Exploring the Innovation Opportunities for Pre-trained Models

MINJUNG PARK, Carnegie Mellon University, USA

JODI FORLIZZI, Carnegie Mellon University, USA

JOHN ZIMMERMAN, Carnegie Mellon University, USA

Innovators innovate by understanding what is working with emerging technologies, using this knowledge to mitigate risks and identify safe starting places. With the emergence of pre-trained models, understanding what works with these models has been essential. To address this, we explored 67 applications that are made with pre-trained models to identify what they do and how they create value. Using an artifact analysis approach, we analyzed applications from CHI '24, focusing on the domains and data types these applications were designed for, as well as their capabilities and robustness. Additionally, we identified interaction design patterns that can guide future innovators in designing applications with pre-trained models. This process allowed us to uncover the opportunity space for innovation with pre-trained models. Our findings provide valuable insights into leveraging pre-trained models for innovative applications.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Additional Key Words and Phrases: LLM, Generative AI, Pre-trained Models, HCI Innovation, Interaction Design Pattern, Artifact Analysis

ACM Reference Format:

Minjung Park, Jodi Forlizzi, and John Zimmerman. 2018. Exploring the Innovation Opportunities for Pre-trained Models. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 33 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Innovators benefit from understanding what is working, from knowing of the situations where products and services will be successful. In many cases, this knowledge helps innovators find a safe, risk-free place to begin product development. By *innovator*, we mean the practitioners who work on new products and services they expect to successfully deploy in the world.

For example, manufacturers of mobile phones noticed the success of small digital cameras that were easy to carry and helped people visually document their lives. They innovated by adding digital cameras to phones, creating the opportunity for people to carry fewer items in their pockets and bags. Later, innovators noticed the emergent behavior of selfie-taking, and they introduced a forward-facing camera on mobile phones.

Similarly, design practitioners create resources that document success, such as online repositories of interaction design patterns, which show conventional ways of overcoming frequent interaction challenges. While researchers most

Authors' addresses: Minjung Park, mpark2@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, PA, USA; Jodi Forlizzi, forlizzi@cs.cmu.edu, Carnegie Mellon University, Pittsburgh, PA, USA; John Zimmerman, johnz@cs.cmu.edu, Carnegie Mellon University, Pittsburgh, PA, USA.

Manuscript submitted to ACM

often want to know the gaps, to identify what is not being done so they can make a novel contribution, innovators notice and discuss what is working to mitigate risk when making new products and services.

Over the last several years, a growing body of HCI research explored AI innovation. This research revealed challenges with integrating data science into the enterprise[30] and challenges HCI/UX practitioners face when trying to envision AI products and services that people want and that can be easily developed [12, 46, 51]. Researchers note a very high failure rate for AI initiatives within companies [43]. They also noted missing low-hanging fruit, situations where simple AI that could create immediate value was not developed [48, 49, 53]. They described an AI innovation gap in which data science teams envision services customers don't want while HCI/UX teams envision services that cannot be built [49, 50]. To address these problems, researchers developed resources documenting AI capabilities found in commercial successful products and services [51], new design processes to help innovators envision better things to build [53], and guidebooks to support prototyping of effective and responsible AI systems [2, 4, 23, 37].

The release of ChatGPT in November 2022 spawned huge interest from innovators and a "gold rush" of investment in creating new AI products and services that make use of pre-trained models. In this paper, we use the term *pre-trained model* to collectively mean Large Language Models, Generative AI, and Foundation Models. We chose this term because the pre-trained aspect of these models offers a major shift for innovators. Instead of collecting data and building a model, innovators can get a faster start by using models that already exist. Pre-trained models offer transfer learning, where a model trained to do one thing can develop many other capabilities. In human terms, transfer learning is when learning one skill provides learning that works for other things. For example, when people learn how to hit a golf ball, they have also developed some of the knowledge and skills needed to hit a baseball (the skill of hitting a thing with a stick).

In the case of pre-trained models, systems developed to translate from one language to another unintentionally developed additional capabilities, such as the ability to generate computer code based on a text description [21]. The large amount of transfer learning in pre-trained models makes it unclear what they can and cannot do, let alone how well they might do these things.

While pre-trained models lower the barriers to AI product development, they also significantly raise costs. Some industry analysts estimate that the use of pre-trained models for web search can increase the cost 10 to 100 times [42]. Question-answering with pre-trained models also complicates the web search business model, becausse traditionally, web search service providers get paid when users click on ad links, and typically, web search questions asked to pre-trained models return answers, not ad links. For these reasons, it is difficult for innovators to know what pre-trained models can or cannot do, where they might produce more value than costs, where users might willingly accept and use new innovations, and where these innovations do not introduce ethical challenges or unintended harms that significantly diminishes a systems overall value. Innovators lack a resource that tells them where pre-trained models are currently succeeding.

We wanted to help innovators make good choices about what to innovate using pre-trained models. To help them, we chose to develop a resource of cases where pre-trained models are successful. Unfortunately, there are very few examples of commercially successful applications. Even popular services like Microsoft's Co-Pilot for programmers cost significantly more to operate than it generates from charging users [11, 29].

As the next best option, we chose to analyze the growing number of pre-trained model applications prototyped by HCI researchers. These prototype applications must demonstrate a technical capability, and they must address a user need. In addition, with the growing community interest in responsible AI, researchers developing these prototypes often look for and discuss responsible AI concerns. The one risk of failure that the research applications almost never address is financial. The systems developed by HCI researchers rarely discuss or demonstrate that the idea will generate more value than costs. While imperfect, we felt a resource documenting what pre-trained models can do that people want would be better than nothing in terms of helping innovators choose a good starting place. We analyzed a corpus of 67 HCI research applications made with pre-trained models. From this set, we noted the domains where applications reside, the AI capabilities they provide, the model performance needed to create value for users, and emerging design patterns for human-AI interaction with pre-trained models.

This paper makes three contributions. First, it provides a preliminary resource showing where innovation with pre-trained models might lead to success. This can help innovators make better choices for where to start. Second, it provides a high-level perspective on the kinds of applications HCI researchers are exploring, potentially revealing gaps for new research. Third, it offers some emerging interaction design patterns that can help both innovators and HCI researchers as they create applications, products, and services that make use of pre-trained models.

2 RELATED WORK

Our research draws on four areas of HCI research: research on HCI/UX practitioners' struggles with AI, development and assessment of AI guidelines, research to help with ideation and project selection, and emerging work on innovation with pre-trained models.

2.1 HCI Investigations of Practitioners

HCI has long focused on investigating the work of practitioners, particularly emphasizing the challenges of collaborating with data scientists within the enterprise [12, 26, 27, 30, 47, 48, 52]. For instance, studies by IBM have documented the struggles when these groups attempt to work together, particularly when there is a gap in understanding of AI [30]. Yang et al. observed that AI innovators often choose projects that are too challenging, which increases the likelihood of failure [48]. At the same time, data scientists envision AI services that customers do not actually want, while UX/HCI practitioners may design services that cannot be built [12, 26, 27, 47, 48, 52].

Additional studies have noted that UX/HCI practitioners often struggle to understand AI capabilities — what AI can and cannot do [12, 46]. In contrast, UX/HCI practitioners who have successfully integrated AI into their design processes have internalized a deep understanding of AI capabilities and relevant examples [51]. Yildirim et al found that the collaborations are most effective when UX/HCI practitioners and data scientists are co-located, allowing them to work directly and frequently [50]. This close collaboration fosters better communication, enabling more effective and practical AI innovation. Moreover, UX/HCI practitioners are increasingly developing resources and frameworks to better understand and utilize AI capabilities, helping bridge the gap between design and data science expertise [51].

2.2 Resources and Methods for Improving Ideation

Improving the idea of AI concepts that are both feasible to build and valuable to implement has been a primary emphasis of HCI research [51]. This is because most AI initiatives fail, often due to missed opportunities for leveraging simpler, more valuable AI features [7, 25, 49] — often termed low-hanging fruit [50, 53]. To address this challenge, a corpus of 40 commercially successful AI features was developed, covering 14 industrial domains and various AI methods. An important finding from this research was that 25 of these 40 features required only moderate model performance to generate significant value, demonstrating that high performance is not always necessary for impactful innovation [25].

Building on these insights, HCI researchers developed a task expertise-model performance matrix to help identify the opportunity space for innovation [15, 53]. This framework aids innovation teams in exploring AI concepts that are both valuable and easy to develop, promoting a focus on low-risk, high-value applications. The method provides significant clues for teams to envision practical AI solutions that can lead us to AI innovation.

2.3 Innovation with Pre-trained Models

In late 2022, the release of ChatGPT [36] by OpenAI triggered a wave of interest and concern around the use of pre-trained models. Unlike narrow AI, which is designed to perform a single specific task, pre-trained models leverage transfer learning to accomplish tasks that were never intended by their makers [38, 44]. This raises new challenges in understanding what these models might be able to do. The use of pre-trained models also brings significant benefits by lowering the effort required for development [18]. Since there is no longer a need to collect extensive data and build a model from scratch, innovators can simply use a pre-trained model to test their ideas. However, operating pre-trained models comes with much higher costs [40]. For example, using a pre-trained model for web search can cost 10 to 100 times more than a traditional web search. Additionally, these models are prone to "hallucinations," where they generate incorrect or nonsensical outputs.

