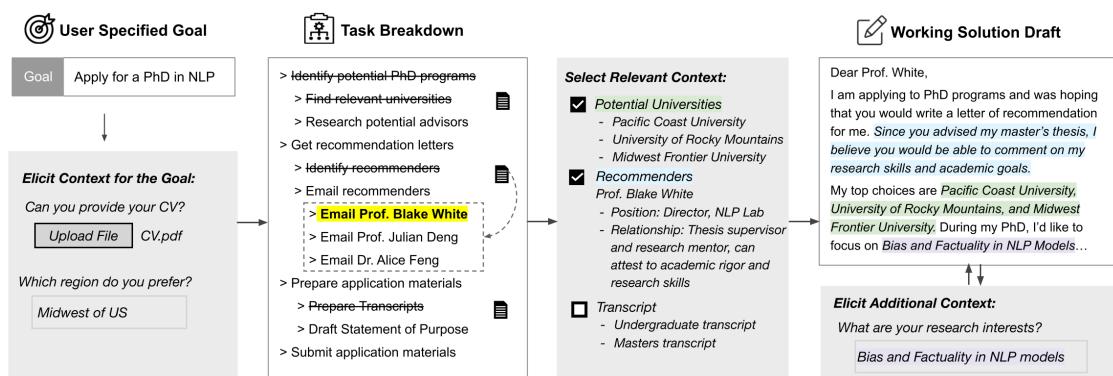


1 **JumpStarter: Getting Started on Personal Goals with Adaptive Personal Context**  
2 **Curation**

3  
4 SITONG WANG\*, Columbia University, USA  
5 XUANMING ZHANG\*, Columbia University, USA  
6 JENNY MA, Columbia University, USA  
7 ALYSSA HWANG, University of Pennsylvania, USA  
8 ZHOU YU, Columbia University, USA  
9 LYDIA B. CHILTON, Columbia University, USA  
10  
11  
12  
13



28 Fig. 1. JumpStarter helps users get started on their personal goals through adaptive personal context curation. It first takes the user's  
29 goal and elicits context for the goal. It then breaks down the goal into actionable subtasks. For each subtask, it helps users select  
30 relevant context and draft working solutions. It also aids users in refining these drafts by eliciting further context.

31 Everyone aspires to achieve personal goals. However, getting started is often complex and daunting, especially for large projects.  
32 AI has the potential to create plans and help jumpstart progress, but it often lacks sufficient personal context to be useful. Personal  
33 context includes background information, preferences, and resources that can help tailor AI responses to individuals and their goals.  
34 We introduce JumpStarter, an AI-powered system that creates action plans and drafts by adaptively soliciting and integrating personal  
35 context. JumpStarter assists users by posing questions to elicit relevant context, breaking down goals into manageable steps based on  
36 context, and identifying relevant context to draft working solutions. A technical evaluation indicates that context curation results  
37 in plans and working solutions of higher quality. A user study demonstrates that compared to ChatGPT, JumpStarter significantly  
38 reduces mental load and increases efficiency for users. We discuss the design implications of adaptive personal context curation to  
39 better support users in complex problem solving.  
40  
41

42 CCS Concepts: • Human-centered computing → Interactive systems and tools.

43 Additional Key Words and Phrases: context curation, planning, action initiation, personal goal management, productivity

44 \*These authors contributed equally to this work.

45  
46 Authors' Contact Information: Sitong Wang, Columbia University, New York, NY, USA, sw3504@columbia.edu; Xuanming Zhang, Columbia University,  
47 New York, NY, USA, xz2995@columbia.edu; Jenny Ma, Columbia University, New York, NY, USA, jm5676@columbia.edu; Alyssa Hwang, University of  
48 Pennsylvania, Philadelphia, PA, USA, ahwang16@seas.upenn.edu; Zhou Yu, Columbia University, New York, NY, USA, zy2461@columbia.edu; Lydia B.  
49 Chilton, Columbia University, New York, NY, USA, chilton@cs.columbia.edu.

## 53    1 Introduction

54  
 55 Achieving personal goals is a common aspiration. Personal projects are part of everyday life, ranging from moving  
 56 to a new apartment, to starting a YouTube channel, to applying for PhD programs. Getting started on these personal  
 57 projects is crucial to turn intention into actual fulfillment [19, 47]. However, getting started on complex goals can  
 58 be daunting and challenging. When faced with a new project, one needs to engage in sensemaking: constructing a  
 59 mental model of the process and adapting it to their specific situation to figure out the actionable steps, all of which  
 60 can be cognitively demanding [26]. Another major challenge is initiating concrete actions. For example, there is a  
 61 significant gap between knowing when one needs to write a request email and actually being ready to write it [47].  
 62 This shift requires concretization—such as defining the exact purpose of the request, deciding on the appropriate tone,  
 63 and gathering the necessary information like relevant background details and supporting documents. Identifying and  
 64 utilizing the right resources to operationalize the plan can be a significant hurdle to getting started.  
 65

66 Generative AI has the potential to create plans and help jumpstart progress, but it often lacks the personal context  
 67 needed to be useful. Everyone has unique situations and requirements to consider when embarking on personal projects.  
 68 For example, to get started on the goal of “applying for a PhD,” it is important to consider personal context, such as one’s  
 69 academic background and geographic preferences. Moreover, a complex goal like this must be further decomposed to  
 70 become actionable. It involves multiple tasks, such as researching universities and requesting recommendation letters,  
 71 each of which requires further planning and coordination with other tasks. For example, requesting recommendation  
 72 letters is itself a sub-goal that needs further breakdown—one must first identify recommenders, gather supporting  
 73 documents, draft individualized request emails, send the emails, and follow up accordingly. Each of these tasks demands  
 74 adaptive curation of relevant context to craft personalized solutions. For example, when drafting emails to recommenders,  
 75 it is essential to solicit relevant context to ensure that the drafts are personalized to the recipient, which might include  
 76 one’s relationship with the recommender, the programs being applied to, and aspirations for doctoral studies.  
 77

78 Popular AI tools like ChatGPT allow users to provide context throughout their interactions. However, it can be  
 79 mentally demanding for users to recall and articulate the necessary context relevant to the task and to structure all the  
 80 information within the bounds of prompts and chat history. Ideally, ChatGPT would remember all the context in the  
 81 chat and automatically integrate the most relevant ones to support each task. However, this capability is limited by  
 82 the inherent restrictions on context window size [9, 43, 57], and the context integration process is not transparent to  
 83 users. Moreover, linear chat interfaces like ChatGPT lack the structure to help people systematically plan their roadmap  
 84 toward their goals. Although people can ask ChatGPT to break down the steps further, it is easy to lose track of their  
 85 progress and the context of each task.

86 Furthermore, theories about personal goals link their successful implementation to executive functions such as  
 87 planning, task initiation, and working memory [16], so approaches should be grounded in these functions. Planning  
 88 involves a step-by-step breakdown of a broader goal [32]. This can be challenging as it requires identifying key tasks,  
 89 estimating realistic timelines, and foreseeing potential obstacles, all of which demand significant effort [50]. While  
 90 having a plan is crucial, it can sometimes remain abstract and lack the specificity and granularity required for tangible  
 91 real-world actions [19, 32]. Initiating concrete tasks demands heightened focus and meticulous attention to detail,  
 92 which significantly increases cognitive load [49]. Effective execution of such tasks requires an active working memory,  
 93 which aids in retaining and manipulating the relevant information needed to manage the task [4]. Additionally, the  
 94 Ovsiankina effect [46] indicates that people are more likely to finish the task once they initiate it, underscoring the  
 95 critical role of action initiation in goal completion.

