.75$; BASELINE: $Mdn = 1.5$, $IQR = 1.75$). Finally, both techniques were rated similarly in terms of ease of navigation (ANNOTATEGPT: $Mdn = 4.0$, $IQR = 2.25$; BASELINE: $Mdn = 4.5$, $IQR = 1.0$), indicating overall similarity in user experience across conditions.

FEEDBACK RATINGS Participants also verified each AnnotateGPT’s annotations. On average, there were 54 (95% CI: [32, 78]) *accepted* annotations, 21 (95% CI: [14, 30]) *rejected* annotations, 9 (95% CI: [3, 17]) *helpful* annotations and 10 (95% CI: [7, 12]) *user* annotations. ANNOTATEGPT significantly ($Z = 0.0$, $p < 0.05$, $r = 0.70$) made more annotations (see Figure 29).

FEEDBACK QUALITY Participants generally perceived AnnotateGPT as providing broader and more efficient support during the annotation process when compared to the baseline. AnnotateGPT was able to “look at a bigger scope” [P1] and “relying on [AnnotateGPT] to read it for me and find all those errors” [P5], suggesting a perceived enhancement in coverage. One participant noted they “graded more in total using [AnnotateGPT]” [P3], attributing this to the AI’s ability to “look at things I wasn’t able to and save time” [P1] under time constraints.

A common theme in the difference between each technique’s feedback is that the baseline only states what the problem is (e.g. “awk” or “reword”), while AnnotateGPT explicitly states the reason (see Table 6). Several participants acknowledged AnnotateGPT’s strengths in content-related feedback, with one stating that AnnotateGPT was particularly effective when they were “looking more at the content of answers” [P5] and another reflecting that “[AnnotateGPT] gave better feedback than I would write” [P4]. However, this benefit also introduced concerns about over-reliance. For example, one participant noted, “I liked the suggestions it gave for [AnnotateGPT], I didn’t find it always consistent” [P6] in finding the issues, while another reflected that they were letting “the AI would write for me in a way” [P5].

Category	Purpose	Example Feedback
Grammar	Grammar and Punctuation Issues	Grammar issue: Change “brought him in” to “brought him into.”
Vocabulary	Vocabulary Enhancement	Revise “having a blast” to a formal or descriptive term that suits the discussion’s tone.
Sentence Structure	Promoting Clarity in Expression	Sentence is overly complex and verbose; divide into smaller sentences for clarity and focus.
Organization & Coherence	Importance of Transitional Phrases	Use preceding transitions to connect the personal example to the broader argument smoothly. Example: “This pursuit often results in...”.
Task Achievement	Recognition of Strength	Emphasizes the emotional aspect of guided tours effectively. Specify how guides enable this emotional connection.

Table 6: Example feedback by AnnotateGPT from each category and has been accepted by the participants.

Participants generally found AnnotateGPT beneficial for creating faster and more in-depth feedback, particularly in complex or time-sensitive situations. However, they also emphasized the importance of balancing it with manual review. As one participant proposed, a better workflow might be to “*go through it and mark it first, and then use the assistance as the secondary tool*” [P5], which followed the workflows of P3 and P6. AnnotateGPT was regarded as a valuable support tool for surfacing major issues and enhancing the feedback process, especially when used in conjunction with manual annotation.

5.4 SUMMARY

A user study ($N = 6$) was conducted to evaluate AnnotateGPT, a system that leverages LLMs to generate feedback based on user annotations during the evaluation of English essays. Participants engaged with AnnotateGPT and a baseline annotation tool to evaluate essays, revealing a tendency to rely on the model for interpreting short or unclear annotations, but AnnotateGPT often accurately inferred user intent (RQ1). Participants acknowledged the high quality of feedback but noted some inconsistencies in coverage (RQ2). Usability ratings were comparable for both tools, with AnnotateGPT perceived to reduce the physical effort required for annotation. These insights indicate that LLM-based systems can significantly enhance human annotation by providing context-sensitive suggestions, particularly when user input is explicit and intentional.

DISCUSSION

This chapter examines the design implications of using annotations to guide LLMs in feedback workflows. It also introduces example scenarios on how it can be integrated into current annotation practices and possible applications outside of making sense of text. It concludes with a discussion of the study's limitations and future work.

6.1 DESIGN IMPLICATIONS

Study findings suggest several design directions for LLM-integrated annotation tools. These include using LLMs to augment feedback workflows, support educational equity for diverse educators, and rethinking annotation as a rich interaction paradigm for guiding AI output. Together, these insights suggest that users can collaborate with LLMs by using annotations.

6.1.1 *LLMs as Cognitive Augmentation in Feedback Workflows*

The user study suggests that AnnotateGPT functions as a cognitive augmentation, helping users transform telegraphic annotations into fully articulated feedback with ease. Users often relied on AnnotateGPT to extrapolate from little to no textual signals and found the model's interpretations aligned with their intentions. This aligns with previous works on LLM-integrated systems [26, 31, 35, 44, 91, 97, 116], where they aid in expression rather than idea generation, enabling users to concentrate more on the overall content and less on specifics. For example, Code Shaping [116] enables users to create and edit code by making annotations and sketches, shifting the programming process from a syntax-focused method to one that prioritizes code structure and flow.

In this context, AnnotateGPT collaborates with the teacher in the annotation process. Such systems can be especially valuable in scenarios with high annotation throughput (e.g., large class sizes or timely return time) by externalizing the labour of phrasing and formatting, and allowing educators to focus cognitive effort on identifying issues rather than constructing detailed responses. For example, participants marked a telegraphic annotation on a student essay, and AnnotateGPT returned a comment explaining the problems across paragraphs. The user confirmed it was what they intended by verifying the annotations, but would not have typed it out themselves. This demonstrates that LLMs are not just automation tools, but cognitive collaborators that can empower educators to focus on tasks that require critical expertise, provided the system maintains alignment with pedagogical intent.

6.1.2 *Support Equity in Education*

A crucial yet often overlooked impact of AI-driven systems is their capacity to support educational equity. In educational settings where classroom funding is limited, or a shortage of teachers results in teachers having limited time, training, or language ability to offer high-quality, detailed feedback to every individual student, AnnotateGPT could effectively replace telegraphic notes with richer feedback. This could be beneficial for teaching assistants or novice teachers who struggle to provide clear and effective feedback in a timely manner. For example, participants identified grammar and sentence structure issues most frequently during the baseline. Conversely, AnnotateGPT was used more to address organization and coherence issues, providing feedback on broader aspects of the text.

Previous research on human-AI collaboration indicates that LLMs can help level the playing field by enabling users access to domain-specific knowledge or communication skills they may lack [61, 102]. In educational contexts, where the quality of teaching can vary, AnnotateGPT can provide novice teachers with timely, accurate, and tailored feedback, especially in high-enrollment or under-resourced settings. For instance, novice teachers can guide AnnotateGPT with annotations and provide helpful, in-depth feedback aligned with the curriculum objectives they are assessing, especially for a large number of student submissions. The curriculum objectives could also be passed to the LLM to further align the feedback.

However, this benefit also has some limitations, where relying too heavily on LLMs can promote a standard way of speaking or overlook cultural differences, especially if these models are trained on data from dominant cultural or academic perspectives. Therefore, it's essential to design systems that adapt to diverse cultures and personalize the experience, ensuring that AI supports the teacher's voice rather than overshadowing it. For example, participants suggest marking the document as AI-free first, then allowing the AI to review it a second time. This was supported by earlier research indicating that revising an essay with AI tools, after writing without AI, led to greater ownership, better memory recall, and more strategic integration [53].

6.1.3 *Annotation as an Interaction Paradigm*

Our study highlights annotation as a central component, not just as a tool for annotation, but as an interaction paradigm for engaging with AI systems. While annotations have traditionally been private or instructor-focused, their combination with LLMs like AnnotateGPT transforms these annotations into expressive inputs that directly impact system behaviour and feedback. This aligns with a broader trend toward low-friction, in-context interfaces that leverage users' micro-actions (e.g., highlights, marginal notes, shorthand feedback) as signals to AI systems [4].

In our study, users frequently employed the system to “fill out” or elaborate brief annotations into full suggestions, effectively using annotations as a shared language between humans and AI. This aligns with recent research on intent-aware AI [67], wherein systems aim to infer more robust context from user cues, and follows from previous research that situates annotation as a space of interpretation and sense-making [68, 112]. Instead of requiring explicit and deliberate commands or form-based input, AnnotateGPT supports opportunistic interaction wherein users write openly and let the AI infer intent, thus reducing overhead and physical effort.

Future systems can build upon this paradigm by providing multi-modal annotations or real-time previews of AI inferences to further enhance transparency and accuracy. Ultimately, framing annotation as an interaction paradigm creates a promising design space for collaborative, intent-aware interfaces that augment rather than replace human input.

6.2 EXAMPLE SCENARIOS

Two usage scenarios are created to highlight the benefits of AnnotateGPT for teachers and demonstrate how it can be integrated into existing annotation practices.