HCI research has begun to explore the challenges of developing systems using pre-trained models. One study examined the difficulty of fixing errors that cause previously fixed errors to reappear [54]. Furthermore, many new systems are being developed to explore how pre-trained models can provide value to people, such as OpenAI's ChatGPT [36], Google's Gemini [16], Microsoft's Copilot [32], Anthropic's Claude [3], and Meta's LLaMA [31].

Our research builds on prior efforts to help practitioners envision what they can create with AI. While earlier studies have primarily focused on narrow AI, we shift the focus to pre-trained models, aiming to advance available resources and expand the scope of possibilities for innovation. We have captured insights from commercially successful applications and noted the relationship between task expertise and model performance. By integrating these approaches, our work provides a more comprehensive understanding of how we can innovate with pre-trained models by understanding what is working with pre-trained models.

3 METHOD

We wanted to help innovators by identifying what pre-trained models can do that can create value in the world. We took a designerly approach, playing with the pre-trained models as a way of understanding what they might or might not be able to do. We thought of this as engaging with pre-trained models as a design material, a subset of the prior work on AI as a design material [12–14, 28, 33, 50]. We planned to design things to gain a felt understanding of what is possible.

We began by identifying a number of tasks where we assumed pre-trained models would work, and then attempted to develop prototypes that demonstrate the capability. We explored many tasks, including providing feedback on a poster, analyzing tabular information, classifying messages, scanning resumes, and standardizing formats for references and citations. These efforts failed. For almost all of the applications we tried, we could not achieve an acceptable level of performance. Our pre-trained model applications just created more work for people, not less. We found this process frustrating. Adding to the frustration was a general level of uncertainty surrounding pre-trained models. When we could not get systems to do what we wanted, we could not easily tell if the problem was our prompting skills or if we were simply asking too much of the pre-trained models. Our frustration in ideating ideas that could be built seemed eerily similar to Yang et al.'s work on Sketching NLP [47], where researchers struggled to envision useful ideas that could be built.

Our frustration drove us to consider a different goal and a different approach. We reframed the problem. Instead of asking what pre-trained models can do, we shifted to asking, "what have people been successful at getting pre-trained models to do?" Instead of playing with and building things using pre-trained models, we shifted to analyzing applications that made effective use of this technology.

3.1 Selecting a Corpus

Building on the success of Yildirim's AI capabilities taxonomy that detailed 40 *commercially successful* AI features, we wanted to help innovators by offering them a set of pre-trained model capabilities and applications that avoided the four main causes of AI project failure [43]: (i) cannot achieve the minimally acceptable model performance, (ii) development and operational costs outweigh the application's value, users will not accept and use (typically because the system does not address a real user need), and (iv) the application has an ethical challenge that creates unintended harm. Unfortunately, there are not enough examples of commercially successful applications to create this resource.

We struggled to find a good enough proxy for commercially viable uses of pre-trained models. After much debate, we chose to analyze applications made with pre-trained models published as full papers at CHI 2024. Typically, applications published at ACM CHI must demonstrate a technical capability that is grounded in a real user need. In addition, with the growing community interest in Responsible AI, reviewers have started to require that authors at least consider the likely ethical harms of technical systems submitted for publication, such as the impact of bias in data. What CHI papers are not required to have is any demonstration of financial viability. Business concerns are largely considered to be outside the scope of research. Financial viability is largely viewed as a concern for practitioners, not researchers.

We viewed the CHI papers to be a convenient collection of artifacts that addressed two and a half of the four main risks (technical risk, user acceptance risk, and half of ethical risk). Note, we were interested in the applications built with pre-trained models, not in the research questions the researchers were asking. Our focus was on making a valuable resource that innovators might use as a starting place for envisioning new AI products and services that make use of pre-trained models.

To create the collection, we searched in https://programs.sigchi.org/chi/2024. We used the terms "Large Language Model", "LLM", "Foundational Model", and "Generative AI". Our search returned 604 CHI publications (Large Language Model: 163; LLM: 166; Large Language: 168; Generative AI; Generative AI: 100, Foundational Model: 7) in their title or abstract. We filtered out overlapping based on search criterias and papers which are unrelated to pre-trained models. Initially, we focused on those that only mentioned search terms in the abstract but were unrelated to the search terms in the entire paper. Then, we filtered the proceedings to exclude anything that was not a full paper, resulting in 140 full papers. From these, we identified 67 full papers that demonstrated applications using pre-trained models.

3.2 Artifact Analysis

We followed an artifact analysis approach [1, 19, 24, 41], analyzing the 67 AI applications. Artifact analysis comes from social anthropology. Researchers analyze the things people use as a way of better understanding people. HCI researchers have used this method to explore highly uncertain tasks like the design of a robot. DiSalvo, et al. developed an initial understanding of which features and dimensions of a humanoid robot's face most significantly influence people's perception of its humanness [10]. They used artifact analysis to gain insights into which specific facial features of robots are effective in evoking anthropomorphism. Similarly, Odom et al. applied artifact analysis in the context of slow design cases, with the ultimate goal of producing new concepts that could support innovative practices within an expanded, design-oriented theoretical framework [35].

Given our focus on trying to steer innovators to safe starting places, we chose four aspects of the applications to investigate. We looked at the domain where the application functions, we looked for capabilities, what pre-trained models can reasonably do. We looked at task expertise (how hard is the task for a person) and model performance (the minimal performance the AI needs to create value for the user). Finally, we looked for design patterns, for emerging conventions around how users might effectively engage with applications that use pre-trained models.

3.2.1 Identify Domains. Yilidirim's work on AI capabilities offered a list of 14 industrial domains where AI currently creates value for users and service providers. When analyzing the applications, we looked at who authors claimed the user of their system would be, and then we placed the application in the domain that best fit the user. For example, a system made for doctors would get placed in healthcare. In several cases, the user did not map into one of the 14 industrial domains, so we added it to Yildirim's list of domains.

3.2.2 Extract Capabilities. We extended the process used by Yildirim et al. [51], including their focus on capturing capabilities (what AI can do) and not mechanism (how it makes an inference). We followed their bottom up, inductive, approach. We first detailed specific capabilities for each application. This resulted in 241 specific capabilities. We then collapsed these into relevant clusters, hiding unnecessary detail and keeping the focus on what a pre-trained model could reasonably do.

As part of this bottom up process, we iteratively evolved a grammar for capturing a capability. [Capability (action verb)] + [Output form or structure] + [Input data]. For example: [Transcribe] + [into Text] + [from Speech]. This structure allowed us to capture each capability as a sentence and then compare it to the other capabilities to create consistent language across the examples. For example, an individual capability might use the term Talk, Speak, or Vocalize; however, when looking across the set of capabilities, we would choose the best term to bring these capabilities together, in this case, Vocalize.

To guide this inductive process, we would individually document capabilities and then meet as a group to discuss and reach a consensus on the grammar and on the terms. Throughout this process, we kept a tight focus on making choices that would make this resource of capabilities useful to innovators. Similar to Yildirim, this meant we needed to constantly consider the relevant level of granularity, the generality of the specific words we chose, and the breadth our capabilities conveyed. We worried that innovators might incorrectly assume that the pre-trained models were more capable than our dataset really indicated. For example, one capability details how pretrained models can answer a question about a product when given the description of a product. We intentionally kept the term "product" for this example to avoid innovators thinking a pre-trained model could answer a question about anything that could be described.

3.2.3 Infer Expertise and Performance. Yildirim et al.'s work showed that when designers were given a set of commercially successful AI capabilities, they still envisioned things that could not be built [51]. To overcome this challenge, they created the task-expertise/model-performance matrix. Taking their initial failure as a lesson, we followed their process and mapped the 67 applications in terms of how hard a task was for a human to perform (three levels: expertise, typical adult, less than a typical adult), and in terms of the model's minimal level of performance needed to create user value (moderate, good, and excellent). As an example, in situations where a user wants to find an image of a cat from an image dataset, the system would need only moderate model performance. If the user needed to find all the images within the dataset that showed a cat, then the system would need excellent model performance. Based on the application descriptions from the research papers, leveraging Yildirim et al's process, we made collective, subjective assessments for task-expertise and for model-performance.

3.2.4 Search for Interaction Design Patterns. To discover emerging interaction design patterns [6, 34], we analyzed the applications and the papers they were extracted from with a focus on graphics showing any interface and any text describing how the user should interact with the system. We followed a process very similar to that used by Yang et al. [49], in their work to discover design patterns for adaptive UIs used in mobile apps. To find commonalities across the 67 applications, we first identified the interaction problem or challenge the different designs addressed. In most cases this was never specifically stated by the researchers. Next, we looked for commonalities in the interaction flow and sequencing and in the layout of the interfaces and their available elements. This process resulted in four distinct interaction design patterns.

4 FINDINGS

The findings of our artifact analysis revealed the domains where pre-trained models create value, the capabilities researchers employed in their applications, the types of input and output data used, and some emerging design patterns for interacting with applications that employ pre-trained models.