To better support people in getting started on their personal goals, we introduce a system called JumpStarter (see Figure 1). Through mechanisms that adaptively solicit and integrate personal context, the system assists users by 1) decomposing tasks into hierarchical steps and 2) drafting working solutions. Here, the “working solution” refers to any tangible result that enables users to take real-world action. For example, a working solution could be a draft email to send to a recommender or a study schedule for the GRE. The process begins when a user inputs their personal goal. The system then poses questions to elicit the relevant context for the goal. Based on the user’s responses, the system breaks down the goal into subtasks the user can explore. If the system detects that a subtask is not actionable enough, it prompts the user, who can break it down further into additional subtasks. When the user is ready to work on a task, the system selects the relevant context and generates working solutions. If the user is not satisfied with the suggested solution, they can request more elicitation questions or provide their own prompts for refinement. Users can modify the suggested solution, and also save it as context to be used by the system in further subtasks. The user can then move to explore the next subtask, moving towards the completion of their goal.

This paper makes the following contributions:

- JumpStarter, an AI-powered system for helping people get started on their personal goals with three main automated features: 1) adaptive personal context curation (including eliciting, saving, and selecting context), 2) hierarchical task decomposition, and 3) working solution generation.
- A technical evaluation with and without context curation, showing the importance of context curation for getting plans and working solutions of high quality.
- A user study showing that compared to ChatGPT, JumpStarter significantly reduces mental load and increases efficiency in getting started on personal projects.

We conclude by discussing the design implications of adaptive personal context curation to guide the design of future systems that support complex problem solving.

## 2 Related Work

### 2.1 Theory of Goal Achieving

Achieving personal goals is a complex process that involves various aspects of human psychology. This process encompasses numerous psychological factors, including motivation, self-belief, social support, feedback, planning, and more, as highlighted in theories like goal-setting theory [38] and the theory of planned behavior [3]. To better understand how individuals navigate this intricate journey, action phase theory offers a comprehensive framework [19, 20]. It outlines the progression from initial desires to goal completion through distinct stages: pre-decisional, pre-actional (planning), actional, and post-actional phases, each critical in transforming aspirations into accomplishments. Action phase theory also emphasizes the importance of action initiation, as it marks the transition to the actional phase where individuals actively pursue their goals [19, 20].

While other factors such as motivation and self-belief [3, 14, 40] are also crucial, we recognize the potential of computers in supporting the planning and action initiation aspects, especially for goals that primarily involve cognitive tasks or knowledge work (such as applying to PhD programs). In this work, we focus on providing support for individuals during the planning and action initiation phases of achieving their complex cognitive goals, helping them bridge the gap between mere goal intention and taking real-world action.

Planning presents its own set of challenges. In a probe study focusing on managing academic procrastination [6], a majority of participants (13 out of 15) expressed a desire for structured action steps for their tasks. Many participants

157 pointed out the struggles with unstructured or expansive tasks, indicating a need for planning support that includes  
 158 step-by-step guidelines, structuring recommendations, and adaptive task breakdown. According to the implementation  
 159 intention theory, plans that specify when, where, and how actions should be taken can transform abstract goals into  
 160 concrete strategies, bridging the gap between intention and action [21]. Similarly, goal-setting theory emphasizes the  
 161 need for specific and actionable goals [38]. Personalization is emphasized as another important dimension, suggesting  
 162 that adapting goals to fit one's real-life situations can significantly enhance the effectiveness of planning [17].  
 163

164 Action initiation is also challenging. A major hurdle is the need to take concrete actions—each task requires  
 165 specific context to be acted on. However, people often juggle multiple tasks in their daily lives [13]. Frequent context  
 166 switching between tasks can significantly drain cognitive resources, reducing the efficiency with which one can  
 167 work on a project [31]. To initiate action, context curation is necessary to create an enabling environment and to aid  
 168 in concretization. The Fogg Behavior Model suggests that to persuade people to take action, one needs to provide  
 169 facilitators [18]. An effective facilitator should inform people that the target behavior will not require a resource they  
 170 do not have at that moment. Context curation—which involves creating a conducive environment tailored specifically  
 171 for the task at hand—can thus facilitate the initiation of action [53].  
 172

## 173 2.2 Solutions to Support Action Planning

174 An action plan refers to a concrete, step-by-step breakdown of a task that helps people complete it [32]. Prior studies  
 175 show that people prefer not to make action plans for their own tasks [33, 50], as the costs are immediate but benefits  
 176 are deferred [29]. Therefore, it will be beneficial to provide people with support for action planning.  
 177

178 There are many HCI works focusing on community or crowd-based approaches to help people do a step-by-step  
 179 breakdown of their goals. For example, TaskGenies [32] decomposes tasks into concrete steps by using online crowds  
 180 to create new plans and using algorithms to identify and reuse existing ones. PlanHelper [37] helps users construct  
 181 activity plans (such as for bodybuilding or sightseeing) by processing answer posts in community-based QA platforms.  
 182 Specifically for writing tasks, a vocabulary [29] is introduced as a cognitive scaffold to enable other people who  
 183 have the necessary context (such as collaborators), to create action plans for writers. In a collaborative teamwork  
 184 setting, MixTAPE [44] helps generate action plans for design teams by converting call notes and client briefs into  
 185 prioritized, assigned tasks using a synthesis algorithm and machine learning models. While not directly supporting  
 186 action planning, many other works, particularly in the crowdsourced field, focus on decomposing a large task into  
 187 manageable micro-tasks that help people make progress with limited time and resources [8, 50, 54].  
 188

189 With the advancement of generative AI techniques, large language models (LLMs) like GPT-4 [1] hold the potential  
 190 to provide intelligent support for planning. There is a series of NLP works that study the potential of automating  
 191 real-world planning [34, 56, 61]. For example, a multi-agent approach [34] is introduced to determine the steps needed  
 192 to complete a task (such as how to plant a garden), emphasizing the importance of tailoring the steps to meet users'  
 193 specific needs (such as planting a garden without using pesticides). However, studies found that the current language  
 194 agents are not yet capable of automatically handling complex planning tasks [56, 61]; humans still need to be involved  
 195 to make meaningful plans for real-world tasks.  
 196

197 A popular online tool called Goblin Tools MagicTodo<sup>1</sup> uses AI techniques to support people in breaking down their  
 198 tasks layer by layer. Throughout our experiments with MagicTodo, we identified significant limitations, particularly in  
 199 determining the appropriate level of task breakdown granularity. In theory, tasks can be endlessly decomposed into  
 200

---

201<sup>1</sup><https://goblin.tools/>

trivial steps, such as opening a browser. The challenge is to find a manageable granularity that remains meaningful for users. To address this, JumpStarter examines subtask detection to help users decide whether a task requires further breakdown while still allowing for personal control. A recent HCI work, ExploreLLM [39], also explores helping users decompose complex tasks into subtasks with LLMs. Their study shows that a schema-like structure is useful for people to do planning tasks, with ChatGPT as a baseline. The system also provides a dedicated input box to prompt users to enter their personal context and preferences for personalization. However, ExploreLLM only examines one layer of task decomposition and relies on users to provide context themselves. In JumpStarter, we explore multi-layer task decomposition and adaptive personal context curation throughout the process.