6.2.1 *Lightweight Annotation for High-Volume Feedback*

Justina teaches English at a public high school, where her two classes total nearly 50 students. The current assignment, a persuasive essay on environmental responsibility, has generated a high volume of student writing that requires review. Given her limited time between classes and meetings, she needs to provide meaningful feedback without compromising consistency or clarity.

She turns to AnnotateGPT to streamline her evaluation process. Justina prefers to work quickly: she highlights phrases and leaves brief, often fragmentary annotations, such as “vague,” “support?,” or “good rhetoric.” Rather than requiring her to slow down and write full comments, AnnotateGPT treats these annotations as intent-rich cues and generates expanded feedback suggestions on the fly.

Justina reviews each suggestion in line. She accepts many directly, revises a few to match the tone she prefers for students by commenting, and rejects others if they’re redundant. Since AnnotateGPT operates within her natural annotation process, she doesn’t need to switch modes or perform additional steps. Her shorthand becomes a trigger for feedback expansion, enabling a smooth flow between recognition and response.

Over the weekend, she can give detailed feedback on each submission much faster than at other times. Most importantly, the feedback is consistent and structured even in high volume. For Justina, AnnotateGPT is not a substitute, but a cognitive amplifier that enables her to give feedback effectively without compromising her pedagogy.

6.2.2 *Leveraging Annotation History Across Subjects*

Ash is a teaching assistant responsible for evaluating writing assignments across two distinct undergraduate courses: an introductory academic writing class and a course on the ethics of computer science. While both courses involve writing, the standards and priorities differ. In academic writing, he emphasizes structural clarity, effective transitions, and logical organization. In the ethics course, the focus shifts to argumentative rigour, ethical frameworks, and engagement with real-world case studies.

Ash uses AnnotateGPT to support giving feedback in both contexts. As he works through multiple assignments each week, AnnotateGPT silently builds an annotation history, observing patterns in his annotations, identifying the types of annotations used for each context, and linking those patterns to the course context and content.

Now, evaluating a paper for the ethics of computer science course, Ash highlights a paragraph with a vague appeal to fairness in AI. He writes “needs clarity.” Rather than suggesting feedback on vague writing mechanics, as might be provided in a writing course, AnnotateGPT recognizes the course and essay topic and draws on prior annotations related to ethics. AnnotateGPT correctly inferred that “clarity” in this context referred not to sentence-level writing but to conceptual framing, a distinction it has learned from its prior annotations in this course. In contrast, the same input in the writing class would have triggered feedback about transitions or sentence rephrasing.

For the ethics course, it uses exploratory language that encourages ethical engagement, while in the writing course, it favours structural advice. This style, maintained over several weeks, alleviates Ash’s editing responsibilities, ensures consistent feedback, and allows him to focus more on students’ ideas rather than punctuation and sentence structure.

By retaining past annotations, AnnotateGPT enables Ash to teach ethics while preserving the subtleties of his teaching style. The tool adapts to the context of each class, enhancing the quality of feedback, aligning with learning objectives, and providing timely feedback.

6.3 ENVISIONED APPLICATIONS

While annotations are traditionally used for making sense of text, they also represent an underexplored yet powerful modality for interaction. Their nature of intention, context, and simplicity makes them ideal for guiding AI-enabled systems where explicit input is costly or disruptive by re-imagining annotations not merely as metadata but as interactive signals that highlight the expressive potential of annotations as a primary interface paradigm.

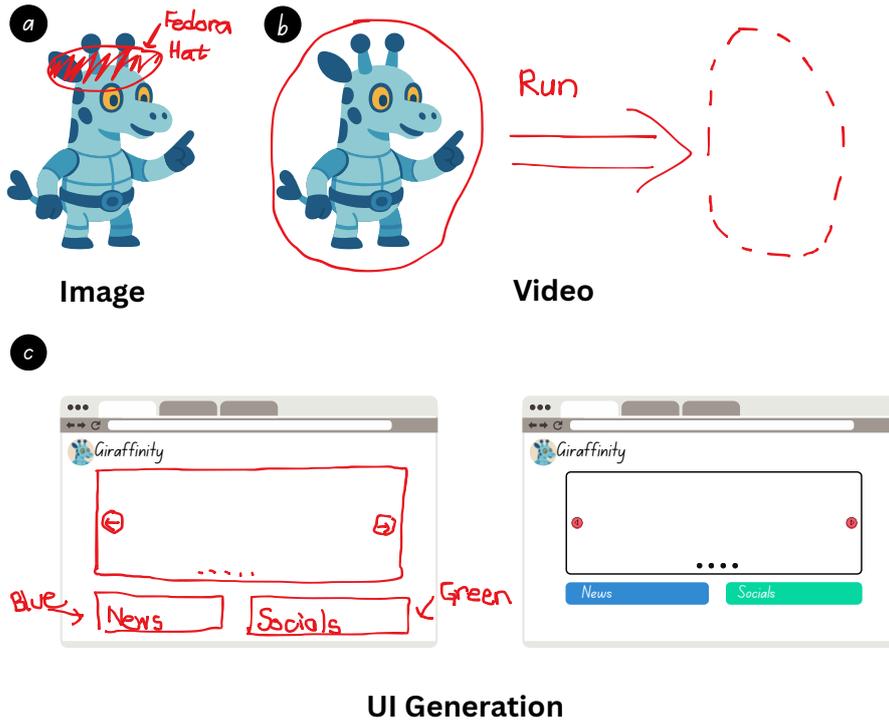


Figure 30: Sample use cases of using annotations to edit and refine generative content, where (a) it can be used to refine image generation, (b) make action flows for video generation, and (c) create and edit user interfaces.

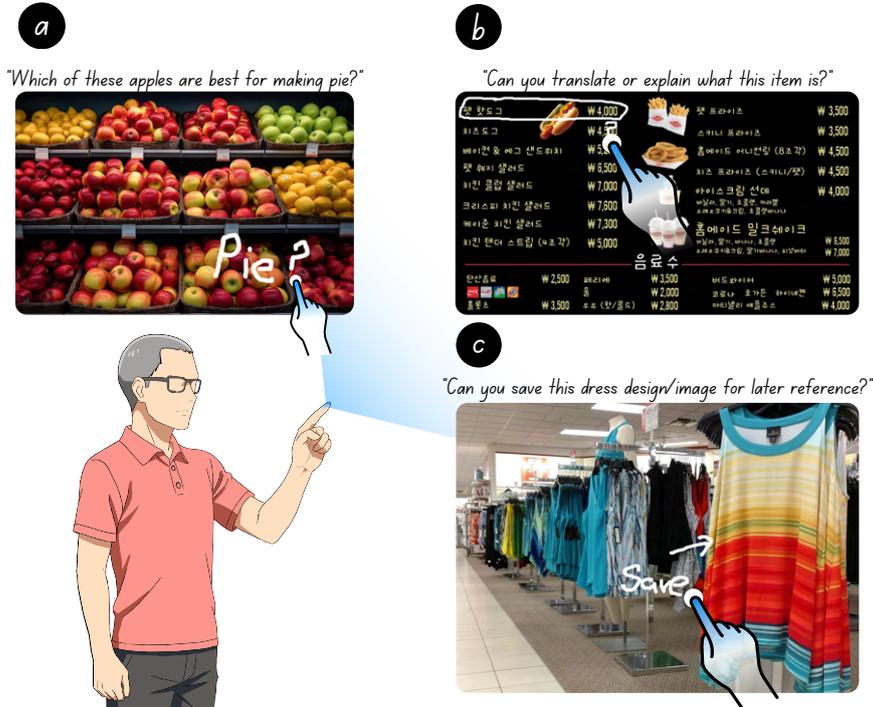


Figure 31: Sample use cases of using annotations to interact with smart glasses. Mid-air hand gestures can be used to create annotations and form complex queries. For example, (a) which apples are best for apple pies, (b) translation and (c) saving an item for later reference. All sample queries are sourced from ChatGPT and provided with the corresponding image.

6.3.1 *Editing Generative Content*

Generative AI tools have been increasingly used in design workflows. While text prompts serve as the primary input modality, users often struggle to create effective prompts to refine generative content [83]. Annotations can be used to define spatial constraints and reference elements within the content [83]. For example, annotations can help refine images, specify action flows for video generation, or create and refine user interfaces. Ultimately, using text prompts to iterate and refine generative content can be costly, making annotations a valuable alternative (see Figure 30).

6.3.2 *Modality for Augmented Reality*

As smart glasses, such as Meta's Ray-Ban and Orion, continue to evolve, they will likely incorporate enhanced augmented reality (AR) capabilities, as well as possibly hand tracking and eye tracking. One of the existing modalities for interacting with smart glasses is to use a voice assistant to make complex queries. However, using voice in public is not ideal as it can be intrusive. Annotations can be used instead to convey their intent. Users can use mid-air hand gestures to create annotations, and the LLM will transform them into a complex query. For example, the user can leave a brief annotation, "Pie?", near a variety of apples, and the LLM can predict "Which of these apples are best for making pie?" (see Figure 31). These queries are actually from ChatGPT, provided with the corresponding image, demonstrating that a single annotation can result in multiple actions (e.g., translation and audio) with the smart glasses.