4.1 Domains Where Pre-trained Models Create Value

Our mapping of the 67 pre-trained model applications to Yildirim et al.'s [51] identified industrial domains showed lots of focus on education (21), leisure (18), and office productivity (8) (see Figure 1). Researchers also created a few innovations for healthcare (4), security (2), energy (1), and marketing (1). Interestingly, none of the applications helped people working in science, human resources, government, transportation, manufacturing, agriculture, utilities, hospitality, or finance. Twelve applications did not easily fit into any of Yildirim et al.'s domains. These included creativity tools for non-professionals, tools to help professional programmers, and tools to help professional designers or other creatives. We placed these into a domain we labeled *Creative and Professional Tools*, indicated in the figure below in orange.

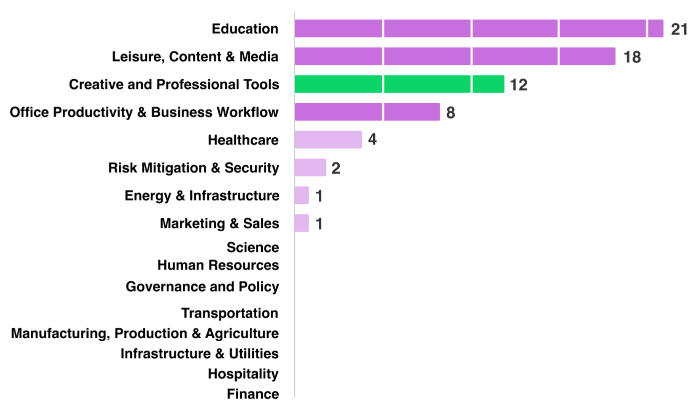


Fig. 1. Mapping of pre-trained model applications to Yildirim et al's industrial domains where AI has traditionally created value.

4.2 Pre-trained Model Capabilities

From our analysis of 67 applications, we inferred 241 individual capabilities (see Appendix for the full list). These included things like *summarize a web page based on title and initial paragraphs*, *transform into a JSON file from a pdf*, and *identify key features based on children's doodles*. We clustered these capabilities based on overlapping abilities, resulting in 33 pre-trained model capabilities (Table 1, column 1). Clustering organized capabilities by their actions and the kinds of data they took as input or produced as output. The 33 capabilities represented 13 specific actions pre-trained models can take (Table 1, column 2).

We clustered the 13 actions into three high-level categories to provide a high-level perspective on what pre-trained models can do (Table 1, column 3). This clustering was based on the quantity and form of input and output data. *Generate New Content* represents situations where pre-trained models take in a small amount of data describing what users want and return a large amount of content. *Transform Content* represents situations where pre-trained models take in and return approximately equal amounts of content that gets returned in a new form. *Understand Content* represents situations where pre-trained models take in lots of content and return small amounts of content characterizing the content that was provided. More than half of capabilities, 72% (174) fit Understand Content, 19% (47) fit Generate New Content, and 9% (20) fit Transforming Content. Among the 13 specific capabilities, 7 fit Understand Content, more than half of the total. Two capabilities fit Generate New Content, and four fit Transform Content.

The 13 specific capabilities included in each category are as follows (Figure 2):

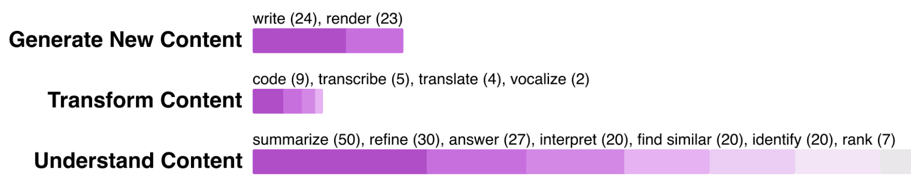


Fig. 2. 13 specific capabilities included in each category.

Among the 13 specific capabilities, we observed that certain capabilities are more frequently utilized in pre-trained model applications. For instance, Summarize (50), Refine (30), and Answer(27) were the top three capabilities with the highest counts. All of capabilities fit *Understand Content*. Within *Generate New Content*, both capabilities — write and render — had over 20 occurrences. Similarly, in *Understand Content*, six out of the seven specific capabilities had over 20 occurrences: Summarize (50), Refine (30), Answer (27), Interpret (20), Find Similar (20), and Identify (20). Interestingly, the highest count in the *Transform Content* category was Code, with 9 occurrences. The four capabilities with the lowest counts — Code, Transcribe, Translate, and Vocalize — all fit *Transform Content*.

4.2.1 Input/output Data Types. 90% (241) of the applications used text as input, and 87% (209) produced text as output. This included various forms of text including stories, descriptions, research papers, dialogues, computer code, and lists of keywords. 8% (19) of the applications used image data as input, and 12% (29) produced images as output. The images used for input and output included images of people, objects, landscape, interior space, and more. It also included complete images and segments of images.2.5% (6) of used audio for input, 0.8% (2) output audio. In all cases, audio was of human speech. Depth maps were used as output by a single application (.04%). None of the applications appeared to use time series data (e.g., web usage logs, medical records), graphs/network data (e.g., relationships, clusters), or

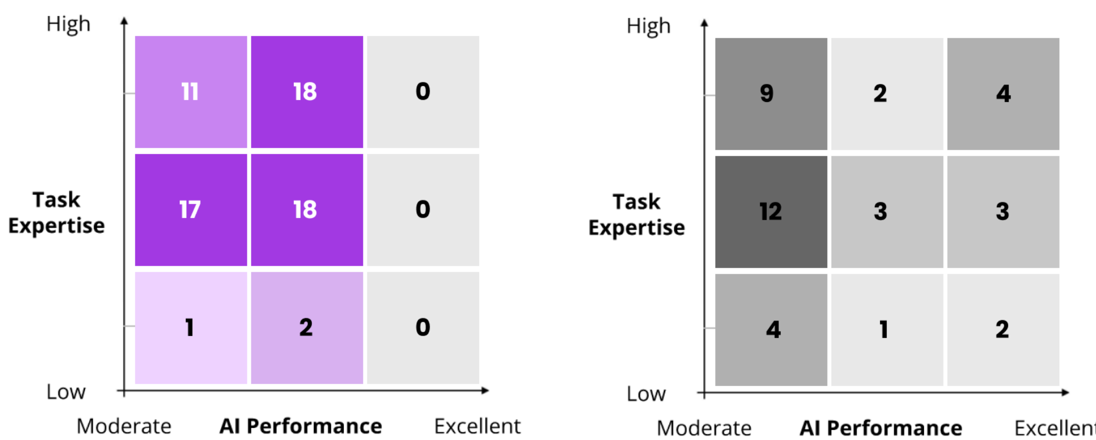
Table 1. Pre-trained Model Capabilities:
(Left to Right) 33 capability clusters, 13 specific actions, and three high-level capability themes.

Capability as [Action verb] + [Ourtput Form} + [Input Data]	capability actions and definition	capability themes
Render an image based on a topic, mood, tone, keywords, or description (15)	**Render(23)** Generate a desired image.	**Generate New Content (47)**
Render a persona image based on a persona description (2)		
Render an image that communicates a tone or mood based on an image (6)		
Write a description based on image (9)	**Write(24)** Generate a specific form of text, like a story, dialog, description, or questions.	
Write a story based on a topic or description (3)		
Write a description based on keywords (6)		
Write a character's response, based on dialog (6)		
Code into computer code based on a task description (9)	**Code (9)** Transform a description into a computer program.	**Transform Content(20)**
Transcribe into text from speech (5)	**Transcribe (5)** Transform speech into text.	
Translate into a description from computational code (4)	**Translate (4)** Transform a computer program into a description.	
Vocalize into speech based on a transcript (2)	**Vocalize (2)** Transform text into speech.	
Answer a question about a product, based on a product description (2)	**Answer (27)** Understand content and questions to provide answers.	**Understand Content(174)**
Answer a question informed by context, based on what was mentioned earlier (17)		
Answer how to do something based on a question about programming or scientific knowledge (8)		
Rank program function based on a point in computer code (3)	**Rank (7)** Understand and order actions, elements, and qualities.	
Rank personas based on description (1)		
Rank color scheme based on description (1)		
Rank keyword suggestions based on a point in prompt (2)		
Find similar keywords or document based on dialog, documents, or description (16)	**Find Similar (20)** Find similar content.	
Find similar element in the image based on a group of images (4)		
Identify an inappropriate or offensive response based on dialog (3)	**Identify (20)** Recognize specific things in content.	
Identify if a person is in an image from a tagged images (2)		
Identify sections or elements (problems, methods) based on research paper or dialog (6)		
Identify the argument from a document, image, or dialog (3)		
Identify the sentiment from a document, image, or dialog (6)		
Interpret explanation based on professional terms (6)	**Interpret (20)** Understand the subtext, the meaning of the content.	
Interpret a question to ask someone based on dialog or story (12)		
Interpret reason based on a professional knowledge (2)		
Summarize into keywords or bullet list based on a document, dialog, or diary (31)	**Summarize (50)** Summarize content.	
Summarize into a few sentences based on documents, stories, or dialog (19)		
Refine docment's tone of voice based on document and tone request (21)	**Refine (30)** Improve the quality of the content.	
Refine fix grammar error based on text (2)		
Refine into a more explicit and effective prompt based on a vague prompt (7)		

sensor data (e.g., motion, non-vocal sound, humidity, radar) as input or output, even those these are frequently used for narrow AI systems.