### 2.3 Solutions to Support Action with Context Curation

Context curation is a fundamental design concept that impacts productivity. It involves eliciting, saving, and selecting relevant information to tailor solutions for specific tasks and individuals [30]. The idea of "readiness-to-hand" [24] illustrates how we perceive tools as extensions of our abilities, emphasizing the close intertwining of cognition and surroundings. For example, mise-en-place [53], a strategic arrangement of tools and ingredients before cooking, demonstrates how well-curated contexts can enhance efficiency. According to cognitive theories, our thoughts and actions are adaptive responses to our environment, underscoring the deep connection between human cognition and context [55]. This structural coupling highlights the significance of context curation in improving productivity and interactions with technology.

Context elicitation has been studied in the NLP field. Research in proactive information-seeking dialogue [15, 59] focuses on approaches to asking clarifying questions [41, 58] and eliciting user preferences [22, 35, 60] to better meet users' information needs. However, this body of research often centers on pure information-seeking tasks, such as web searches or online purchases, and the experiments are often conducted in simulated chat environments. JumpStarter broadens this scope by facilitating interactions with real users across a wider variety of real-world tasks in a structured manner. Moreover, it advances this research by eliciting context beyond typical question-and-answer exchanges and incorporating personal files into the process, making context elicitation more practical.

Context saving and selection were studied in task-centric information management systems. In terms of context saving, UMEA [28] helps users create and manage project-specific work context by providing dedicated project spaces and monitoring user activities within a desktop environment. The Personal Project Planner [27] introduces an in-context creation feature that enables users to produce documents and emails within a project plan, automatically inserting hyperlinks to provide context to the task. Similarly, Towel [10] allows users to drag any semantic object (e.g., project, email, person) onto the to-do list, saving them as resources under the to-do item. Context selection has been particularly examined in programming scenarios. For example, Mylar [31] introduces a task context model to help programmers find and identify information relevant to their current task by utilizing task interaction history. JumpStarter not only streamlines the integration of context into users' tasks but also dynamically optimizes task execution, transforming static context management into a proactive and adaptive workflow catalyst.

## 3 JumpStarter System

JumpStarter (see Figure 2) is an interactive system that helps users get started on their personal projects with adaptive personal context curation. The system takes the user-specified goal as the input, and outputs personalized plans and working solutions for the user. Here, a working solution refers to any tangible result that can assist users in taking

real-world actions related to a task. For instance, a working solution for requesting recommendation letters might be an email draft to send to the advisor. Similarly, for passing the GRE test, a working solution could include a study schedule.

Overall, JumpStarter utilizes a hierarchical tree structure. We chose this approach to organize tasks because it offers a clear, logical flow that naturally mirrors the breakdown of complex goals into manageable components. This structure facilitates easy tracking of task dependencies and progression. Features such as collapsing or expanding nodes allow for customizable views, ranging from simple outlines to detailed oversight. Additionally, the ability to attach documents directly to task nodes enhances organization and accessibility, making this approach particularly effective for managing personal projects.

JumpStarter is powered by GPT-4<sup>2</sup> and implemented with a Flask/Python web framework. In this section, we describe how our system works with an example walk-through, and the implementation details of the system.

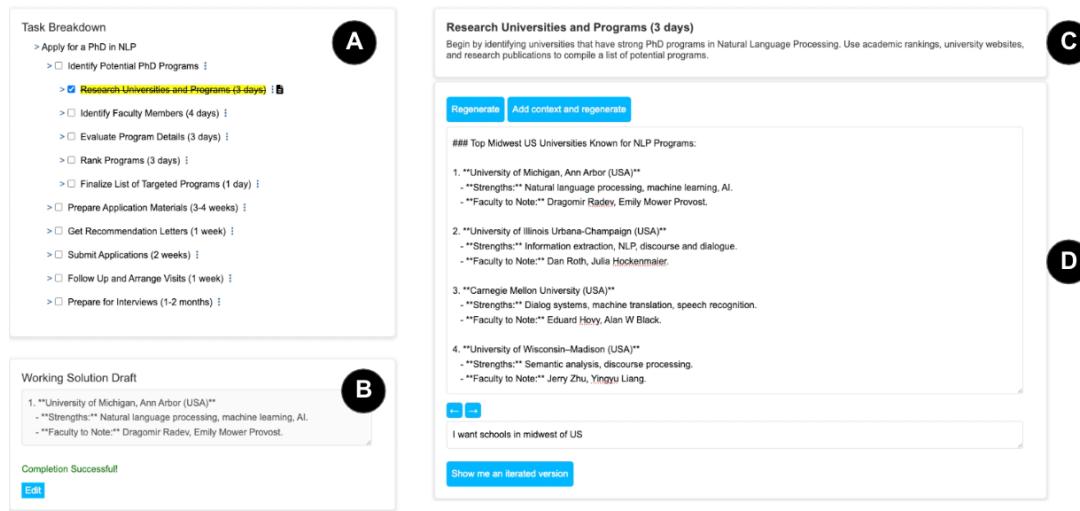


Fig. 2. A screenshot of JumpStarter creating plans and drafting working solutions for the goal *Apply for a PhD in NLP*. (A) Task breakdown is shown as a subtask tree, with the goal being the root node. Subtasks decomposed from the same parent node are shown on the same level. (B) Saving the working solution draft. (C) Detailed descriptions of the selected subtask are shown. (D) The working solution is generated, considering the specification from the user – “*I want schools in midwest of US*”. Users have three options to improve the draft: regenerate, add context and regenerate, and iterate based on users’ new specifications.

### 3.1 System Description

As an example, consider John, who aims to apply for a PhD in NLP. In this subsection, we will demonstrate the typical interactions between John and our system through a detailed walk-through.

First, John establishes his goal by entering “*Apply for a PhD in NLP*” into a text box. He then clicks the *Start* button to submit it.

**3.1.1 Context Elicitation for Goal.** Based on John’s goal, the system uses the LLM to generate questions to elicit context from him (see Figure 3). For John, the system generates three questions: “*Can you provide your CV or Resume?*”, “*Which universities or programs are you considering for your PhD in NLP?*”, and “*Do you have letters of recommendation, or do you*

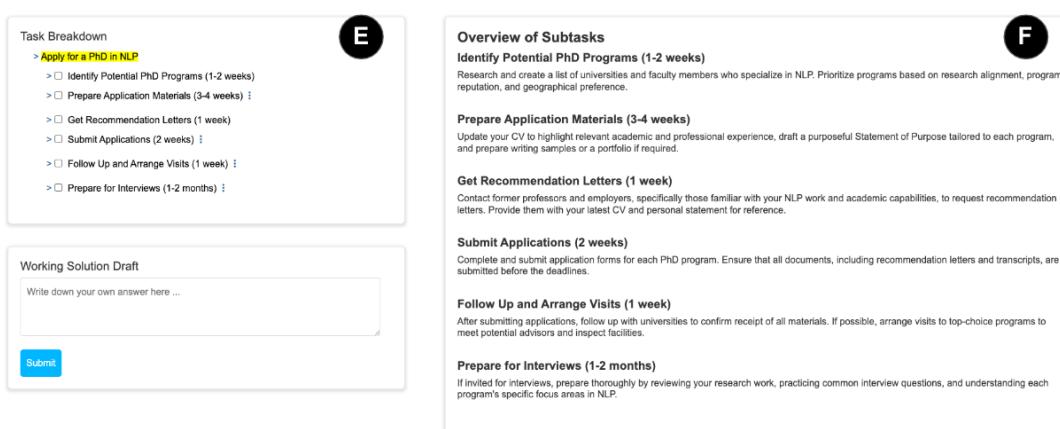
<sup>2</sup>We used gpt-4-turbo throughout the system.