6.4 LIMITATIONS & FUTURE WORK

This work demonstrates that LLMs can enhance educational annotation workflows. However, limitations of the study findings affect the generalizability. First, the user study involved only six participants, which restricted statistical power and diversity in pedagogical goals. While suitable for exploratory HCI research, future studies should recruit a broader participant pool across institutions and diverse teaching experiences to validate the results.

Second, AnnotateGPT was evaluated in a single-session setting, which overlooks the effects of long-term use, including trust and habit formation. There are concerns that AI tools can lead to overreliance, so longitudinal studies are necessary to understand how teachers adapt to LLM-assisted workflows and whether improvements in efficiency or feedback quality persist over time.

Notably, the evaluation focused on instructor experience rather than student outcomes. Participants reported a reduced physical workload and appreciated the fluency of the generated feedback. However, this work did not assess its impact on student learning or trust in instructional communication. For example, participants found that

AnnotateGPT did not find all issues. However, overcommenting can undermine the students' motivation and confidence [95]. Understanding the effects of AI-augmented annotation on students is a critical next step.

Finally, future work must address the ethical implications of using LLMs in education. AI assistance can overshadow the teacher's identity. Transparent design choices, such as disclosing AI involvement and preserving the educator's voice, are crucial to enhancing pedagogical goals. Future work could also explore implementing dynamic scopes to support more workflows or encourage additional user annotations. For instance, instead of searching the entire document, it could be restricted to one question at a time or to one question across multiple documents or students, forcing more user annotations. Future work could also determine whether a user has finished annotating the document or if the annotation is explicit before activating the assistant, which may enhance ownership and preserve the educator's identity in the annotations. [53]

6.5 SUMMARY

In conclusion, the study findings illustrate how LLMs serve as cognitive amplifiers in feedback workflows, transforming educators into curators of feedback. This shift not only reduces physical workload but also facilitates more dynamic interaction. In the future, annotations can serve as a flexible interaction modality that promotes personalization, content creation, and adaptation in various settings. These trends suggest a future where annotations serve not only to make sense of text but also to establish an interactive paradigm between human intentions and AI.

7

CONCLUSION

This work explores how large language models can augment human annotation practices, transforming feedback from a laborious task into a collaborative interaction. This leads to two research questions: (RQ1) whether LLMs can infer the purposes behind annotations and (RQ2) whether they can be used for comprehensive and continuous document annotation. Through a user study of AnnotateGPT, the findings show that annotations are often ambiguous or shorthand, yet still serve as surprisingly rich signals for LLMs, enabling efficient and expressive interaction without compromising control (RQ1). In addition, it was able to articulate in-depth feedback throughout the document, reduce physical workload, and preserve educator control (RQ2). However, these benefits also challenge traditional notions of authorship and agency. As annotations shift from making sense of text to an interaction paradigm, they blur the boundary between user intention and system behaviour. Much like the saying “a picture is worth a thousand words”, with the current technology, even a *scribble* is worth a thousand words.

APPENDIX

A.1 ANNOTATION HISTORY

ID	Last Annotation History Entry
P1	The user has a pattern of marking significant terms and grammatical elements in their annotations, often using red to denote importance, as seen in previous annotations that emphasized key beliefs and grammatical elements like subject pronouns.
P2	The user has previously provided annotations aimed at improving clarity in student writing, recognizing strong arguments, and informing students about structuring their thoughts and engaging with their educational perspectives.
P3	User has a history of using underlining for emphasizing important phrases, red pen strokes for indicating questions, and comments that encourage clarity and conciseness in student writing.
P4	Past annotations have included corrections and highlights for clarity and precision, emphasizing the importance of strong argumentative language and precise wording in student writings.
P5	Previous annotations in the history involve circling significant phrases and marking corrections, emphasizing the importance of clarity and coherent sentence structures. The user often highlights crucial ideas or transitions, indicating a focus on constructive critique and feedback for student improvement.
P6	The user's annotations support both revision by highlighting key educational arguments and targeted feedback by clarifying student performance and future planning.

Table 7: The annotation history shown at the last activation when inferring the annotation purpose for each participant.

A.2 ADDITIONAL STATISTICAL TEST RESULTS

Term	Df	Df_{res}	F	p	η_p^2
TECHNIQUE	1.0	75.0	17.758	0.000	0.191
QUESTIONS	7.0	75.0	2.671	0.016	0.200
TECHNIQUE \times QUESTIONS	7.0	75.0	0.860	0.542	0.074

Table 8: ART ANOVA results for strokes per annotation

Term	Df	Df_{res}	F	p	η_p^2
TECHNIQUE	1.0	75.0	42.704	0.000	0.363
QUESTIONS	7.0	75.0	7.589	0.000	0.415
TECHNIQUE \times QUESTIONS	7.0	75.0	2.882	0.010	0.212

Table 9: ART ANOVA results for number of annotations

Term	Df	Df_{res}	F	p	η_p^2
TECHNIQUE	1.0	75.0	15.364	0.000	0.170
QUESTIONS	7.0	75.0	7.783	0.000	0.421
TECHNIQUE \times QUESTIONS	7.0	75.0	2.736	0.014	0.203

Table 10: ART ANOVA Results for annotating duration

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Q1	–	0.010	0.006	0.000	0.000	0.000	0.000	0.000
Q2		–	0.895	0.226	0.048	0.048	0.160	0.160
Q3			–	0.304	0.072	0.072	0.211	0.211
Q4				–	0.498	0.498	0.873	0.873
Q5					–	1.000	0.632	0.632
Q6						–	0.632	0.632
Q7							–	1.000
Q8								–

Table 11: Pairwise Benjamini–Hochberg adjusted p-values between questions for ANNOTATEGPT regarding the number of annotations

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Q1	–	0.341	0.400	0.020	0.318	0.001	0.000	0.003
Q2		–	0.927	0.200	0.986	0.014	0.005	0.048
Q3			–	0.160	0.895	0.010	0.003	0.035
Q4				–	0.211	0.263	0.136	0.545
Q5					–	0.015	0.005	0.051
Q6						–	0.755	0.657
Q7							–	0.411
Q8								–

Table 12: Pairwise Benjamini–Hochberg adjusted p-values between questions for BASELINE regarding the number of annotations

	BASELINE							
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Q1	0.509	0.820	0.895	0.115	0.787	0.006	0.002	0.022
Q2	0.001	0.020	0.015	0.346	0.022	0.895	0.632	0.787
Q3	0.001	0.014	0.010	0.267	0.015	1.000	0.748	0.664
Q4	0.000	0.000	0.000	0.025	0.000	0.309	0.527	0.128
Q5	0.000	0.000	0.000	0.003	0.000	0.072	0.160	0.021
Q6	0.000	0.000	0.000	0.003	0.000	0.072	0.160	0.021
Q7	0.000	0.000	0.000	0.015	0.000	0.212	0.394	0.077
Q8	0.000	0.000	0.000	0.015	0.000	0.212	0.394	0.077

Table 13: Benjamini–Hochberg adjusted p-values for ANNOTATEGPT vs. BASELINE across questions regarding the number of annotations

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Q1	-	0.000						
Q2		-	0.851	0.244	0.051	0.051	0.121	0.315
Q3			-	0.354	0.084	0.084	0.194	0.434
Q4				-	0.452	0.452	0.736	0.912
Q5					-	1.000	0.723	0.370
Q6						-	0.723	0.370
Q7							-	0.630

Table 14: Pairwise Benjamini–Hochberg adjusted p -values between questions for ANNOTATEGPT regarding annotating duration

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Q1	-	0.194	0.522	0.010	0.058	0.000	0.000	0.000
Q2		-	0.529	0.233	0.598	0.025	0.011	0.009
Q3			-	0.062	0.228	0.003	0.001	0.001
Q4				-	0.535	0.321	0.198	0.175
Q5					-	0.096	0.051	0.043
Q6						-	0.790	0.736
Q7							-	0.957

Table 15: Pairwise Benjamini–Hochberg adjusted p -values between questions for BASELINE regarding annotating duration

	BASELINE							
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Q1	0.951	0.223	0.578	0.013	0.071	0.000	0.000	0.000
Q2	0.000	0.023	0.003	0.301	0.089	0.968	0.817	0.770
Q3	0.000	0.012	0.001	0.212	0.054	0.817	0.968	0.932
Q4	0.000	0.000	0.000	0.025	0.003	0.228	0.370	0.417
Q5	0.000	0.000	0.000	0.002	0.000	0.047	0.090	0.104
Q6	0.000	0.000	0.000	0.002	0.000	0.047	0.090	0.104
Q7	0.000	0.000	0.000	0.008	0.001	0.110	0.208	0.228
Q8	0.000	0.001	0.000	0.037	0.005	0.296	0.452	0.504