4.3 Task-expertise and Model-performance

We plotted the task expertise and the model performance for each of the 67 applications (Figure 3). We were surprised to see that none of the applications required excellent model performance to create value for their users. In addition, we were surprised that only three of the 67 applications had a task expertise less than a typical adult. These three included a system that documents for everyday activities, a semantic image searching tool, and a chatbot designed to reduce loneliness [5, 8, 45]. Almost all applications were evenly spread in the upper-left corner, with task expertise ranging from typical adult to expert and model performance ranging from moderate to good.



(a) Our matrix: Applications with Pre-trained models

(b) Yildirim et al.'s matrix [51]

Fig. 3. Task-expertise and Model-performance

To gain a better understanding of the difference between pre-trained models and narrow AI, we did a quick comparison between our own and Yildirim et al.'s matrix (Figure 3a and Figure 3b). Their matrix had a more even spread of applications across the matrix. It also shows the greatest density of applications in the upper left, only with greater density for moderate model performance and less density for good model performance. Examples of applications with excellent model performance from Yildirim et al. included things like medical imaging analysis (expert task/excellent performance) and biometric security (less than adult task/excellent performance). None of the applications we analyzed required this level of performance. Applications from Yildirim that required a level of expertise less than a typical adult included IoT sensing systems like smartwatch workout detection (less than adult expertise/moderate performance) and simple two-class classifiers like the biometric security. The applications we analyzed did not use pre-trained models for processing low-level sensors, nor did they focus on simple two-class classification tasks.

4.4 Interaction Design Patterns

Our analysis revealed four emerging interaction design patterns used across these applications: Variety from One, Menu of Options, Generate by Example, and Provide Drafts (Figure 4, 5, 6, 7). Below we detail these four patterns discussing the interaction challenges they address and describing how a single interface might employ more than one pattern.

4.4.1 Variety from One. When starting to create new content or to solve a problem, users sometimes do not know exactly what they want. They might have a vague idea in mind, but lack a clear picture of their desire. The ***Variety from One*** pattern addresses this challenge by generating several possible versions (image (color palette), story, document). Users choose the draft that best matches their desire, moving them closer towards their final goal. Seeing several examples can help them restate what they are trying to make. For example, C2Ideas [20] helps users mock up an interior design. Users provide a few keywords to express their desired mood and tone. C2Ideas generates eight color palettes. Users can select one or they can choose new keywords (Figure 4).

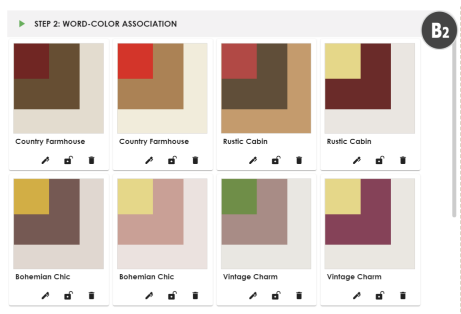


Fig. 4. Eight possible color palettes output by C2Ideas to help users design an interior [20].

4.4.2 Menu of Options. When generating new content, users can be overwhelmed with the number of possibilities, and they can struggle to understand all of the factors they should consider. They need help to reduce the uncertainty and get a handle on the overwhelming set of considerations. The ***Menu of Options*** pattern addresses this challenge by presenting users with an organized and structured set of options to consider. For example, CloChat [17] supports users in designing the persona for a conversational agent. It provides users with six different factors, each with a set of options. It helps users consider features that likely matter so they can create the conversational partner they want (Figure 5).

4.4.3 Generate by Example. When generating new content, users sometimes have a clear idea of want, while lacking the ability to effectively describe this desire. The ***Generate by Example*** pattern allows users to express what they want by providing examples that convey their desire. , making it easier to convey their intentions. For example, CreativeConnect [9] searches for and retrieves several images that are visually similar to an example shared by the user (Figure 6).

4.4.4 Providing Drafts. When creating content, users can be overwhelmed by the blank page, having too much freedom and no idea about how to begin. The ***Providing Drafts*** pattern helps by generating an initial draft that provides a "straw man" for users to respond to. It helps them get started. For example, PlantoGraphy [22] provides users with a first draft of a plan for a garden based on user provided keywords (Figure 7).

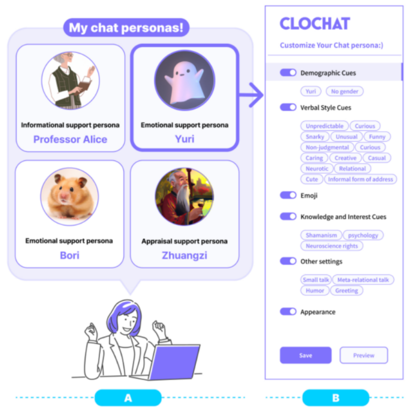


Fig. 5. CloChat presents users with visual stylistic options, demographic options, verbal style options, and more when designing a conversational partner [17].

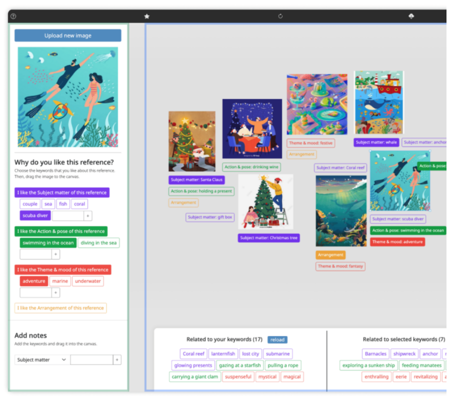


Fig. 6. CreativeConnect allows users to upload images and communicate using examples, like a mood board [9].

5 DISCUSSION

Innovators often want to reduce the risk of making things people don't want or that technically cannot be built by starting their search for opportunity based on where things are working, where products and services are currently experiencing success. We wanted to help innovators gain a better understand of what they might make with pre-trained models by developing resources that detail domains where people are developing applications, documenting the capabilities and types of data they are using, noting the quality of model performance and the difficult of the AI tasks, and offering a preliminary set of emerging design patterns. We analyzed 67 research applications, viewing this convenient set as "better than nothing," given that few commercially successful uses of pre-trained models currently exist. To clarify, while there are lots and lots of publicly available applications made with pre-trained models, it is difficult to call them *commercially successful* as most of them are not yet economically viable nor is it clear that they address a strong and persistent user need. Below we discuss the domains, capabilities, performance, patterns, and the limitations of our research.

Fig. 7. PlantoGraphy generates an initial version of the image, providing users with a starting point. Users can then adjust by modifying the layout and plant elements [22].

5.1 Pre-trained Model's Value in Understanding Content

Many people refer to pre-trained models as Generative AI or GenAI. From this, it is easy to assume innovators might use these to automate content generation as a way to create value. Many of the HCI researchers made applications built on this idea, providing tools for generating content and things. Most of the applications we analyzed focused on helping people create a new text or image artifact. What surprised us when we extracted the pre-trained models' capabilities was not the large number devoted to generation, it was the even larger number of capabilities that help with understanding and making sense of content.

Researchers leveraged pre-trained models *understand content* capabilities to produce summaries, answer questions, find similar things, create rankings and recommendations, as a classifier to monitor for or extract specific elements and items, to refine and improve content, and to interpret meanings and draw out subtextual insights. In hindsight, it seems obvious that in order to generate content, a system would also need some level of content understanding. We were impressed with the researchers' ability to draw on these less than obvious capabilities and view this as a potential starting place for innovation and as a topic for additional HCI research. An open challenge for both researchers and innovators is addressing the moderate to good level of model performance. How should researchers or innovators discover opportunities where moderate to good levels of content understanding might be valuable and useful? This seems like a ripe target for new research on AI design methods.

5.2 Uncovering Missed Opportunities and Non-opportunities

In reflecting on what HCI researchers made with pre-trained models, we began to notice things that were not showing up. We identified two areas that researchers seem to be overlooking, but—in our understanding of the technology—appear to have great potential for co-creating user and service provider value. First, we were surprised that none of the applications focused on producing many artifacts. In the case of almost every application we examined, the tools worked to help the user make a single thing or to complete a single task. This might be developing an interior design plan, the crafting of a research question based on research literature, or designing a chatbot to converse with. Interestingly, none of the applications investigated how pre-trained models might be used to make many similar versions of the same thing. Industrial designers typically make a prototype from which many copies of the same thing get manufactured. Craft is different in that a glass blower or ceramic artist is involved in production; they make each thing, often making many, many versions of the same thing. Pre-trained models seem well positioned to land between design and craft in that the user can design a thing and the model can rapidly create many slightly different, final versions. The pre-trained models can automate the production. For example, a copywriter might create an email marketing message describing a new lipstick. A pre-trained model with some details on different markets could then rapidly generate hundreds of targeted messages based on the example and the market segmentation data. One reason HCI researchers might not be exploring these types of applications could be because HCI research most often does not attend to financial or business concerns. It could also be that researchers did develop these sorts of applications, but they were not accepted for publication.