313 need guidance on how to obtain them?" Since John does not know which universities he might be applying to, and he  
 314 has not obtained the recommendation letters, he only uploads his CV to the first question. Then, he clicks the *Let's get*  
 315 *started* button to proceed. This first step aims to clarify John's goal by eliciting the necessary context that will be used  
 316 globally throughout John's interaction with JumpStarter, ensuring that the plans and working solutions are created  
 317 based on John's situations.  
 318

The screenshot shows the 'JumpStarter' application interface. At the top, there is a header bar with the title 'JumpStarter' and a search bar containing 'Apply for a PhD in NLP'. Below the header is a large input field with the placeholder 'Can you provide your CV or Resume?' and a 'Choose File' button with the file name 'CV.txt'. To the right of this is a 'Start' button. Below this input field, there are three smaller input fields: one for 'Which universities or programs are you considering for your PhD in NLP?', another for 'Do you have letters of recommendation, or do you need guidance on how to obtain them?', and a third for 'Add your own context'. Each of these has a corresponding input field below it. At the bottom of the form is a blue 'Let's get started!' button.

320 Fig. 3. JumpStarter generates questions to elicit context from users to clarify the goal. The user uploads his CV.  
 321  
 322  
 323  
 324  
 325  
 326  
 327  
 328  
 329  
 330  
 331  
 332  
 333  
 334  
 335

336 3.1.2 *Subtask Suggestion*. Leveraging the context obtained from the first step, JumpStarter generates the initial subtask  
 337 tree for John, as depicted in Figure 4E, which consists of a list of subtasks. It also presents John with an overview of the  
 338 titles, descriptions, and duration of completion of these subtasks (see Figure 4F). After reviewing them, John gains brief  
 339 insights into each subtask and decides to explore them in order.  
 340  
 341  
 342  
 343



344  
 345 Fig. 4. The initial subtask tree and the overview for the goal *Apply for a PhD in NLP*. (E) The task breakdown for the goal. (F) The  
 346 overview of all the subtasks of the goal, including the titles, descriptions, and duration of completion of the subtasks.  
 347  
 348  
 349  
 350  
 351  
 352  
 353  
 354  
 355  
 356  
 357  
 358  
 359  
 360  
 361  
 362

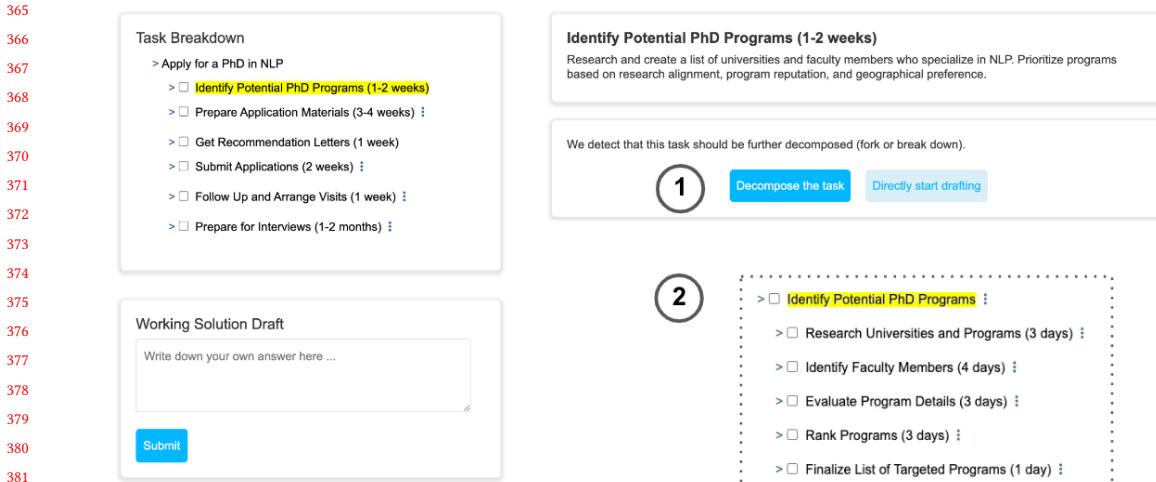
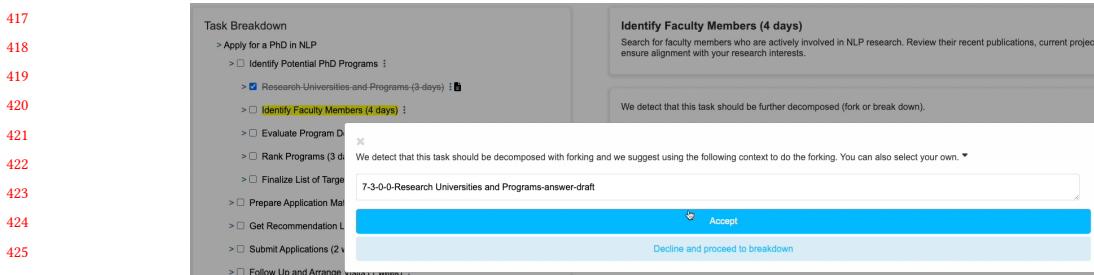


Fig. 5. JumpStarter suggests further decomposition for the first subtask *Identify Potential PhD Programs*. (1) John presses the button *Decompose the task*. (2) JumpStarter generates the subtasks for John.

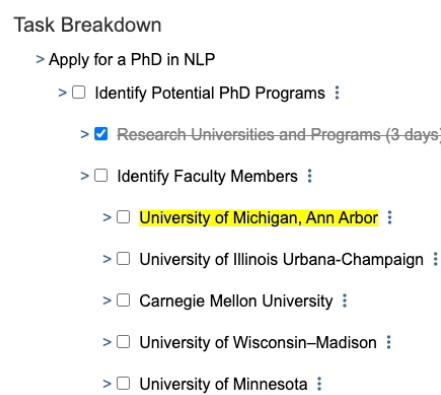
3.1.3 *Subtask Detection.* After reviewing the initial subtask tree, John thinks the list generally makes sense and proceeds to click on the first node: *Identify Potential PhD Programs*. As shown in Figure 5, the node is highlighted, and its title and description are shown on the top right of the interface. At the same time, the system shows a message as it detects this task needs to be further decomposed to make it actionable. The system offers two options for the user to select: the recommended option *Decompose the task* and the alternative option *Directly start drafting*. Clicking on the first button will further break down the selected task into several subtasks, while clicking on the second will generate a working solution for the task. Following the recommendation, John clicks the button *Decompose the task*, after which the subtasks for *Identify Potential PhD Programs* are generated, as shown in Figure 5.