Table 16: Benjamini–Hochberg adjusted p -values for ANNOTATEGPT vs. BASELINE across questions regarding annotating duration

Subscale	<i>Mdn</i> (<i>GPT</i> – <i>Base</i>)	<i>IQR</i> (<i>GPT</i> – <i>Base</i>)	<i>Z</i>	<i>p</i>	<i>r</i>
Physical	2.00 – 2.50	0.75 – 2.50	0.00	0.03	0.62
Frustration	2.00 – 2.50	1.50 – 2.50	4.00	0.22	0.36
Temporal	5.00 – 5.50	3.75 – 4.00	4.50	0.69	0.12
Performance	4.00 – 3.00	2.25 – 0.75	4.00	1.00	0.00
Effort	5.50 – 5.50	1.75 – 2.50	10.00	1.00	0.00
Mental	6.00 – 6.00	1.50 – 2.25	9.00	0.84	0.06

Table 17: NASA-TLX scores for ANNOTATEGPT and BASELINE

BIBLIOGRAPHY

- [1] A. Abdul-Rahman, J. Lein, K. Coles, Eamonn Maguire, M. Meyer, M. Wynne, C. Johnson, Anne Trefethen, and Min Chen. "Rule-based visual mappings - with a case study on poetry visualization." In: *Computer Graphics Forum* 32 (June 2013). DOI: [10.1111/cgf.12125](https://doi.org/10.1111/cgf.12125).
- [2] Adobe. *Adobe Acrobat*. 2024. URL: <https://acrobat.adobe.com/> (visited on 05/25/2024).
- [3] Natalie M. Agius and Ann Wilkinson. "Students' and teachers' views of written feedback at undergraduate level: A literature review." In: *Nurse Education Today* 34.4 (2014), pp. 552–559. ISSN: 0260-6917. DOI: <https://doi.org/10.1016/j.nedt.2013.07.005>.
- [4] Saleema Amershi et al. "Guidelines for human-AI interaction." In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–13. ISBN: 9781450359702. DOI: [10.1145/3290605.3300233](https://doi.org/10.1145/3290605.3300233). URL: <https://doi.org/10.1145/3290605.3300233>.
- [5] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. "Foundation models defining a new era in vision: A survey and outlook." In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2025), pp. 1–20. DOI: [10.1109/TPAMI.2024.3506283](https://doi.org/10.1109/TPAMI.2024.3506283).
- [6] Richard Bailey and Mark Garner and. "Is the feedback in higher education assessment worth the paper it is written on? Teachers' reflections on their practices." In: *Teaching in Higher Education* 15.2 (2010), pp. 187–198. DOI: [10.1080/13562511003620019](https://doi.org/10.1080/13562511003620019). eprint: <https://doi.org/10.1080/13562511003620019>. URL: <https://doi.org/10.1080/13562511003620019>.
- [7] Elaine Ball, Helen Franks, Jane Jenkins, Maureen McGrath, and Jackie Leigh. "Annotation is a valuable tool to enhance learning and assessment in student essays." In: *Nurse Education Today* 29.3 (2009). Special Issue: Selected papers from the 2nd Int. Nurse Education Conf. Research and Innovation in Int. Nurse Education 9-11 June, 2008, Dublin, Ireland, pp. 284–291. ISSN: 0260-6917. DOI: <https://doi.org/10.1016/j.nedt.2008.10.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0260691708001469>.
- [8] Aaron Bangor, Philip T. Kortum, and James T. Miller and. "An empirical evaluation of the system usability scale." In: *Int. J. of Human-Computer Interaction* 24.6 (2008), pp. 574–594. DOI: [10.1080/10447310802205776](https://doi.org/10.1080/10447310802205776). eprint: <https://doi.org/10.1080/10447310802205776>.

- 1080/10447310802205776. URL: <https://doi.org/10.1080/10447310802205776>.
- [9] Elton Barker, Anna Foka, and Kyriaki Konstantinidou. "Coding for the many, transforming knowledge for all: Annotating digital documents." In: *PMLA/Publications of the Modern Language Association of America* 135.1 (2020), pp. 195–202. DOI: [10.1632/pmla.2020.135.1.195](https://doi.org/10.1632/pmla.2020.135.1.195).
- [10] Karim Benharrak, Tim Zindulka, Florian Lehmann, Hendrik Heuer, and Daniel Buschek. "Writer-defined AI personas for on-demand feedback generation." In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2024. ISBN: 9798400703300. DOI: [10.1145/3613904.3642406](https://doi.org/10.1145/3613904.3642406). URL: <https://doi.org/10.1145/3613904.3642406>.
- [11] Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. "TOEFL₁₁: A corpus of non-native English." In: *ETS Research Report Series* 2013.2 (2013), pp. i–15. DOI: <https://doi.org/10.1002/j.2333-8504.2013.tb02331.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2333-8504.2013.tb02331.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2333-8504.2013.tb02331.x>.
- [12] Paolo Bottoni, Roberta Civica, Stefano Levialdi, Laura Orso, Emanuele Panizzi, and Rosa Trinchese. "MADCOW: A multimedia digital annotation system." In: *Proc. of the Working Conf. on Advanced Visual Interfaces*. ACM, 2004, pp. 55–62. ISBN: 1581138679. DOI: [10.1145/989863.989870](https://doi.org/10.1145/989863.989870). URL: <https://doi.org/10.1145/989863.989870>.
- [13] Adam James Bradley, Victor Sawal, Sheelagh Carpendale, and Christopher Collins. "Textension: Digitally augmenting document spaces in analog texts." In: *DHQ: Digital Humanities Quarterly* 13.3 (2019).
- [14] Lyn Brodie and Birgit Loch. "Annotations with a tablet PC or typed feedback: Does it make a difference?" In: *Proc. Australasian Association for Engineering Education Conf.* (Jan. 2009).
- [15] John Brooke. "SUS: A quick and dirty usability scale." In: *Usability Eval. Ind.* 189 (Nov. 1995).
- [16] Tom Brown et al. "Language models are few-shot learners." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- [17] Tom B. Brown et al. "Language models are few-shot learners." In: *Proc. Int. Conf. on Neural Information Processing Systems*. Curran Associates Inc., 2020. ISBN: 9781713829546.

- [18] Alice Jane Bernheim Brush and Alan Borning. "Annotating digital documents for asynchronous collaboration." PhD thesis. University of Washington, 2002. ISBN: 0493919007.
- [19] Guillaume Cabanac, Max Chevalier, Claude Chrisment, and Christine Julien. "A social validation of collaborative annotations on digital documents." In: *Int. Workshop on Annotation for Collaboration - Methods, Tools and Practices*. 2005, pp. 31–40.
- [20] Nan Cao and Weiwei Cui. "Overview of text visualization techniques." In: *Introduction to Text Visualization*. Atlantis Press, 2016, pp. 11–40. ISBN: 978-94-6239-186-4. DOI: [10.2991/978-94-6239-186-4_2](https://doi.org/10.2991/978-94-6239-186-4_2). URL: https://doi.org/10.2991/978-94-6239-186-4_2.
- [21] Yupeng Chang et al. "A survey on evaluation of large language models." In: *ACM Trans. Intell. Syst. Technol.* 15.3 (Mar. 2024). ISSN: 2157-6904. DOI: [10.1145/3641289](https://doi.org/10.1145/3641289). URL: <https://doi.org/10.1145/3641289>.
- [22] Yu-Chien Chen, Ren-Hung Hwang, and Cheng-Yu Wang. "Development and evaluation of a Web 2.0 annotation system as a learning tool in an e-learning environment." In: *Computers & Education* 58.4 (2012), pp. 1094–1105.
- [23] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. "Benchmarking large language models in retrieval-augmented generation." In: *Proc. AAAI Conf. on Artificial Intelligence and Conf. on Innovative Applications of Artificial Intelligence and Symp. on Educational Advances in Artificial Intelligence*. AAAI Press, 2024. ISBN: 978-1-57735-887-9. DOI: [10.1609/aaai.v38i16.29728](https://doi.org/10.1609/aaai.v38i16.29728). URL: <https://doi.org/10.1609/aaai.v38i16.29728>.
- [24] Xiang 'Anthony' Chen, Chien-Sheng Wu, Lidiya Murakhov'ska, Philippe Laban, Tong Niu, Wenhao Liu, and Caiming Xiong. "Marvista: Exploring the design of a human-AI collaborative news reading tool." In: *ACM Trans. Comput.-Hum. Interact.* 30.6 (Sept. 2023). ISSN: 1073-0516. DOI: [10.1145/3609331](https://doi.org/10.1145/3609331). URL: <https://doi.org/10.1145/3609331>.
- [25] Patrick Chiu and Lynn Wilcox. "A dynamic grouping technique for ink and audio notes." In: *Proc. ACM Symp. on User Interface Software and Technology*. ACM, 1998, pp. 195–202. ISBN: 1581130341. DOI: [10.1145/288392.288605](https://doi.org/10.1145/288392.288605). URL: <https://doi.org/10.1145/288392.288605>.
- [26] Jean-Peic Chou, Alexa Fay Siu, Nedim Lipka, Ryan Rossi, Franck Dérnoncourt, and Maneesh Agrawala. "TaleStream: Supporting story ideation with trope knowledge." In: *Proc. ACM Symp. on User Interface Software and Technology*. ACM, Oct. 2023. DOI: [10.1145/3586183.3606807](https://doi.org/10.1145/3586183.3606807). URL: <http://dx.doi.org/10.1145/3586183.3606807>.