Second, the HCI research applications only touched on a small number of domains. Researchers do not seem to be exploring applications for manufacturing, agriculture, transportation, government, or finance. Interestingly, media reports share that the banking and financial services sector have traditionally been one of the fastest adopters of new technology, and some are playing with pre-trained model services, such as intelligent customer support [39]. The small number of domains explored by researchers might say more about the inchoate state of HCI research on pre-trained models or on the partnerships and collaborations researchers happen to have in 2024. Given this unexpected gap, we encourage researchers to explore these less investigated domains.

It was less surprising to see that across the capabilities described by Yildirim et al. [51], that optimization and forecasting, which seem to play a large role in narrow AI, did not appear in the applications researchers developed. We saw nothing close to well known and very successful AI capabilities such as predictive maintenance, demand prediction (smart warehousing), or digital twins, which help companies prototype new ways to optimize. We suspect that we did not see these things because they are not central to the more generative capabilities that seem to dominate pretrained models. We do not claim optimization and forecasting are not possible with pre-trained models, only that HCI researchers do not seem to be trying to get them to take on these traditional AI strengths.

5.3 Bridging the Gap for Designers

We speculate that interaction design patterns can play an important and growing role for HCI and UX designers who will be creating the next wave of interfaces for applications that leverage pre-trained models. Interaction design patterns have played a crucial role in the design process, because they provide a common framework for communication, enabling designers to address fundamental interaction problems with solutions that have a high probability of working. The interaction design patterns for pre-trained models could serve as an important bridge, helping designers create more effective solutions. In this context, our four high-level design patterns offer an overview and a starting point for identifying interaction design patterns specific to pre-trained models. For future research, we aim to investigate these patterns in more detail. What specific interaction design patterns fall under these four high-level categories? What potential UI components correspond to each pattern? We believe that identifying these interaction design patterns will provide huge value to designers, allowing them to envision pre-trained models as a new design material.

5.4 Limitations

A big limitation of our research and of the resources we developed for innovators comes from our choice to exclusively focus on pre-trained model applications from ACM CHI 2024. We could have included research using pre-trained models more broadly, from a larger set of AI researchers. However, we chose to focus on CHI as it brings with it a concern for user needs and an emerging understanding of the issues and risks around responsible AI. We feel this makes our resources incomplete, but not incorrect. We view these resources as preliminary and hope other researchers will add new capabilities, new insights on model performance and expertise, and many more interaction design patterns.

Innovators want to know where they are likely to experience success. Our use of the term "commercially successful" is meant to capture products and services that have a long history of value creation and success. Things like *restaurants* have been around for a long time, and there is already a market of people who want to go out and eat. This does not mean that every new restaurant will experience success, but it implies that trying to create a new restaurant is less risky than trying to create things that require larger changes in people's behaviors. We do not see any of the current, publicly available applications that leverage pre-trained models as commercially successful because they do not have evidence from many, many years of success. Most of the applications are funded by investors who are hopeful for future commercial success. We recognize that the HCI research corpus has a huge blind spot in terms of financial risks. However, we worried that a corpus based on things investors have chosen to fund would bring a host of hidden issues with respect to responsible AI and to real user needs.

6 CONCLUSION

Our work provides an overview of what is working with pre-trained models. From previous cases of innovation, we see that understanding what works with specific technology allows innovators to mitigate risk by providing a safe starting point. To this end, we examined applications built with pre-trained models, analyzing their domains and data types, exploring their capabilities, and assessing their model performance and task expertise. Our exploration reveals that pre-trained models have significant potential and value in understanding content, and we have identified unexplored opportunity spaces for their use. Throughout our investigation, we identified four high-level interaction design patterns that could play a crucial role in bridging the gap for designers. We advocate that our exploration of what is working with pre-trained models can provide valuable insights into finding an appropriate starting point for designing with these models, and envision new ways to innovate with cutting-edge pre-trained models.

REFERENCES

[1] Philip Adeoye. 2023. *Artifact Analysis.* Retrieved Jan 30, 2023 from https://philipadeoye.com/100_days_of_ux/artifact_analysis.html#:~:text=Artifact%20Analysis%20is%20the%20study,in%20which%20it%20typically%20exists.

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems.* 1–13.

[3] Anthropic. [n. d.]. Claude. https://claude.ai/login?returnTo=%2F%3F

[4] Apple. [n. d.]. Human Interface Guidelines: Machine Learning. https://developer.apple.com/design/human-interface-guidelines/technologies/machine-learning/introduction/

[5] Celeste Barnaby, Qiaochu Chen, Chenglong Wang, and Isil Dillig. 2024. PhotoScout: Synthesis-Powered Multi-Modal Image Search. In *Proceedings of the CHI Conference on Human Factors in Computing Systems.* 1–15.

[6] Jan O Borchers. 2000. A pattern approach to interaction design. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques.* 369–378.

[7] Claus Bossen and Kathleen H Pine. 2023. Batman and robin in healthcare knowledge work: Human-AI collaboration by clinical documentation integrity specialists. *ACM Transactions on Computer-Human Interaction* 30, 2 (2023), 1–29.

[8] Runze Cai, Nuwan Janaka, Yang Chen, Lucia Wang, Shengdong Zhao, and Can Liu. 2024. PANDALens: Towards AI-Assisted In-Context Writing on OHMD During Travels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems.* 1–24.

[9] DaEun Choi, Sumin Hong, Jeongeon Park, John Joon Young Chung, and Juho Kim. 2024. CreativeConnect: Supporting Reference Recombination for Graphic Design Ideation with Generative AI. In *Proceedings of the CHI Conference on Human Factors in Computing Systems.* 1–25.

[10] C DiSalvo. 2002. All Robots Are Not Created Equal: The Design and Perception of Humanoid Robot Heads. *Human Computer Interaction Institute and school of Design, Carnegie Mellon University* (2002).

[11] T Dotan and D Seetharaman. 2023. Big Tech struggles to turn AI hype into profits. *The Wall Street Journal* (2023).

[12] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX design innovation: Challenges for working with machine learning as a design material. In *Proceedings of the 2017 chi conference on human factors in computing systems.* 278–288.

[13] KJ Feng, Q Vera Liao, Ziang Xiao, Jennifer Wortman Vaughan, Amy X Zhang, and David W McDonald. 2024. Canvil: Designerly Adaptation for LLM-Powered User Experiences. *arXiv preprint arXiv:2401.09051* (2024).

[14] KJ Kevin Feng, Maxwell James Coppock, and David W McDonald. 2023. How Do UX Practitioners Communicate AI as a Design Material? Artifacts, Conceptions, and Propositions. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 2263–2280.

[15] Frederic Gmeiner and Nur Yildirim. 2023. Dimensions for Designing LLM-based Writing Support. In *In2Writing Workshop at CHI*.

[16] Google. [n. d.]. Gemini. https://gemini.google.com/app

[17] Juhye Ha, Hyeon Jeon, Daeun Han, Jinwook Seo, and Changhoon Oh. 2024. CloChat: Understanding How People Customize, Interact, and Experience Personas in Large Language Models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–24.

[18] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open* 2 (2021), 225–250.

[19] Bruce Hanington and Bella Martin. 2019. *Universal methods of design expanded and revised: 125 Ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport publishers.

[20] Yihan Hou, Manling Yang, Hao Cui, Lei Wang, Jie Xu, and Wei Zeng. 2024. C2Ideas: Supporting Creative Interior Color Design Ideation with a Large Language Model. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–18.

[21] HAOMIAO HUANG. 2023. *The generative AI revolution has begun—how did we get here?* Retrieved Jan 30, 2023 from https://arstechnica.com/gadgets/2023/01/the-generative-ai-revolution-has-begun-how-did-we-get-here/

[22] Rong Huang, Haichuan Lin, Chuanzhang Chen, Kang Zhang, and Wei Zeng. 2024. PlantoGraphy: Incorporating iterative design process into generative artificial intelligence for landscape rendering. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–19.

[23] IBM. [n. d.]. Design for AI. https://www.ibm.com/design/ai/

[24] Lars-Erik Janlert and Erik Stolterman. 2017. *Things that keep us busy: The elements of interaction*. MIT Press.

[25] Rafal Kocielnik, Saleema Amershi, and Paul N Bennett. 2019. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[26] Sean Kross and Philip Guo. 2021. Orienting, framing, bridging, magic, and counseling: How data scientists navigate the outer loop of client collaborations in industry and academia. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–28.

[27] Michelle S Lam, Zixian Ma, Anne Li, Izequiel Freitas, Dakuo Wang, James A Landay, and Michael S Bernstein. 2023. Model sketching: centering concepts in early-stage machine learning model design. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–24.

[28] Q Vera Liao, Hariharan Subramonyam, Jennifer Wang, and Jennifer Wortman Vaughan. 2023. Designerly understanding: Information needs for model transparency to support design ideation for AI-powered user experience. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–21.

[29] Tobias Mann. [n. d.]. Microsoft reportedly runs GitHub's AI Copilot at a loss. https://www.theregister.com/2023/10/11/github_ai_copilot_microsoft/

[30] Yaoli Mao, Dakuo Wang, Michael Muller, Kush R Varshney, Ioana Baldini, Casey Dugan, and Aleksandra Mojsilović. 2019. How data scientistswork together with domain experts in scientific collaborations: To find the right answer or to ask the right question? *Proceedings of the ACM on Human-Computer Interaction* 3, GROUP (2019), 1–23.