3.1.4 *Working Solution Drafting.* John reviews all subtasks of *Identify Potential Programs* and proceeds to click on the first subtask *Research Universities and Programs*. JumpStarter deems this subtask actionable and suggests John directly start drafting the working solution for it. Following the suggestion, he obtained the initial working solution that contains a list of universities offering PhD programs in NLP. Upon reviewing the list, John realizes it includes many global institutions. However, he prefers schools in the midwest of the United States. To iterate, he enters "*I want schools in midwest of US*" (see Figure 2D). After reading it, John is happy to have this list as the first draft of his school application list. Note that John is provided with three alternatives to further improve the draft, as shown in Figure 2D: 1) regenerate; 2) add more context and regenerate; and 3) specify details and iterate on the current draft. Subsequently, John copies and pastes the draft into the working solution section of the task *Research Universities and Programs* (see Figure 2B) and saves it, which appears as a file icon on the tree (see Figure 2A) with the subtask crossed out.

3.1.5 *Forking Detection and Context Selection.* After obtaining the initial working solution for *Research Universities and Programs*, John moves on to the next subtask titled *Identify Faculty Members*. JumpStarter suggests that the subtask lacks the specificity needed for direct action and suggests further breakdown. John agrees and clicks the *Decompose the task* button. Following this, JumpStarter determines that the subtask should be forked to address the specifics of each



(a) Selecting relevant context for forking.



(b) Entity-based task decomposition (forking).

Fig. 6. Context selection for forking and the task decomposition after applying forking on *Identify Faculty Members*.

program identified earlier. It then selects and presents the relevant context, specifically the working solution of the previously completed step *Research Universities and Programs* that includes a list of universities (see Figure 6a). John agrees with the suggestion and clicks the *Accept* button to proceed with forking using the selected context. Note that if John disagrees with the suggested context, he can select other context variables from a dropdown button; alternatively, he can also *Decline and proceed with breakdown* if he thinks that the subtask should not be forked. The updated subtask tree after forking is displayed in Figure 6b. With the new subtasks of evaluating different programs, John proceeds to explore each program in greater detail by clicking on them sequentially.

**3.1.6 Context Selection for Working Solution Generation.** Next, John decides to explore the task *Get Recommendation Letters* and decomposes it as the system suggests. John selects the first subtask *Compile a list of recommenders*. JumpStarter uses his CV uploaded in the first step to generate a draft of recommenders: Prof. Blake White, Prof. Julian Deng, and Dr. Alice Feng. John likes the draft and saves it. Subsequently, John moves on to the subtask *Reach out to potential recommenders*, and JumpStarter breaks down the task based on the recommenders identified in the previous step. John then selects the subtask *Reach Out to Potential Recommenders: Prof. Blake White*, which requires “Send a personalized email to Prof. Blake White about your PhD aspirations.” In addition to John’s CV, JumpStarter suggests two more relevant context: the information of each recommender (contains the relationship between Prof. Blake White and John) and the list of potential schools, as shown in Figure 7a. Following the suggestion, John receives the initial email draft.

469     3.1.7 *Context Elicitation for Working Solution Iteration.* After reviewing the initial draft of the email for Prof. Blake  
 470     White, John decides it needs improvement, though he is unsure how to proceed. He clicks the "Add Context and  
 471     Regenerate" button, prompting JumpStarter to generate several questions designed to draw more context from him  
 472     (See Figure 7b). John then includes details about specific projects and papers he collaborated on with Prof. White and  
 473     regenerates a refined draft. John likes the new version because it highlights specific aspects of his past collaborations  
 474     with Prof. White, giving it a more personalized touch.  
 475

476

477

478

479

480

481

482

483

484

485

486

487

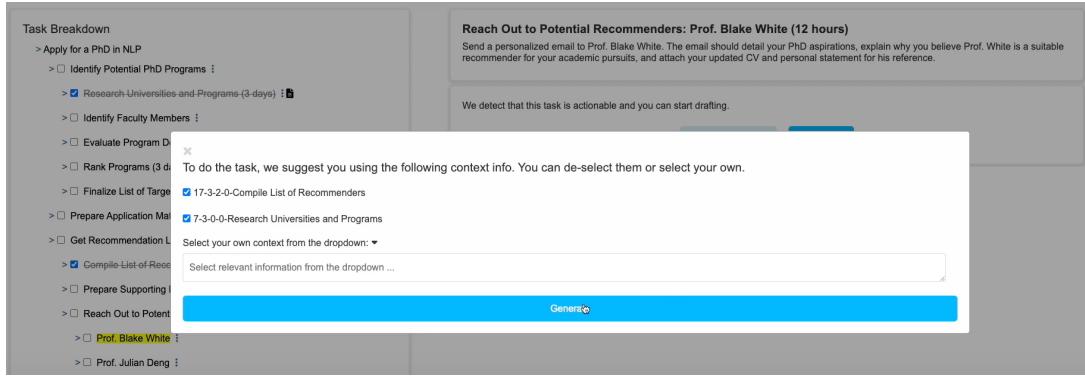
488

489

490

491

492



(a) Context selection for working solution generation.

493

494

495

496

497

498

499

500

501

502

503

504

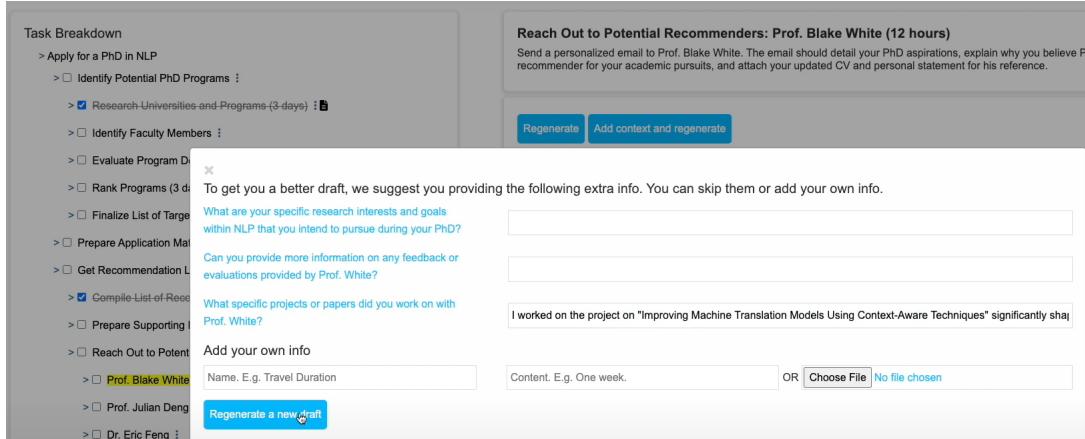
505

506

507

508

509



(b) Context elicitation for working solution iteration.

510

511

512

513

Fig. 7. Context selection and elicitation for drafting and iterating the working solution of the subtask *Reach Out to Potential Recommenders: Prof. Blake White*.