- [27] Andrew D. Cohen and Marilda C. Cavalcanti. "Feedback on compositions: Teacher and student verbal reports." In: *Second Language Writing (Cambridge Applied Linguistics): Research Insights for the Classroom*. Ed. by Barbara Editor Kroll. Cambridge Applied Linguistics. Cambridge University Press, 1990, pp. 155–177.
- [28] Christopher Collins, Sheelagh Carpendale, and Gerald Penn. "DocuBurst: Visualizing document content using language structure." In: *Computer Graphics Forum (Proc. of the Eurographics / IEEE-VGTC Symp. on Visualization (EuroVis))* 28.3 (2009), pp. 1039–1046. DOI: [10.1111/j.1467-8659.2009.01439.x](https://doi.org/10.1111/j.1467-8659.2009.01439.x).
- [29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of deep bidirectional Transformers for language understanding." In: *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423). URL: <https://aclanthology.org/N19-1423/>.
- [30] Tom Driessen, Dimitra Dodou, Pavlo Bazilinskyy, and Joost De Winter. "Putting ChatGPT vision (GPT-4V) to the test: Risk perception in traffic images." In: *Royal Society Open Science* 11.5 (2024), p. 231676.
- [31] Min Fan, Xinyue Cui, Jing Hao, Renxuan Ye, Wanqing Ma, Xin Tong, and Meng Li. "StoryPrompt: Exploring the design space of an AI-empowered creative storytelling system for elementary children." In: *Extended Abstracts of the CHI Conf. on Human Factors in Computing Systems*. ACM, 2024. DOI: [10.1145/3613905.3651118](https://doi.org/10.1145/3613905.3651118). URL: <https://doi.org/10.1145/3613905.3651118>.
- [32] David K. Farkas and Steven E. Poltrock. "Online editing, markup models, and the workplace lives of editors and writers." In: *Writing and Speaking in the Technology Professions*. John Wiley & Sons, Ltd, 2015, pp. 279–286. ISBN: 9781119134633. DOI: <https://doi.org/10.1002/9781119134633.ch45>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119134633.ch45>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119134633.ch45>.
- [33] George Fitzmaurice, Azam Khan, Robert Pieké, Bill Buxton, and Gordon Kurtenbach. "Tracking menus." In: *Proc. ACM Symp. on User Interface Software and Technology*. ACM, 2003, pp. 71–79. ISBN: 1581136366. DOI: [10.1145/964696.964704](https://doi.org/10.1145/964696.964704). URL: <https://doi.org/10.1145/964696.964704>.
- [34] Manuel J Fonseca, César Pimentel, and Joaquim A Jorge. "CALI: An online scribble recognizer for calligraphic interfaces." In: *AAAI Spring Symp. on Sketch Understanding*. AAAI Press Menlo Park. 2002, pp. 51–58.

- [35] Yue Fu, Sami Foell, Xuhai Xu, and Alexis Hiniker. “From text to self: Users’ perception of AIMC tools on interpersonal communication and self.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, May 2024. DOI: [10.1145/3613904.3641955](https://doi.org/10.1145/3613904.3641955). URL: <http://dx.doi.org/10.1145/3613904.3641955>.
- [36] Joaquín Gayoso-Cabada, Antonio Sarasa-Cabezuelo, and José-Luis Sierra-Rodríguez. “A review of annotation classification tools in the educational domain.” In: *Open Computer Science* 9.1 (2019), pp. 299–307. DOI: [doi:10.1515/comp-2019-0021](https://doi.org/10.1515/comp-2019-0021). URL: <https://doi.org/10.1515/comp-2019-0021>.
- [37] Akshay Goel et al. “LLMs accelerate annotation for medical information extraction.” In: *Proc. Machine Learning for Health Symp.* Ed. by Stefan Hegselmann, Antonio Parziale, Divya Shanmugam, Shengpu Tang, Mercy Nyamewaa Asiedu, Serina Chang, Tom Hartvigsen, and Harvineet Singh. Vol. 225. PMLR, Oct. 2023, pp. 82–100. URL: <https://proceedings.mlr.press/v225/goel23a.html>.
- [38] Gene Golovchinsky, Morgan N. Price, and Bill N. Schilit. “From reading to retrieval: Freeform ink annotations as queries.” In: *Proc. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*. ACM, 1999, pp. 19–25. ISBN: 1581130961. DOI: [10.1145/312624.312637](https://doi.org/10.1145/312624.312637). URL: <https://doi.org/10.1145/312624.312637>.
- [39] Feng Han, Yifei Cheng, Megan Strachan, and Xiaojuan Ma. “Hybrid paper-digital interfaces: A systematic literature review.” In: *Proc. ACM Designing Interactive Systems Conf.* ACM, 2021, pp. 1087–1100. ISBN: 9781450384766. DOI: [10.1145/3461778.3462059](https://doi.org/10.1145/3461778.3462059). URL: <https://doi.org/10.1145/3461778.3462059>.
- [40] Mark S. Hancock and Kellogg S. Booth. “Improving menu placement strategies for pen input.” In: *Proc. Graphics Interface*. Canadian Human-Computer Communications Society. Canadian Human-Computer Communications Society, 2004, pp. 221–230. ISBN: 1-56881-227-2. URL: <http://portal.acm.org/citation.cfm?id=1006058.1006085>.
- [41] S. G. Hart. “Nasa-task load index (NASA-TLX); 20 years later.” In: *Proc. of the Human Factors and Ergonomics Society Annual Meeting* 50 (2006), pp. 904–908. URL: <https://api.semanticscholar.org/CorpusID:6292200>.
- [42] Sandra G. Hart and Lowell E. Staveland. “Development of NASA-TLX (task load index): Results of empirical and theoretical research.” In: *Human Mental Workload*. Ed. by Peter A. Hancock and Najmedin Meshkati. Vol. 52. Advances in Psychology. North-Holland, 1988, pp. 139–183. DOI: [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9). URL: <https://www.sciencedirect.com/science/article/pii/S0166411508623869>.

- [43] Annika Hinze, Ralf Heese, Alexa Schlegel, and Markus Luczak-Rösch. “User-defined semantic enrichment of full-text documents: experiences and lessons learned.” In: *Theory and Practice of Digital Libraries*. Ed. by Panayiotis Zaphiris, George Buchanan, Edie Rasmussen, and Fernando Loizides. Springer Berlin Heidelberg, 2012, pp. 209–214. ISBN: 978-3-642-33290-6.
- [44] Md Naimul Hoque, Tasfia Mashiat, Bhavya Ghai, Cecilia D. Shelton, Fanny Chevalier, Kari Kraus, and Niklas Elmqvist. “The HaLLMark effect: Supporting provenance and transparent use of large language models in writing with interactive visualization.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2024. ISBN: 9798400703300. DOI: [10.1145/3613904.3641895](https://doi.org/10.1145/3613904.3641895). URL: <https://doi.org/10.1145/3613904.3641895>.
- [45] Lei Huang et al. “A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.” In: *ACM Trans. Inf. Syst.* 43.2 (Jan. 2025). ISSN: 1046-8188. DOI: [10.1145/3703155](https://doi.org/10.1145/3703155). URL: <https://doi.org/10.1145/3703155>.
- [46] Jiun-Chi Jan, Chih-Ming Chen, and Po-Han Huang. “Enhancement of digital reading performance by using a novel web-based collaborative reading annotation system with two quality annotation filtering mechanisms.” In: *Int. J. of Human-Computer Studies* 86 (2016), pp. 81–93.
- [47] Peiling Jiang, Jude Rayan, Steven P. Dow, and Haijun Xia. “Graphologue: Exploring large language model responses with interactive diagrams.” In: *Proc. of the ACM Symp. on User Interface Software and Technology*. ACM, Oct. 2023. DOI: [10.1145/3586183.3606737](http://dx.doi.org/10.1145/3586183.3606737). URL: <http://dx.doi.org/10.1145/3586183.3606737>.
- [48] Matthew Kam, Jingtao Wang, Alastair Iles, Eric Tse, Jane Chiu, Daniel Glaser, Orna Tarshish, and John Canny. “Livenotes: A system for cooperative and augmented note-taking in lectures.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2005, pp. 531–540. ISBN: 1581139985. DOI: [10.1145/1054972.1055046](https://doi.org/10.1145/1054972.1055046). URL: <https://doi.org/10.1145/1054972.1055046>.
- [49] Levent Burak Kara, Leslie Gennari, and Thomas F. Stahovich. “A sketch-based tool for analyzing vibratory mechanical systems.” In: *J. of Mechanical Design* 130.10 (Sept. 2008), p. 101101. ISSN: 1050-0472. DOI: [10.1115/1.2965595](https://doi.org/10.1115/1.2965595). eprint: https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/130/10/101101/5806361/101101_1.pdf. URL: <https://doi.org/10.1115/1.2965595>.
- [50] Hannah Kim, Kushan Mitra, Rafael Li Chen, Sajjadur Rahman, and Dan Zhang. “MEGAnno+: A human-LLM collaborative annotation system.” In: *Proc. Conf. of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Ed. by Nikolaos Aletras and Orphee De Clercq. Association