[31] Meta. [n. d.]. LLaMA. https://llama.meta.com

[32] Microsoft. [n. d.]. Copilot. https://www.microsoft.com/en-us/microsoft-copilot

[33] Steven Moore, Q Vera Liao, and Hariharan Subramonyam. 2023. fAIlureNotes: Supporting Designers in Understanding the Limits of AI Models for Computer Vision Tasks. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[34] Donald A Norman. 1986. User-centered System Design: New Perspectives on Human–computer Interaction.

[35] William Odom, Erik Stolterman, and Amy Yo Sue Chen. 2022. Extending a theory of slow technology for design through artifact analysis. *Human–Computer Interaction* 37, 2 (2022), 150–179.

[36] OpenAI. [n. d.]. OpenAI. https://openai.com/chatgpt/

[37] Google PAIR. [n. d.]. People + AI Guidebook. pair.withgoogle.com/guidebook

[38] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China technological sciences* 63, 10 (2020), 1872–1897.

[39] Beena Parmar Annapurna Roy. 2024. *Banking on GenAI: The artificially intelligent future of finance*. Retrieved July 16, 2024 from https://economictimes.indiatimes.com/tech/artificial-intelligence/banking-on-genai-the-artificially-intelligent-future-of-finance/articleshow/111761377.cms?from=mdr

[40] Craig S. Smith. 2023. *What Large Models Cost You – There Is No Free AI Lunch*. Retrieved Sep 08, 2023 from https://www.forbes.com/sites/craigsmith/2023/09/08/what-large-models-cost-you--there-is-no-free-ai-lunch/

[41] usabilityfirst. 2015. *Artifact Analysis*. Retrieved Jan 30, 2015 from https://www.usabilityfirst.com/glossary/artifact-analysis/

[42] Brian Wang. 2024. *IF AI LLM Queries Replace Google Internet Search*. Retrieved April 9, 2024 from https://www.nextbigfuture.com/2024/04/if-ai-llm-queries-replace-google-internet-search.html

[43] Joyce Weiner. 2022. *Why AI/data science projects fail: how to avoid project pitfalls*. Springer Nature.

[44] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3 (2016), 1–40.

[45] Anna Xygkou, Chee Siang Ang, Panote Siriaraya, Jonasz Piotr Kopecki, Alexandra Covaci, Eiman Kanjo, and Wan-Jou She. 2024. MindTalker: Navigating the Complexities of AI-Enhanced Social Engagement for People with Early-Stage Dementia. In *Proceedings of the CHI Conference on*

Human Factors in Computing Systems. 1–15.

[46] Qian Yang, Nikola Banovic, and John Zimmerman. 2018. Mapping machine learning advances from hci research to reveal starting places for design innovation. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–11.

[47] Qian Yang, Justin Cranshaw, Saleema Amershi, Shamsi T Iqbal, and Jaime Teevan. 2019. Sketching nlp: A case study of exploring the right things to design with language intelligence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[48] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining whether, why, and how human-AI interaction is uniquely difficult to design. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–13.

[49] Qian Yang, John Zimmerman, Aaron Steinfeld, and Anthony Tomasic. 2016. Planning adaptive mobile experiences when wireframing. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. 565–576.

[50] Nur Yildirim, Alex Kass, Teresa Tung, Connor Upton, Donnacha Costello, Robert Giusti, Sinem Lacin, Sara Lovic, James M O'Neill, Rudi O'Reilly Meehan, et al. 2022. How experienced designers of enterprise applications engage AI as a design material. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–13.

[51] Nur Yildirim, Changhoon Oh, Deniz Sayar, Kayla Brand, Supritha Challa, Violet Turri, Nina Crosby Walton, Anna Elise Wong, Jodi Forlizzi, James McCann, et al. 2023. Creating design resources to scaffold the ideation of AI concepts. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 2326–2346.

[52] Nur Yildirim, Mahima Pushkarna, Nitesh Goyal, Martin Wattenberg, and Fernanda Viégas. 2023. Investigating how practitioners use human-ai guidelines: A case study on the people+ ai guidebook. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–13.

[53] Nur Yildirim, Susanna Zlotnikov, Deniz Sayar, Jeremy M Kahn, Leigh A Bukowski, Sher Shah Amin, Kathryn A Riman, Billie S Davis, John S Minturn, Andrew J King, et al. 2024. Sketching AI Concepts with Capabilities and Examples: AI Innovation in the Intensive Care Unit. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–18.

[54] JD Zamfirescu-Pereira, Heather Wei, Amy Xiao, Kitty Gu, Grace Jung, Matthew G Lee, Bjoern Hartmann, and Qian Yang. 2023. Herding AI cats: Lessons from designing a chatbot by prompting GPT-3. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 2206–2220.

A 241 INDIVIDUAL CAPABILITIES

Table 2. 241 capability list

241 capabilities	**33 capability clusters**
summary: output summary of a product description	summarize into a few sentences based on documents, stories, or dialogue
correlate: output similar clusters of a topic similarity	find similar documents based on a conversation, a group of documents, or an item description
Answer: output popular search criteria based on a product	answer a question about a product, based on a product description
criteria: output an explanation of the product based on the criterias	write a description based on keywords
interpret: output the criterias from most relevant to the topic	find similar documents based on a conversation, a group of documents, or an item description
interpret: output the criterias from frequently considered criterias	find similar keywords or document based on dialog, documents, or description (16)
Rank: output most proper criteria among criterias	Rank keyword suggestions based on a point in prompt (2)

Table 2 continued from previous page

241 capabilities	**33 capability clusters**
Identify: output paper elements (problems, methods, findings) from a scientific paper	Identify sections or elements (problems, methods) based on research paper
summary: output summary of collection of abstracts	summarize into a few sentences based on documents, stories, or dialogue
correlate: output a group of similar papers from a single paper + corpus	find similar documents based on a conversation, a group of documents, or an item description
correlate: output similarities from a group of papers	
paraphrase: output a paragraph description from a list of similarities (overlapping keywords)	write a description based on keywords
find similar documents or keywords based on a conversation, a group of documents, or an item description	Identify sections or elements (problems, methods) based on research paper
Rank: output top 5 papers from smilar papers	find similar documents or keywords based on dialogue, documents, or description
interpret: output description from a list of similarities (overlapping information)	find similar documents based on a conversation, a group of documents, or an item description
Synthesize: output keywords to prompt from user's description of image and user-provided image examples	summarize into keywords or bullet list based on a document, dialogue, or diary
correlate: output overlapping featured images from overwhealmed image data	Identify if a person is in an image from a tagged images
identify: output speific images from user's description on expecting photo	Identify if a person is in an image from a tagged images
summarize: output conversation's shorter version (keywords) from conversation data	summarize into keywords or bullet list based on a document, dialogue, or diary
Filter: output some keywords from abstracted conversatino data	summarize into keywords or bullet list based on a document, dialogue, or diary
transcribe: output transcript from speech conversation	transcribe into text from speech
summarize: output shorter/organized conversation from transcripts	summarize into keywords or bullet list based on a document, dialogue, or diary

Table 2 continued from previous page

241 capabilities	33 capability clusters
Identify: output similar information with shorter conversation from stored transcript	find similar documents or keywords based on dialogue, documents, or description
vocalize: output audio from stored transcript	vocalize into speech based on a transcript
program: output JSON format data from text based information	code into computer code based on a task description
Identify: output overlapping contents with target document from relevant documents	find similar documents or keywords based on dialogue, documents, or description
Filter: output relevant information from user's request	find similar documents or keywords based on dialogue, documents, or description
Answer: output the answer from user's questions	answer how to do something based on a question about programming or scientific knowledge
program: output description of persona from JSON	translate into a description from computational code
render: outputs images of personas based on persona description	render a persona image based on a persona description
Write: output dialogue from conversation	interpret a question to ask someone based on dialog or story
context: : output character's construction from the dialogue with user	summarize into keywords or bullet list based on a document, dialogue, or diary
render: output character's background image from charcter's features(attributes, backstory, context)	render a persona image based on a persona description
Cluster: output categories of dimensions(mood) from user's poem descriptions	summarize into keywords or bullet list based on a document, dialogue, or diary
Answer: output answers from user's poem description	answer a question informed by context, based on what was mentioned earlier
paraphrase: output various tone of answers from one answer	refine docment's tone of voice based on document and tone request
Answer: output responses from specific value of dimension	answer a question informed by context, based on what was mentioned earlier

Table 2 continued from previous page

241 capabilities	33 capability clusters
transcribe: output text based transcript from visitor's speech	transcribe into text from speech
transcribe: output text transcript from user's speech requirements	transcribe into text from speech
Answer: output audio from user's speech request	answer a question informed by context, based on what was mentioned earlier
Filter: ouput text based key points from user's vague requirement transcript	refine into a more explicit and effective prompt based on a vague prompt
Answer: output the answer from visitor's inquiries	answer a question informed by context, based on what was mentioned earlier
context: ouput keywords from supplementary speech guide	summarize into keywords or bullet list based on a document, dialogue, speech transcript, or diary
sentiment understand: output user's interest&preference from user's audio reaction and rating	Identify the sentiment from a document, image, or dialog
Detect: output user's interest&preference from user's audio reaction and rating	summarize into keywords or bullet list based on a document, dialogue, or diary
Identify: output specific artwork from user's repeated speech mentions	render an image based on a topic, mood, tone, keywords, or description
describe: output captions from image segments	write a description based on image
criteria: output criteria's captions from the image captions	refine docment's tone of voice based on document and tone request
Rank: output top 10 segments from image similarity	find similar element in the image based on a group of images
render: output keywords of image mood from uploaded image	write a description based on image
refine: output relevant keyword options from selected keywords	refine docment's tone of voice based on document and tone request
extend: output longer version of image descriptions from selected keywords	write a description based on keywords