514

515

516

517

518

519

With all of these functionalities of JumpStarter, John manages to generate a plan and working solutions that he is satisfied with. This enables him to progress further and ultimately achieve his goal of applying for a PhD in NLP.

520

521    **3.2 Data Representation**

522  
 523 In our system, each context is represented as a key-value pair, where the key denotes the context’s name, and the value  
 524 is its content. For instance, the pair “Location preference: Midwest of US” indicates the user’s preference for a specific  
 525 location—Midwest of US.

526 There are two types of context in our system: global context and local context. Global context refers to context that  
 527 is universally applied by default throughout the system. Specifically, it consists of all context elicited from the questions  
 528 the system asks at the very beginning after the user enters their goal. The local context includes all saved working  
 529 solution drafts and the context added by the user throughout their interaction with the system.

530 Additionally, we represent lists of subtask nodes generated by our system in the form of a tree, with the user’s goal  
 531 serving as the root node. Each subtask node stores task-specific attributes, including working solution drafts, task titles,  
 532 task descriptions, and estimated duration of completion, among others.

533    **3.3 Feature Implementation**

534 In this section, we provide implementation details for the key features of our system, as introduced in Section 3.1. Note  
 535 that the user’s goal and the global context are integrated into the system prompt, which we use to implement all of our  
 536 key features.

537    *3.3.1 Context Elicitation for Goal.* Based on the user’s input goal, JumpStarter poses questions to elicit context relevant  
 538 to that goal from the user. To achieve this, we instruct GPT-4 to assess which existing documents the user might possess  
 539 that could serve as the context. GPT-4 will suggest what file to upload along with an elicitation question. Our system  
 540 then lists that question in the UI and allows the user to upload their file. Alternatively, if GPT-4 generates a question  
 541 that does not require any documents from the user, an input text box will be provided in the UI, allowing users to  
 542 directly type their responses. See the GPT prompts we used in Appendix A.1.

543    *3.3.2 Subtask Suggestion.* Given a task that the user decides to break down, JumpStarter decomposes it into a list of  
 544 subtasks and attaches them as child nodes to the task node. We instruct GPT-4 to generate the list, where each subtask  
 545 should have a name, associated descriptions, and an estimated duration for completion. This helps users understand the  
 546 scope and duration of each task. To minimize the occurrence of repetitive subtask nodes across the tree, we prompt  
 547 GPT-4 to consider the overall tree structure and suggest subtasks that naturally complement the existing hierarchy. See  
 548 the GPT prompts we used in Appendix A.4.

549    *3.3.3 Subtask Detection.* Given a task that the user selects, JumpStarter evaluates whether it is actionable and sufficiently  
 550 detailed. If it is not actionable, JumpStarter recommends further breakdown. If it is actionable, JumpStarter recommends  
 551 that the user directly drafts the working solution. We use GPT-4 to perform subtask detection. To determine the optimal  
 552 prompting method, we experimented with different prompting techniques. To optimize both the accuracy and user  
 553 experience in terms of loading time, we decided to adopt Chain of Thought (CoT) prompting with few-shot examples  
 554 and integrate the tree level of the target task into the prompt. More details about the experiments and results can be  
 555 found in Section 4.1. See the GPT prompts we used in Appendix B.

556    *3.3.4 Working Solution Drafting.* Given a task that the user decides to work on, JumpStarter helps users draft working  
 557 solutions, which contains the following parts.

573 *Context Selection for Working Solution Generation.* JumpStarter starts by selecting the relevant context for the given  
574 task. Specifically, given the title and detailed descriptions of the task, we prompt GPT-4 to select the most relevant  
575 context from the current local context collection. The selected context keys are presented as a checklist in a pop-up  
576 window in the system. Users are allowed to de-select any context key if necessary. Users can also add any context they  
577 deem useful from a drop-down list. We adopt the CoT prompting method to achieve this feature. See the prompt we  
578 used in Appendix A.3.1.  
579

580  
581 *Working Solution Generation.* The selected context is then applied to generate the working solution draft. We directly  
582 prompt GPT-4 to help with the task in a zero-shot manner by providing the task title, its descriptions and selected  
583 context as the input (along with the user’s goal and the global context in the system prompt). See the prompt we used  
584 in Appendix A.7.  
585

586  
587 *Context Elicitation for Working Solution Iteration.* As mentioned in Section 3.1, users can click on the “Add context  
588 and regenerate” button if they attempt to iterate on a working solution draft but are not sure how. The system will then  
589 perform context elicitation for working solution iteration for the user. Similar to context elicitation for goal (see Section  
590 3.3.1), the system provides questions to elicit more context from users, hopefully enabling them to get a better draft. To  
591 achieve this, we prompt GPT-4 in the CoT manner. See the prompt we used in Appendix A.2. With the newly elicited  
592 context, the system then triggers the “working solution generation” feature to generate an iterated draft for users.  
593  
594

### 595 3.3.5 Forking Detection and Context Selection.

596  
597 *Forking Detection.* Given a task that requires further breakdown (as determined by the “Subtask Detection” feature),  
598 JumpStarter will then assess whether it requires forking, i.e., whether it needs to be decomposed based on different  
599 entities. Unlike the typical task breakdown that treats subtasks as sequential, forking focuses on decomposing subtasks  
600 in a way that allows them to be completed in parallel, independent of the order in which they are generated. If it is  
601 not forking, JumpStarter directly goes into a typical task breakdown without any changes in the UI. If it is forking,  
602 JumpStarter then triggers the context selection for forking feature (see the paragraph below). We prompt GPT-4 in a  
603 CoT fashion with four in-context examples. See the GPT prompts we used in Appendix A.6.  
604  
605

606 *Context Selection for Forking.* Given a task that requires forking (as determined by the “Forking Detection” feature),  
607 we instruct GPT-4 to select the most relevant context keys for forking. See the prompt we used in Appendix A.3.2.  
608  
609

## 610 4 Technical Evaluation

611 JumpStarter helps people create a hierarchical action plan and draft working solutions by detecting subtasks and  
612 curating personal context. In this section, we describe how we evaluated the accuracy of subtask detection and the  
613 impact of context curation on generated plans and working solutions.  
614  
615

### 616 4.1 Subtask Detection Experiments

617 *4.1.1 Experiment settings.* Subtask detection involves determining whether a task needs further decomposition to  
618 become actionable. With vanilla zero-shot and few-shot prompting as our baselines, we experimented with three  
619 methods for detecting subtasks with GPT-4 (see Appendix B for detailed prompts). We first experimented with few-shot  
620 prompting in the Chain-of-Thought (“CoT”) style [52]. We then integrated the tree levels of the task nodes into the  
621 prompt, hypothesizing that nodes at shallower levels (e.g., the first level) would be less likely to be actionable than  
622 Manuscript submitted to ACM  
623  
624

Prompting techniques	Accuracy		Statistics	
	Mean	SD	p	Sig.
Zero-shot	.35	.000		
Few-shot	.58	.040		
+ CoT	.62	.050	.405	-
+ CoT + Tree	.69	.020	.004	**
+ CoT + Draft	.72	.020	.009	**
+ CoT + Tree + Draft	.87	.040	.000	***

Table 1. The technical evaluation results for Subtask Detection comparing different prompting techniques, where the p-values (−:  $p > .100$ , +:  $.050 < p < .100$ , \*:  $p < .050$ , \*\*:  $p < .010$ , \*\*\*:  $p < .001$ ) are reported. Note that the p-values are computed against the few-shot-only baseline. Few-shot combined with CoT+Tree+Draft achieved the best accuracy.

those at deeper levels (“Tree”). We also explored incorporating the initial solution draft into the prompt to encourage the system to suggest task decomposition depending on the quality of the draft (“Draft”). We evaluated the accuracy of these methods on an expert-labeled test suite. In all experiments, the temperature was set to 1, the max tokens were set to 2048, and *top-P* was set to 1. We report the average across five runs for each experimental setting.