- for Computational Linguistics, Mar. 2024, pp. 168–176. URL: <https://aclanthology.org/2024.eacl-demo.18/>.
- [51] Konstantin Klamka and Raimund Dachsel. “IllumiPaper: Illuminated interactive paper.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2017, pp. 5605–5618. ISBN: 9781450346559. DOI: [10.1145/3025453.3025525](https://doi.org/10.1145/3025453.3025525). URL: <https://doi.org/10.1145/3025453.3025525>.
- [52] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. “Large language models are zero-shot reasoners.” In: *Proc. Int. Conf. on Neural Information Processing Systems*. Curran Associates Inc., 2022. ISBN: 9781713871088.
- [53] Nataliya Kosmyrna, Eugene Hauptmann, Ye Tong Yuan, Jessica Situ, Xian-Hao Liao, Ashly Vivian Beresnitzky, Iris Braunstein, and Pattie Maes. *Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task*. 2025. arXiv: [2506.08872 \[cs.AI\]](https://arxiv.org/abs/2506.08872). URL: <https://arxiv.org/abs/2506.08872>.
- [54] Vivian Lai, Chacha Chen, Alison Smith-Renner, Q. Vera Liao, and Chenhao Tan. “Towards a science of human-AI decision making: An overview of design space in empirical human-subject studies.” In: *Proc. ACM Conf. on Fairness, Accountability, and Transparency*. ACM, 2023, pp. 1369–1385. ISBN: 9798400701924. DOI: [10.1145/3593013.3594087](https://doi.org/10.1145/3593013.3594087). URL: <https://doi.org/10.1145/3593013.3594087>.
- [55] J.A. Landay and B.A. Myers. “Sketching interfaces: Toward more human interface design.” In: *Computer* 34.3 (2001), pp. 56–64. DOI: [10.1109/2.910894](https://doi.org/10.1109/2.910894).
- [56] David G Lebow and Dale W Lick. “HyLighter: An effective interactive annotation innovation for distance education.” In: *Conf. on Distance Teaching and Learning*. 2005, pp. 1–5.
- [57] Joanne Leong, Pat Pataranutaporn, Valdemar Danry, Florian Perteneder, Yaoli Mao, and Pattie Maes. “Putting things into context: Generative AI-enabled context personalization for vocabulary learning improves learning motivation.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2024. DOI: [10.1145/3613904.3642393](https://doi.org/10.1145/3613904.3642393). URL: <https://doi.org/10.1145/3613904.3642393>.
- [58] Vladimir I. Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals.” In: *Soviet Physics. Doklady* 10 (1965), pp. 707–710. URL: <https://api.semanticscholar.org/CorpusID:60827152>.
- [59] Patrick Lewis et al. “Retrieval-augmented generation for knowledge-intensive NLP tasks.” In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. URL: <https://arxiv.org/abs/2007.13698>.

- proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.
- [60] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. *A survey on retrieval-augmented text generation*. 2022. arXiv: [2202.01110](https://arxiv.org/abs/2202.01110) [cs.CL].
- [61] Jiawei Li et al. “Fundamental capabilities of large language models and their applications in domain scenarios: A survey.” In: *Proc. Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Association for Computational Linguistics, Aug. 2024, pp. 11116–11141. DOI: [10.18653/v1/2024.acl-long.599](https://doi.org/10.18653/v1/2024.acl-long.599). URL: <https://aclanthology.org/2024.acl-long.599/>.
- [62] Junkai Li, Siyu Wang, Meng Zhang, Weitao Li, Yunghwei Lai, Xinhui Kang, Weizhi Ma, and Yang Liu. *Agent hospital: A simulacrum of hospital with evolvable medical agents*. 2024. arXiv: [2405.02957](https://arxiv.org/abs/2405.02957) [cs.AI].
- [63] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. *Large language models for generative recommendation: A survey and visionary discussions*. 2024. arXiv: [2309.01157](https://arxiv.org/abs/2309.01157) [cs.IR].
- [64] Yinheng Li. “A practical survey on zero-shot prompt design for in-context learning.” In: *Proc. of the Conf. Recent Advances in Natural Language Processing - Large Language Models for Natural Language Processings*. INCOMA Ltd., Shoumen, BULGARIA, 2023. DOI: [10.26615/978-954-452-092-2_069](https://doi.org/10.26615/978-954-452-092-2_069). URL: http://dx.doi.org/10.26615/978-954-452-092-2_069.
- [65] Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. “Guiding large language models via directional stimulus prompting.” In: *Conf. on Neural Information Processing Systems*. 2023. URL: <https://openreview.net/forum?id=UvIN8oQ4uI>.
- [66] Chunyuan Liao, François Guimbretière, Richard Anderson, Natalie Linnell, Craig Prince, and Valentin Razmov. “Papercp: Exploring the integration of physical and digital affordances for active learning.” In: *Proc. IFIP TC 13 Int. Conf. on Human-Computer Interaction - Volume Part II*. Springer. 2007, pp. 15–28.
- [67] Q. Vera Liao, Daniel Gruen, and Sarah Miller. “Questioning the AI: Informing design practices for explainable AI user experiences.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–15. ISBN: 9781450367080. DOI: [10.1145/3313831.3376590](https://doi.org/10.1145/3313831.3376590). URL: <https://doi.org/10.1145/3313831.3376590>.

- [68] Catherine C. Marshall. "Annotation: From paper books to the digital library." In: *Proc. ACM Int. Conf. on Digital Libraries*. ACM, 1997, pp. 131–140. ISBN: 0897918681. DOI: [10.1145/263690.263806](https://doi.org/10.1145/263690.263806). URL: <https://doi.org/10.1145/263690.263806>.
- [69] Catherine C. Marshall. "Toward an ecology of hypertext annotation." In: *Proc. ACM Conf. on Hypertext and Hypermedia: Links, Objects, Time and Space—Structure in Hypermedia Systems: Links, Objects, Time and Space—Structure in Hypermedia Systems*. ACM, 1998, pp. 40–49. ISBN: 0897919726. DOI: [10.1145/276627.276632](https://doi.org/10.1145/276627.276632). URL: <https://doi.org/10.1145/276627.276632>.
- [70] Hrim Mehta, Adam Bradley, Mark Hancock, and Christopher Collins. "Metatation: Annotation as implicit interaction to bridge close and distant reading." In: *ACM Trans. Comput.-Hum. Interact.* 24.5 (Nov. 2017). ISSN: 1073-0516. DOI: [10.1145/3131609](https://doi.org/10.1145/3131609). URL: <https://doi.org/10.1145/3131609>.
- [71] Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. "Reframing instructional prompts to GPTk's language." In: *Findings of the Association for Computational Linguistics*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, May 2022, pp. 589–612. DOI: [10.18653/v1/2022.findings-acl.50](https://aclanthology.org/2022.findings-acl.50). URL: <https://aclanthology.org/2022.findings-acl.50/>.
- [72] Meredith Ringel Morris, A.J. Bernheim Brush, and Brian R. Meyers. "Reading revisited: Evaluating the usability of digital display surfaces for active reading tasks." In: *IEEE Int. Workshop on Horizontal Interactive Human-Computer Systems*. 2007, pp. 79–86. DOI: [10.1109/TABLETOP.2007.12](https://doi.org/10.1109/TABLETOP.2007.12).
- [73] Leonard Muellner. "Annotations and the ancient Greek hero: Past, present, and future." spa. In: *Comunicar* 22.44 (2015), pp. 45–53. ISSN: 1134-3478. DOI: [10.3916/C44-2015-05](https://doi.org/10.3916/C44-2015-05).
- [74] Shankar Narayanaswamy. *Pen and speech recognition in the user interface for mobile multimedia terminals*. University of California, Berkeley, 1996.
- [75] David M. Nichols, Duncan Pemberton, Salah Dalhoumi, Omar Larouk, Claire Belisle, and Michael B. Twidale. "DEBORA: Developing an interface to support collaboration in a digital library." In: *Research and Advanced Technology for Digital Libraries*. Ed. by José Borbinha and Thomas Baker. Springer Berlin Heidelberg, 2000, pp. 239–248. ISBN: 978-3-540-45268-3.
- [76] Kun-Peng Ning, Shuo Yang, Yu-Yang Liu, Jia-Yu Yao, Zhen-Hui Liu, Yu Wang, Ming Pang, and Li Yuan. "Peer-review-in-LLMs: Automatic evaluation method for LLMs in open-environment." In: *arXiv preprint arXiv:2402.01830* (2024).