Table 2 continued from previous page

241 capabilities	**33 capability clusters**
criteria: output image explanations from selected keyword	render an image based on a topic, mood, tone, keywords, or description
Identify: output image elements from image explainations	render an image based on a topic, mood, tone, keywords, or description
Render: output image variations from image's layout boxes	render an image that communicates a tone or mood based on an image
transcribe: output transcript from elcturer's audio	transcribe into text from speech
Identify: output transcript keywords from transcript	summarize into keywords or bullet list based on a document, dialogue, or diary
Cluster: output categories(learner's understanding level) from transcript keywords	Identify sections or elements (problems, methods) based on research paper or dialog
Identify: output criterias from learner's understanding level	Identify sections or elements (problems, methods) based on research paper or dialog
Write: output dialogues from criterias for grading	write a character's response, based on dialog
summarize: output title of a paper(all the text cotents)	summarize into a few sentences based on documents, stories, or dialogue
summarize: output one-line description of paper	summarize into a few sentences based on documents, stories, or dialogue
render: output 4 images in different mood from computational prompt	render an image based on a topic, mood, tone, keywords, or description
summarize: output two-sentence description from paper abstract	summarize into keywords or bullet list based on a document, dialogue, or diary
Identify: output image elements from input image	find similar element in the image based on a group of images
context: output relationship of image elements from input image	find similar element in the image based on a group of images
synthesize: output image descriptions from generated image	write a description based on image

Table 2 continued from previous page

241 capabilities	33 capability clusters
render: output variations of images from user's preference(like/save)	render an image based on a topic, mood, tone, keywords, or description
program: output domain-oriented promts from text-based human prompt	code into computer code based on a task description
criteria: output color criterias(warm, cool, netural) from user's description on expecting room mood	summarize into keywords or bullet list based on a document, dialogue, or diary
interpret: output color descriptions from 3-color schemes	write a description based on image
identify: selected color schemes from the previous color schemes	Rank color scheme based on description
compare: output most proper visualization data from overwhealmed random data	Identify sections or elements (problems, methods) based on research paper or dialog
Filter: output selected data from overwhealmed data	find similar keywords or document based on dialog, documents, or description (16)
Identify: output UI widgets from user's text prompt	render an image based on a topic, mood, tone, keywords, or description
summarize: output shorter keywords from user's expecting image description	summarize into keywords or bullet list based on a document, dialogue, or diary
Render: output image's plant elements from bounding boxes	render an image that communicates a tone or mood based on an image
render: output scene layout from natural language prompts	render an image based on a topic, mood, tone, keywords, or description
code: output programming code from user's request on generating plots	code into computer code based on a task description
paraphrase: output structured prompt from unstructured user's prompt	refine into a more explicit and effective prompt based on a vague prompt
summarize: output keywords from interview transcript	summarize into keywords or bullet list based on a document, dialogue, or diary
Rank: output most relevant thematic codes from keywords	interpret explanation based on professional terms

Table 2 continued from previous page

241 capabilities	**33 capability clusters**
filter: output code-themes from multiple suggestions	interpret explanation based on professional terms
context: output context-specific questions from contextual audio and text information	interpret a question to ask someone based on dialog or story
render: output travel blog from images and texts of user's captured moments	write a description based on image
context: output next questions from previous conversation	interpret a question to ask someone based on dialog or story
summarize: output shorter version of document from input travel document	summarize into a few sentences based on documents, stories, or dialogue
correlate: output relevant painting themes from information similarity with user's image description	find similar keywords or document based on dialog, documents, or description (16)
context: output painting themes from user's drawing request with context	summarize into keywords or bullet list based on a document, dialogue, or diary
render: output image with selected painting theme from original image	render an image that communicates a tone or mood based on an image
program: output image generation model from painting themes	code into computer code based on a task description
cluster: output group of message types from whole dialogue	summarize into keywords or bullet list based on a document, dialogue, or diary
contextual: output counterfactuals for conflict resolution score from conversation with user	interpret reason based on a professional knowledge
identify: output conflicts from conflict confrontation response	Identify the argument from a document, image, or dialog
Write: output simulated responsed from conflict resolution score	answer a question informed by context, based on what was mentioned earlier
Detect: output irrelavant dialogues from topic similarity on the dialogue	find similar keywords or document based on dialog, documents, or description (16)
vocalize: output speech from text-response	vocalize into speech based on a transcript

Table 2 continued from previous page

241 capabilities	33 capability clusters
cluster: output relevant topic from the dialogues	find similar keywords or document based on dialog, documents, or description (16)
Write: output dialogues from previous questions and responses	write a character's response, based on dialog
rank: output five AI agents from personas and perspectives	Rank personas based on description
Answer: output response options for agents from previous answers	answer a question informed by context, based on what was mentioned earlier
paraphrase: output persuasive tone prompt from user's general text prompts	refine docment's tone of voice based on document and tone request
paraphrase: output tailored questions from user's responses	interpret a question to ask someone based on dialog or story
Write: output analogy from pre-defined text templates	write a description based on keywords
paraphrase: output news report toned analogy from original analogy	refine docment's tone of voice based on document and tone request
Render: output analgy image element options from chosen analogy strategies	render an image based on a topic, mood, tone, keywords, or description
paraphrase: output numerical values from analogy objects	refine docment's tone of voice based on document and tone request
render: output the related illustration from analogy keywords	render an image based on a topic, mood, tone, keywords, or description
paraphrase: output proper text prompt from user's vague image description	refine into a more explicit and effective prompt based on a vague prompt
render: output image from modified text prompt	render an image based on a topic, mood, tone, keywords, or description
Rank: output top ranked modifiers from text-to-image prompts	Rank keyword suggestions based on a point in prompt
render: output synonom from children's repeated speech	refine docment's tone of voice based on document and tone request

Table 2 continued from previous page

241 capabilities	33 capability clusters
paraphrase: output antonym from synonom words	refine docment's tone of voice based on document and tone request
context: output definition from a word	interpret explanation based on professional terms
cluster: output part-of-speech categories from word definitions	summarize into keywords or bullet list based on a document, dialogue, or diary
context: output image explanation from input image	write a description based on image
render: output image from user's text prompt	render an image based on a topic, mood, tone, keywords, or description
Render: output style change of image from original image	render an image that communicates a tone or mood based on an image
Identify: output segments from images	render an image that communicates a tone or mood based on an image
Identify: output image layers from image depth information	render an image that communicates a tone or mood based on an image
filter: output specific code function from programming code	answer how to do something based on a question about programming or scientific knowledge
context: output code explanations from programming code	interpret explanation based on professional terms
context: output code annotation from code explanation	summarize into keywords or bullet list based on a document, dialogue, or diary
context: output story related questions from the story	interpret a question to ask someone based on dialog or story
Write: output creative stories based on children's response	write a story based on a topic or description
context: output math term explanations from the story using math terms	interpret explanation based on professional terms
Answer: output responses from the phase of conversation with patients	answer a question informed by context, based on what was mentioned earlier

Table 2 continued from previous page

241 capabilities	**33 capability clusters**
summarize: output shorter dialogue from long dialogue	summarize into a few sentences based on documents, stories, or dialogue
Rank: output recommendation on the questions to ask to patients from previous shorter dialogue	interpret a question to ask someone based on dialog or story
paraphrase: output refined sentence from user's sentence tone request	refine docment's tone of voice based on document and tone request
summarize: output shorter version of sentence from user's draft	summarize into a few sentences based on documents, stories, or dialogue
summarize: output summarized labels from the instructions	summarize into keywords or bullet list based on a document, dialogue, or diary
paraphrase: output refined sentence from user's predetermined keywords	write a description based on keywords
context: output next sentence from existing diary sentences and keywords	write a story based on a topic or description
paraphrase: output suggestion of similar sentences from original target sentence	refine docment's tone of voice based on document and tone request
summarize: output shorter version of diary from paragraphed diary	summarize into a few sentences based on documents, stories, or dialogue
paraphrase: output revised version of sentence from original sentence and user's prompt explaining intention	refine docment's tone of voice based on document and tone request
paraphrase: output grammar revised version of sentence from original sentence	refine fix grammar error based on text
summarize: output shorter version of sentence from user's draft	summarize into a few sentences based on documents, stories, or dialogue
answer: output response for the patients from history feed related to the patients	answer a question informed by context, based on what was mentioned earlier
Answer: output the proper answers from guided topics	answer how to do something based on a question about programming or scientific knowledge