**4.1.2 Test suite construction.** Drawing inspiration from [36], we created a test suite comprising four real-world task scenarios, each representing a distinct aspect of everyday life: 1) Applying for a PhD program (Academic), 2) Obtaining a driver’s license (Practical), 3) Finding a surfing camp (Recreational), and 4) Arranging a trip abroad (Travel). For each scenario, we created five distinct test cases, resulting in a total of 20 test cases. An example test case is shown below:

*Compile a List of Potential Universities: start by identifying the universities that offer PhD programs in Natural Language Processing (NLP). Research and compile a comprehensive list of these universities.*

We recruited four expert participants—one for each task scenario—via a university mailing list (average age=26.3, two female, two male). These participants reported having completed the tasks in the past six months and were compensated \$10 for their participation. For each test case, the experts were asked to determine *if the current task should be further decomposed to make it actionable*. For the example test case shown above, its label is *No* as it is actionable and does not require further task decomposition.

**4.1.3 Results and Findings.** As summarized in Table 1, the zero-shot method yielded the lowest accuracy of only 0.35, which also implies an inherent difficulty in the task itself. We observe enhanced performance with few-shot prompting (0.58), with accuracy rising even more when using the CoT prompting paradigm (0.62). Combining CoT few-shot prompting with the task node tree levels (0.69), initial solution draft (0.72), or both (0.87), all significantly led to enhanced performance. Generating the initial solution draft introduced a trade-off between latency and accuracy, so we opted for the slightly less performant few-shot CoT with tree levels for a better user experience.

## 4.2 Comparative Study with and without Context Curation

To evaluate how context curation affects the quality of JumpStarter’s proposed plans and working solutions, we conducted a controlled within-subjects lab study for the following conditions:

- (1) *Context dumping*: context saving only.
- (2) *Context filtering*: context saving and context selection.

Participant	Personal goal	Condition order
E1	Apply to a fellowship	(1) → (2) → (3)
E2		(2) → (3) → (1)
E3	Get a driver's license	(3) → (1) → (2)
E4		(1) → (2) → (3)
E5	Organize a team event	(2) → (3) → (1)
E6		(3) → (1) → (2)

Table 2. Overview of expert participants for the comparative study. Six experts were assigned one of three goals to evaluate under all three conditions, which were presented in shuffled order to avoid biasing the results.

	Subtask quality	Working solution quality
(2) vs. (1)	$p = .436$ -	$p = .001$ **
(3) vs. (1)	$p < .001$ ***	$p < .001$ ***
(3) vs. (2)	$p < .001$ ***	$p < .001$ ***

Table 3. The statistical test results comparing three conditions, where the p-values (-:  $p > .100$ , +:  $.050 < p < .100$ , \*:  $p < .050$ , \*\*:  $p < .010$ , \*\*\*:  $p < .001$ ) are reported. Our full context curation method (3) outperformed context saving alone (1) and context saving and selection (2).

### (3) Context curation: context saving, context selection, and context elicitation.

Besides ablating the use of both context selection and context elicitation in Condition (1) and context elicitation alone in Condition (2), all features and UI were kept exactly the same. We hypothesized that action plans and working solutions developed with the full context curation method, represented by Condition (3), would be judged as higher quality than those created without it.

**4.2.1 Participants and Procedure.** We evaluated the effect of context curation on subtask and working solution quality for three preselected personal goals inspired by [36]: (a) Apply to a fellowship, (b) Get a driver's license, and (c) Organize a team event. We then used a university mailing list and word of mouth to recruit participants. Each goal was assigned two “expert participants”—those who reported achieving the goal in the past six months. Overall, we recruited six expert participants for this study (average age=25.8, three female, three male). Participants were compensated \$20 per hour, with sessions lasting about 1.5 hours each.

During the study, we first introduced the expert participants to the concept of action plans and working solutions with examples. We instructed the participants that they would be using three versions of the same system for their respective goal, from start to finish. The system versions represented the three experimental conditions, which were presented in shuffled order among participants to counterbalance the learning effect (see Table 2). We demonstrated how to use each version before each participant used it. We asked the participants to generate the subtasks and working solutions exactly once. They then rated the perceived quality of the subtasks and working solutions on a seven-point Likert scale, providing a verbal explanation for their choices.

**4.2.2 Results and Findings.** Overall, 46 subtasks, or plan items, were generated for Condition 1 (*context dumping*: context saving only), 50 for Condition 2 (*context filtering*: context saving and context selection), and 50 for Condition 3 (*context curation*: context saving, context selection and context elicitation). An equal number of corresponding working solution drafts were also produced, as we asked participants to generate a draft only once for each subtask. Our method

of *context curation*, represented by Condition (3), performed the best for both subtask quality and working solution quality. For subtask quality, *context curation* ( $\mu = 6.12$ ,  $\sigma = 0.95$ ) was rated higher than the similarly performing *context dumping* ( $\mu = 5.26$ ,  $\sigma = 1.26$ ) and *context filtering* ( $\mu = 5.28$ ,  $\sigma = 1.25$ ) methods. For the quality of working solutions, *context curation* ( $\mu = 6.36$ ,  $\sigma = 0.95$ ) outperformed *context filtering* ( $\mu = 5.68$ ,  $\sigma = 1.01$ ) by a similar margin, which also outperformed *context dumping* ( $\mu = 5.04$ ,  $\sigma = 1.04$ ). These results and corresponding p-values are reported in Table 3.

Our results show that the context selection feature significantly enhances the quality of working solutions compared to the baseline. Unlike context saving only, which keeps all context in the context window all the time, context selection requires the LLM to explicitly choose relevant context from the available pool. During sessions under this condition, participants often remarked that the generated solutions appeared to take into account what they had input in previous subtasks. This was particularly evident in “summarizing” tasks, where the system could provide a personalized checklist for tasks like applying for scholarships and driver’s licenses, or an overall itinerary for a team event. E5 referred to the event itinerary they got as a “*very useful synthesis of everything I’ve explored*.” In contrast, the context saving only condition tended to produce only general tips for creating the itinerary. As another example, E1 mentioned that the email draft generated with context selection was more personalized than that created without it. E1 stated, “*I like the recommendation letter request email draft it gives me, as it considers much of my background that I saved in the previous ‘update your CV’ task. I did the same thing in the previous round [context dumping] but did not feel it was as effective.*”

In addition to context selection, context elicitation improves the quality of both subtasks and working solutions, outperforming both *context dumping* and *context filtering* conditions. Participants reported that the elicitation questions posed at the beginning “*provided the right plan to start with.*” (E4) For instance, E2 uploaded the fellowship requirements document as initially suggested by the system, later rating the generated subtasks a perfect 7/7. “*It captured the requirements quite accurately...*” E2 noted. “*The subtasks were precise, fitting the unique aspects of the fellowship I am applying to, which requires only one recommendation letter, though typically more are needed.*” Similarly, elicitation questions about which state to obtain a driver’s license in (for the “get a driver license” goal) and how many people are in the team (for the “organize a team event” goal) both resulted in subtasks that were better tailored to participants’ individual situations. In addition, preference elicitation questions such as “What type of vehicle do you intend to drive?” and “When do you prefer to hold the event? Weekday or weekend? Noon or night?” prompted participants to provide answers as personal context, ultimately resulting in more tailored working solutions that they rated highly.