- [77] Kun-Peng Ning, Shuo Yang, Yuyang Liu, Jia-Yu Yao, Zhenhui Liu, Yonghong Tian, Yibing Song, and Li Yuan. “PiCO: Peer review in LLMs based on consistency optimization.” In: *Int. Conf. on Learning Representations*. 2025. URL: <https://openreview.net/forum?id=sfQ6XpApfS>.
- [78] Kenton O’Hara and Abigail Sellen. “A comparison of reading paper and on-line documents.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 1997, pp. 335–342. ISBN: 0897918029. DOI: [10.1145/258549.258787](https://doi.org/10.1145/258549.258787). URL: <https://doi.org/10.1145/258549.258787>.
- [79] OpenAI et al. *GPT-4 technical report*. 2024. arXiv: [2303.08774](https://arxiv.org/abs/2303.08774) [cs.CL].
- [80] OpenAI. *OpenAI Node API Library*. 2025. URL: <https://platform.openai.com>.
- [81] Ilia A. Ovsianikov, Michael A. Arbib, and Thomas H. McNeill. “Annotation technology.” In: *Int. J. Hum.-Comput. Stud.* 50.4 (Apr. 1999), pp. 329–362. ISSN: 1071-5819. DOI: [10.1006/ijhc.1999.0247](https://doi.org/10.1006/ijhc.1999.0247). URL: <https://doi.org/10.1006/ijhc.1999.0247>.
- [82] W. Bradford Paley. “Textarc: Showing word frequency and distribution in text.” In: *Poster presented at IEEE Symp. on Information Visualization*. Vol. 2002. 2002.
- [83] Hyerim Park, Malin Eiband, Andre Luckow, and Michael Sedlmair. “Exploring Visual Prompts: Refining Images with Scribbles and Annotations in Generative AI Image Tools.” In: *Proc. of the Extended Abstracts of the SIGCHI Conf. on Human Factors in Computing Systems*. Association for Computing Machinery, 2025. ISBN: 9798400713958. DOI: [10.1145/3706599.3719802](https://doi.org/10.1145/3706599.3719802). URL: <https://doi.org/10.1145/3706599.3719802>.
- [84] Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. “Generative agents: Interactive simulacra of human behavior.” In: *Proc. ACM Symp. on User Interface Software and Technology*. ACM, 2023. ISBN: 9798400701320. DOI: [10.1145/3586183.3606763](https://doi.org/10.1145/3586183.3606763). URL: <https://doi.org/10.1145/3586183.3606763>.
- [85] Michelle G Paterno. “Responding to student writing.” In: *Kritika Kultura* 1.2 (2002), p. 5.
- [86] Jennifer Pearson, George Buchanan, and Harold Thimbleby. “Improving annotations in digital documents.” In: *Research and Advanced Technology for Digital Libraries*. Ed. by Maristella Agosti, José Borbinha, Sarantos Kapidakis, Christos Papatheodorou, and Giannis Tsakonas. Springer Berlin Heidelberg, 2009, pp. 429–432. ISBN: 978-3-642-04346-8.
- [87] Ken Pfeuffer, Ken Hinckley, Michel Pahud, and Bill Buxton. “Thumb + pen interaction on tablets.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2017, pp. 3254–3266. ISBN: 9781450346559. DOI: [10.1145/3025453.3025567](https://doi.org/10.1145/3025453.3025567). URL: <https://doi.org/10.1145/3025453.3025567>.

- [88] Laria Reynolds and Kyle McDonell. "Prompt programming for large language models: Beyond the few-shot paradigm." In: *Proc. of the Extended Abstracts of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2021. ISBN: 9781450380959. DOI: [10.1145/3411763.3451760](https://doi.org/10.1145/3411763.3451760). URL: <https://doi.org/10.1145/3411763.3451760>.
- [89] Yotam Sechayk, Ariel Shamir, and Takeo Igarashi. "SmartLearn: Visual-temporal accessibility for slide-based e-learning videos." In: *Extended Abstracts of the CHI Conf. on Human Factors in Computing Systems*. ACM, 2024. DOI: [10.1145/3613905.3650883](https://doi.org/10.1145/3613905.3650883). URL: <https://doi.org/10.1145/3613905.3650883>.
- [90] Olivia Seow. "LingoLand: An AI-assisted immersive game for language learning." In: *Adjunct Proc. of the ACM Symp. on User Interface Software and Technology*. ACM, 2023. DOI: [10.1145/3586182.3625117](https://doi.org/10.1145/3586182.3625117). URL: <https://doi.org/10.1145/3586182.3625117>.
- [91] Orit Shaer, Angelora Cooper, Osnat Mokryn, Andrew L Kun, and Hagit Ben Shoshan. "AI-augmented brainwriting: Investigating the use of LLMs in group ideation." In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2024. ISBN: 9798400703300. DOI: [10.1145/3613904.3642414](https://doi.org/10.1145/3613904.3642414). URL: <https://doi.org/10.1145/3613904.3642414>.
- [92] Jingyu Shi, Rahul Jain, Hyungjun Doh, Ryo Suzuki, and Karthik Ramani. *An HCI-Centric Survey and Taxonomy of Human-Generative-AI Interactions*. 2024. arXiv: [2310.07127](https://arxiv.org/abs/2310.07127) [cs.HC]. URL: <https://arxiv.org/abs/2310.07127>.
- [93] Frank Shipman, Morgan Price, Catherine C. Marshall, and Gene Golovchinsky. "Identifying useful passages in documents based on annotation patterns." In: *Research and Advanced Technology for Digital Libraries*. Ed. by Traugott Koch and Ingeborg Torvik Sølvsberg. Springer Berlin Heidelberg, 2003, pp. 101–112. ISBN: 978-3-540-45175-4.
- [94] Simranjit Singh, Georgios Pavlakos, and Dimitrios Stamoulis. *Evaluating zero-shot GPT-4V performance on 3D visual question answering benchmarks*. 2024. arXiv: [2405.18831](https://arxiv.org/abs/2405.18831) [cs.CV].
- [95] Cvetka Sokolov. "Challenges of written response to student writing: Praise, over-Commenting and appropriation." In: *Arbeiten aus Anglistik und Amerikanistik* (2022). DOI: <https://doi.org/10.24053/aaa-2022-0006>.
- [96] Addison YS Su, Stephen JH Yang, Wu-Yuin Hwang, and Jia Zhang. "A Web 2.0-based collaborative annotation system for enhancing knowledge sharing in collaborative learning environments." In: *Computers & Education* 55.2 (2010), pp. 752–766.
- [97] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. "Luminate: Structured generation and exploration of design space with large language models for human-AI co-creation." In: *Proc. of the SIGCHI Conf. on Human Factors in*

- Computing Systems*. ACM, May 2024. DOI: [10.1145/3613904.3642400](https://doi.org/10.1145/3613904.3642400). URL: <http://dx.doi.org/10.1145/3613904.3642400>.
- [98] Craig J. Sutherland, Andrew Luxton-Reilly, and Beryl Plimmer. “Freeform digital ink annotations in electronic documents: A systematic mapping study.” In: *Computers & Graphics* 55 (2016), pp. 1–20. ISSN: 0097-8493. DOI: <https://doi.org/10.1016/j.cag.2015.10.014>. URL: <https://www.sciencedirect.com/science/article/pii/S0097849315001818>.
- [99] Robert Tally. “Graphs, maps, trees: Abstract models for a literary history.” In: *Modern Language Quarterly* 68 (Feb. 2007), pp. 132–135. DOI: [10.1215/00267929-2006-032](https://doi.org/10.1215/00267929-2006-032).
- [100] Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. “Let me speak freely? A study on the impact of format restrictions on performance of large language models.” In: *Proc. Conf. on Empirical Methods in Natural Language Processing: Industry Track*. Ed. by Franck Dernoncourt, Daniel Preoțiuc-Pietro, and Anastasia Shimorina. Association for Computational Linguistics, Nov. 2024, pp. 1218–1236. DOI: [10.18653/v1/2024.emnlp-industry.91](https://doi.org/10.18653/v1/2024.emnlp-industry.91). URL: <https://aclanthology.org/2024.emnlp-industry.91/>.
- [101] Joseph G Tront, Vinod Eligeti, and Jane Prey. “Classroom presentations using tablet PCs and WriteOn.” In: *Proc. Frontiers in Education*. IEEE, 2006, pp. 1–5.
- [102] Stephanie Valencia, Richard Cave, Krystal Kallarackal, Katie Seaver, Michael Terry, and Shaun K. Kane. ““The less I type, the better”: How AI language models can enhance or impede communication for AAC users.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. Association for Computing Machinery, 2023. ISBN: 9781450394215. DOI: [10.1145/3544548.3581560](https://doi.org/10.1145/3544548.3581560). URL: <https://doi.org/10.1145/3544548.3581560>.
- [103] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need.” In: *Proc. Int. Conf. on Neural Information Processing Systems*. Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964.
- [104] Vercel. *Next.js – The React Framework for the Web*. 2025. URL: <https://nextjs.org>.
- [105] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. “GPT-4V(ision) for robotics: Multimodal task planning from human demonstration.” In: *IEEE Robotics and Automation Letters* 9.11 (2024), pp. 10567–10574. DOI: [10.1109/LRA.2024.3477090](https://doi.org/10.1109/LRA.2024.3477090).
- [106] Qian Wan, Xin Feng, Yining Bei, Zhiqi Gao, and Zhicong Lu. “Metamorpheus: Interactive, affective, and creative dream narration through metaphorical visual storytelling.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM,