Table 2 continued from previous page

241 capabilities	33 capability clusters
Write: output dialogues from human-written dialogues with patients	write a character's response, based on dialog
context: output code descriptions in summary from student's code	answer how to do something based on a question about programming or scientific knowledge
Answer: output answer the user's programming questions	answer how to do something based on a question about programming or scientific knowledge
paraphrase: output revised version of code from user's codes	refine into a more explicit and effective prompt based on a vague prompt
Answer: output rapport from conversation with children	answer a question informed by context, based on what was mentioned earlier
Identify: output key event labels from conversation with children	summarize into keywords or bullet list based on a document, dialogue, or diary
summarize: output structured bullet summary from dialogue with children	summarize into keywords or bullet list based on a document, dialogue, or diary
sentiment understanding: output children's emotion from structured summary	Identify the sentiment from a document, image, or dialog
cluster: output negative or positive from conversation with children	Identify the sentiment from a document, image, or dialog
Filter: output inapproporate contents from conversation with children	identify an inappropriate or offensive response based on dialogue
paraphrase: output positive contents from negative contents in conversation	refine docment's tone of voice based on document and tone request
context: output feedback from children's answer (rather correct or wrong)	identify an inappropriate or offensive response based on dialogue
Sentiment understanding: output children's emotion from children's previous dialogue	Identify the sentiment from a document, image, or dialog
Answer: output answer for children's previous question	answer a question informed by context, based on what was mentioned earlier
descriptive: output the code explaination from user's code	translate into a description from computational code

Table 2 continued from previous page

241 capabilities	**33 capability clusters**
Identify: output fragments from similarity of code evaluation	find similar keywords or document based on dialog, documents, or description (16)
Compare: output the code from user's code and chatbot's code	Rank needed function for a programmer based on computer code
summarize: output summarization of prompt from user's text prompt	summarize into a few sentences based on documents, stories, or dialogue
summarize: output shorter version description from revising and confirming information	summarize into a few sentences based on documents, stories, or dialogue
paraphrase: output structured tree visualization text from random unorganized information	summarize into keywords or bullet list based on a document, dialogue, or diary
cluster: output group of categories from news	Identify sections or elements (problems, methods) based on research paper or dialog
context: output explanation and clarification from the propaganda	interpret explanation based on professional terms
Identify: output conversation pattern from the latest dialogues	Identify the sentiment from a document, image, or dialog
code: output relevant code from conversational context	code into computer code based on a task description
Write: output dialogues from previous dialogue	write a character's response, based on dialog
Identify: output drawing's specific elements from children's drawings	find similar element in the image based on a group of images
Filter: output characters and scenes from children's image descriptions	summarize into keywords or bullet list based on a document, dialogue, or diary
code: output system prompt for image generation model from children's image descriptions	code into computer code based on a task description
transcribe: output text from children's conversation in voice	transcribe into text from speech
program: output block prompting combiation from questions	refine into a more explicit and effective prompt based on a vague prompt

Table 2 continued from previous page

241 capabilities	**33 capability clusters**
context: output specific knowledge from conversation with children	answer how to do something based on a question about programming or scientific knowledge
program: output structured programming code from block prompting	summarize into keywords or bullet list based on a document, dialogue, or diary
Compare: output student's score from students' responses	identify an inappropriate or offensive response based on dialogue
Answer: output student's answer from student's questions	answer how to do something based on a question about programming or scientific knowledge
summarize: output question summarization from user's question	summarize into a few sentences based on documents, stories, or dialogue
paraphrase: output personalized story letter from gamification mission letters	refine docment's tone of voice based on document and tone request
Answer: output response from user's question	answer how to do something based on a question about programming or scientific knowledge
write: output follow-up questions from previous conversation	interpret a question to ask someone based on dialog or story
answer: output response from conversation with patients	answer a question informed by context, based on what was mentioned earlier
Detect: output vague response from conversation with patients	refine into a more explicit and effective prompt based on a vague prompt
Answer: output emotional answers from patient's previous conversation	answer a question informed by context, based on what was mentioned earlier
correlate: output related questions to ask from UI components	interpret a question to ask someone based on dialog or story
Write: output responses from actionable operation scripts	answer a question informed by context, based on what was mentioned earlier
paraphrase: output alternative versions from main story	refine docment's tone of voice based on document and tone request
paraphrase: output alternative versions of story from children's target works	refine docment's tone of voice based on document and tone request

Table 2 continued from previous page

241 capabilities	33 capability clusters
summarize: output 3-sentence abstracts from existing story for chidren	summarize into a few sentences based on documents, stories, or dialogue
summarize: output sentence blocks(keywords) from 3-sentences	summarize into keywords or bullet list based on a document, dialogue, or diary
correlate: output the specific relevant argument from the description	Identify the argument from a document, image, or dialog
context: output keywords to explain from a sentence	summarize into keywords or bullet list based on a document, dialogue, or diary
paraphrase: output refined version segments from words	refine docment's tone of voice based on document and tone request
program: output action code function from input refined segments	code into computer code based on a task description
Rank: output proper know-action from action code functions	Rank needed function for a programmer based on computer code
Answer: output answers from user's questions	answer a question informed by context, based on what was mentioned earlier
context: output specific claims from sentences	Identify the argument from a document, image, or dialog
context: output questions from specific claims	interpret a question to ask someone based on dialog or story
answer: output questions based on the request	interpret a question to ask someone based on dialog or story
sentiment understanding: output semantic context from pre-assembled database	Identify the sentiment from a document, image, or dialog
identify: output keywords for the search from the prompt	summarize into keywords or bullet list based on a document, dialogue, or diary
identify: output the name of the app	answer a question about a product, based on a product description
decode: output natural language test instructions from actionable steps	translate into a description from computational code

Table 2 continued from previous page

241 capabilities	**33 capability clusters**
context: output personal questions from conversation with user	interpret a question to ask someone based on dialog or story
context: output practical questions from conversation with user	interpret a question to ask someone based on dialog or story
identify: output metaphor keywords from users' NL prompt	summarize into keywords or bullet list based on a document, dialogue, or diary
write: output metaphorical text from storyline visualization	write a description based on image
render: output 512*512 resolution images from user's NL prompt	render an image based on a topic, mood, tone, keywords, or description
context: output text depiction from generated images	write a description based on image
program: output codes from intermediary variable	code into computer code based on a task description
program: output code function from user's natural language	code into computer code based on a task description
decode: output textual prompts from code function	translate into a description from computational code
cluster: output type of goal from rhetorical problem or writing goal	interpret reason based on a professional knowledge
Write: output phase from rhetorical problem or writing goal	write a description based on keywords
paraphrase: output revised version from user's draft	refine docment's tone of voice based on document and tone request
summarize: output shorter sentence from user's draft	summarize into a few sentences based on documents, stories, or dialogue
interpret: output student's comment options from previous selected options	answer a question informed by context, based on what was mentioned earlier
Write: output sentences from student's comment options	write a character's response, based on dialog
context: output hint-text from GUI prompt	summarize into keywords or bullet list based on a document, dialogue, or diary

Table 2 continued from previous page

241 capabilities	33 capability clusters
summarize: output hint-text(extracted keyword) from suggested input contents	summarize into keywords or bullet list based on a document, dialogue, or diary
context: output hint-text from the feedback	summarize into keywords or bullet list based on a document, dialogue, or diary
paraphrase: output diverse version of research ideation sentences from idea	refine docment's tone of voice based on document and tone request
correlate: output relevant sentence from the research sentence	find similar keywords or document based on dialog, documents, or description (16)
summarize: output summary of related wrok from the relevant research papers	summarize into a few sentences based on documents, stories, or dialogue
summary: output the shorter version of document from research paper	summarize into a few sentences based on documents, stories, or dialogue
summarize: output shorter version of document from research paper	summarize into a few sentences based on documents, stories, or dialogue
Compare: output relevant sentences from existing works	find similar keywords or document based on dialog, documents, or description (16)
paraphrase: output clean transcript from original transcript	refine fix grammar error based on text
summarize: output summarization from transcript	summarize into a few sentences based on documents, stories, or dialogue
summarize: output keywords from summary	summarize into keywords or bullet list based on a document, dialogue, or diary
summarize: output sentences from transcription	summarize into a few sentences based on documents, stories, or dialogue
paraphrase: output synonym from the keywords	refine docment's tone of voice based on document and tone request
paraphrase: output related antonym words from synonym words	refine docment's tone of voice based on document and tone request
render: output dream related image from explaination of user's dream	render an image based on a topic, mood, tone, keywords, or description

Table 2 continued from previous page

241 capabilities	**33 capability clusters**
Rank: output most proper brainstorming design concepts from keywords	Rank keyword suggestions based on a point in prompt
render: output image from text prompt and edge map	render an image based on a topic, mood, tone, keywords, or description
synthesize: output music description from the uploaded image	write a description based on image
render: output the image from the textual prompts	render an image based on a topic, mood, tone, keywords, or description
answer: output the response from textual prompts	answer a question informed by context, based on what was mentioned earlier
paraphrase: output the constructive feedback from prompts	refine into a more explicit and effective prompt based on a vague prompt
context: output the evaluation result from the violation	refine docment's tone of voice based on document and tone request
Answer: output give context related answer from the user's questions and previous answers	answer a question informed by context, based on what was mentioned earlier
context: output scenario based dialogues from selected scenario and persona	write a character's response, based on dialog
Write: output creative edge scenarios from simulation options	write a story based on a topic or description