2024. ISBN: 9798400703300. DOI: [10.1145/3613904.3642410](https://doi.org/10.1145/3613904.3642410). URL: <https://doi.org/10.1145/3613904.3642410>.
- [107] Junke Wang, Lingchen Meng, Zejia Weng, Bo He, Zuxuan Wu, and Yu-Gang Jiang. *To see is to believe: Prompting GPT-4V for better visual instruction tuning*. 2023. arXiv: [2311.07574](https://arxiv.org/abs/2311.07574) [cs.CV].
- [108] Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. "Human-LLM collaborative annotation through effective verification of LLM labels." In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2024. DOI: [10.1145/3613904.3641960](https://doi.org/10.1145/3613904.3641960). URL: <https://doi.org/10.1145/3613904.3641960>.
- [109] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. "Chain-of-thought prompting elicits reasoning in large language models." In: *Proc. Int. Conf. on Neural Information Processing Systems*. Curran Associates Inc., 2022. ISBN: 9781713871088.
- [110] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. "The aligned rank transform for nonparametric factorial analyses using only anova procedures." In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. Association for Computing Machinery, 2011, pp. 143–146. ISBN: 9781450302289. DOI: [10.1145/1978942.1978963](https://doi.org/10.1145/1978942.1978963). URL: <https://doi.org/10.1145/1978942.1978963>.
- [111] Patricia G Wojahn, Christine M Neuwirth, and Barbara Bullock. "Effects of interfaces for annotation on communication in a collaborative task." In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. 1998, pp. 456–463.
- [112] Joanna Wolfe. "Annotation technologies: A software and research review." In: *Computers and Composition* 19.4 (2002), pp. 471–497. ISSN: 8755-4615. DOI: [https://doi.org/10.1016/S8755-4615\(02\)00144-5](https://doi.org/10.1016/S8755-4615(02)00144-5). URL: <https://www.sciencedirect.com/science/article/pii/S8755461502001445>.
- [113] Chaoyi Wu et al. *Can GPT-4V(ision) serve medical applications? Case studies on GPT-4V for multimodal medical diagnosis*. 2023. arXiv: [2310.09909](https://arxiv.org/abs/2310.09909) [cs.CV].
- [114] Minghao Wu, Yulin Yuan, Gholamreza Haffari, and Longyue Wang. *(Perhaps) beyond human translation: Harnessing multi-agent collaboration for translating ultra-long literary texts*. 2024. arXiv: [2405.11804](https://arxiv.org/abs/2405.11804) [cs.CL].
- [115] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. *The dawn of LMMs: Preliminary explorations with GPT-4V(ision)*. 2023. arXiv: [2309.17421](https://arxiv.org/abs/2309.17421) [cs.CV].

- [116] Ryan Yen, Jian Zhao, and Daniel Vogel. “Code Shaping: Iterative code editing with free-form AI-interpreted sketching.” In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: [10.1145/3706598.3713822](https://doi.org/10.1145/3706598.3713822). URL: <https://doi.org/10.1145/3706598.3713822>.
- [117] Gokul Yenduri et al. “GPT (Generative Pre-Trained Transformer)— A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions.” In: *IEEE Access* 12 (2024), pp. 54608–54649. DOI: [10.1109/ACCESS.2024.3389497](https://doi.org/10.1109/ACCESS.2024.3389497).
- [118] Dongwook Yoon, Nicholas Chen, and François Guimbretière. “TextTearing: Opening white space for digital ink annotation.” In: *Proc. ACM Symp. on User Interface Software and Technology*. ACM, 2013, pp. 107–112. ISBN: 9781450322683. DOI: [10.1145/2501988.2502036](https://doi.org/10.1145/2501988.2502036). URL: <https://doi.org/10.1145/2501988.2502036>.
- [119] Dongwook Yoon, Nicholas Chen, François Guimbretière, and Abigail Sellen. “RichReview: Blending ink, speech, and gesture to support collaborative document review.” In: *Proc. ACM Symp. on User Interface Software and Technology*. ACM, 2014, pp. 481–490. ISBN: 9781450330695. DOI: [10.1145/2642918.2647390](https://doi.org/10.1145/2642918.2647390). URL: <https://doi.org/10.1145/2642918.2647390>.
- [120] Danni Yu, Luyang Li, Hang Su, and Matteo Fuoli. *Assessing the potential of AI-assisted pragmatic annotation: The case of apologies*. 2024. arXiv: [2305.08339](https://arxiv.org/abs/2305.08339) [cs.CL].
- [121] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. “Evaluation of retrieval-augmented generation: A survey.” In: *Big Data*. Ed. by Wenwu Zhu, Hui Xiong, Xiuzhen Cheng, Lizhen Cui, Zhicheng Dou, Junyu Dong, Shanchen Pang, Li Wang, Lanju Kong, and Zhenxiang Chen. Springer Nature Singapore, 2025, pp. 102–120.
- [122] Zahra Zahedi and Subbarao Kambhampati. *Human-AI symbiosis: A survey of current approaches*. 2021. arXiv: [2103.09990](https://arxiv.org/abs/2103.09990) [cs.AI]. URL: <https://arxiv.org/abs/2103.09990>.
- [123] Vivian Zamel. “Responding to student writing.” In: *TESOL Quarterly* 19.1 (1985), pp. 79–101. ISSN: 00398322. URL: <http://www.jstor.org/stable/3586773> (visited on 06/01/2025).
- [124] Jingyi Zhang, Jiaying Huang, Sheng Jin, and Shijian Lu. “Vision-language models for vision tasks: A survey.” In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 46.8 (2024), pp. 5625–5644. DOI: [10.1109/TPAMI.2024.3369699](https://doi.org/10.1109/TPAMI.2024.3369699).
- [125] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. *Retrieval-augmented generation for AI-generated content: A survey*. 2024. arXiv: [2402.19473](https://arxiv.org/abs/2402.19473) [cs.CV].

- [126] Wayne Xin Zhao et al. *A survey of large language models*. 2023. arXiv: [2303.18223](https://arxiv.org/abs/2303.18223) [cs.CL].
- [127] Yijun Zhao, Jiangyu Pan, Yan Dong, Tianshu Dong, Guanyun Wang, Fangtian Ying, Qihang Shen, and Jiacheng Cao. “Language urban odyssey: A serious game for enhancing second language acquisition through large language models.” In: *Extended Abstracts of the CHI Conf. on Human Factors in Computing Systems*. ACM, 2024. DOI: [10.1145/3613905.3651112](https://doi.org/10.1145/3613905.3651112). URL: <https://doi.org/10.1145/3613905.3651112>.
- [128] Zihuai Zhao et al. “Recommender systems in the era of large language models (LLMs).” In: *IEEE Trans. on Knowledge and Data Engineering* 36.11 (2024), pp. 6889–6907. DOI: [10.1109/TKDE.2024.3392335](https://doi.org/10.1109/TKDE.2024.3392335).
- [129] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. “Large language models are human-level prompt engineers.” In: *NeurIPS Foundation Models for Decision Making Workshop*. 2022. URL: <https://openreview.net/forum?id=YdqwNaCLCx>.
- [130] Yucheng Zhou, Xiubo Geng, Tao Shen, Chongyang Tao, Guodong Long, Jian-Guang Lou, and Jianbing Shen. *Thread of Thought unraveling chaotic contexts*. 2023. arXiv: [2311.08734](https://arxiv.org/abs/2311.08734) [cs.CL]. URL: <https://arxiv.org/abs/2311.08734>.
- [131] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhao Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. “Large language models for information retrieval: A survey.” In: *CoRR* abs/2308.07107 (2023). arXiv: [2308.07107](https://arxiv.org/abs/2308.07107). URL: <https://arxiv.org/abs/2308.07107>.