## 5 User Study

To understand how JumpStarter might assist users in getting started on their personal goals, we conducted a within-subjects study with ten participants, comparing JumpStarter to ChatGPT. We chose ChatGPT as the baseline because it is a popular AI tool that many people are familiar with, and it allows users to feasibly add contextual information throughout the chat session.

During the study, each participant was asked to explore one personal goal they would like to achieve in the near future using each system. After finishing the exploration with each system, participants were asked to evaluate the system through questionnaires on task load, outcome satisfaction, and confidence level in taking the next steps on their goal. We also conducted semi-structured interviews to understand their experiences with each system. At the end of each study, we conducted another interview to get feedback about their overall experience and system preferences.

### 5.1 Hypotheses

In the study, we investigate the following hypotheses:

Participant	Personal goal	Goal type
P1	Start a side job	Career
P2	Organize a weekly game night	Life
P3	Land a job offer	Career
P4	Prepare for the LSAT	Academia
P5	Manage social media accounts	Creativity
P6	Move to a new apartment	Life
P7	Create a portfolio website	Creativity
P8	Prepare to deliver a tutorial	Academia
P9	Start a YouTube channel	Creativity
P10	Organize a family reunion	Life

Table 4. Overview of personal goals picked by participants in the user study.

- H1 Compared to ChatGPT, JumpStarter significantly **lowers users' task load (H1)** for mental demand (H1a), temporal demand (H1b), performance (H1c), effort (H1d) and frustration (H1e).
- H2 Compared to ChatGPT, JumpStarter significantly **increases users' exploration efficiency (H2)** in terms of the number of plan items explored (H2a) and the number of working solutions drafted (H2b) within the given time.
- H3 Compared to ChatGPT, JumpStarter significantly **increases users' satisfaction level with the quality of results (H3)** in terms of plan quality (H3a) and working solution quality (H3b).
- H4 Compared to ChatGPT, JumpStarter significantly **increases users' confidence in taking the next steps on their personal projects (H4)**.

## 5.2 Participants and Procedure

We recruited ten participants (average age=23.8, six female, four male) through a university mailing list and word of mouth. Each participant reported that they are familiar or very familiar with the use of ChatGPT. Before the study, we asked each participant to pick one personal goal they wanted to achieve in the next six months. The personal goals that each participant picked can be found in Table 4.

During the session, we first introduced participants to the concept of action plans and working solutions with examples. Then, each participant was asked to create an action plan for their goal and draft working solutions for as many plan items as possible, using ChatGPT and JumpStarter. Participants were randomly assigned to a condition order (ChatGPT first and then JumpStarter, or JumpStarter first and then ChatGPT) that was counterbalanced to prevent a learning effect. Before participants started using JumpStarter, we used an example case to guide them through the system and familiarize them with the interface. Then, participants were given a maximum of 25 minutes to complete each task. They also had access to a Google Doc to note their results if needed. The entire process took approximately 1.5 hours. Participants were compensated \$20/hour.

## 5.3 Results and Findings

We collected participants' ratings on a 1-7 point scale through questionnaires (see Table 5 for results). Participants rated their task load, outcome satisfaction, and confidence level in taking the next steps on their goal by using JumpStarter and ChatGPT. We recorded the number of plan items explored and the working solution drafted by each participant in each task. During the interviews, we asked them follow-up questions to understand the reasons behind their scores. We applied the thematic analysis method [7] to analyze the interview transcripts. We report the key findings in this section.

Manuscript submitted to ACM

Category	Factor	JumpStarter		ChatGPT		Statistics		Hypotheses	
		Mean	SD	Mean	SD	p	Sig.		
<b>Task load</b>	Mental demand	3.3	1.64	5.3	1.06	.005	**	<i>H1a</i> accepted	
	Temporal demand	2.4	.97	4.7	1.83	.012	*	<i>H1b</i> accepted	
	Performance	2.7	.95	4.6	1.26	.007	**	<i>H1c</i> accepted	
	Effort	2.8	1.69	4.8	1.23	.007	**	<i>H1d</i> accepted	
	Frustration	2.1	1.60	4.5	1.90	.009	**	<i>H1e</i> accepted	
<b>Exploration efficiency</b>	Plan items explored	4.5	1.43	2.9	.74	.002	**	<i>H2a</i> accepted	
	Working solutions drafted	5.3	1.89	3.3	1.16	.047	*	<i>H2b</i> accepted	
<b>Satisfaction</b>	Plan quality	5.9	.88	3.3	1.49	.004	**	<i>H3a</i> accepted	
	Working solution quality	5.8	.92	4.7	1.25	.017	*	<i>H3b</i> accepted	
<b>Confidence</b>		5.6	1.07	3.9	.99	.007	**	<i>H4</i> accepted	

Table 5. The statistical test results comparing JumpStarter with ChatGPT, where the p-values (−:  $p > .100$ , +:  $.050 < p < .100$ , \*:  $p < .050$ , \*\*:  $p < .010$ , \*\*\*:  $p < .001$ ) are reported.

**5.3.1 Task load.** In the NASA TLX dimensions [23], working with JumpStarter was significantly less demanding in terms of mental demand ( $p=.005$ ,  $Z=-2.52$ ), temporal demand ( $p=.012$ ,  $Z=-2.26$ ), performance ( $p=.007$ ,  $Z=-2.47$ ), effort ( $p=.007$ ,  $Z=-2.45$ ), and frustration ( $p=.009$ ,  $Z=-2.34$ ).

JumpStarter reduces the mental load by providing an easier way for users to visualize the structure and track progress, unlike ChatGPT's linear chat interface. For example, P2 commented, “*ChatGPT info dumps a lot, and I have to keep the structure in my brain, whereas JumpStarter gave me a structure that I could easily follow.*” Additionally, JumpStarter eases users' mental load by posing targeted questions to help them specify their context. For instance, P10 noted, “*I appreciate the questions the system (JumpStarter) asked when I felt stuck about how to iterate on drafting events [for a family reunion]. Entering the ages of the family members actually provided many better choices.*” In contrast, in ChatGPT, users must devise the context themselves and frame the prompts accordingly, as P8 said, “*In ChatGPT, the information load is high—I have to think very hard about what information I should provide in order to get things that work for me.*”

Moreover, participants reported putting significantly more effort into the task while using ChatGPT, although this did not necessarily result in better performance. For example, feeling that the current response was too general and niche, P1 asked ChatGPT to generate questions they could answer to improve the response. ChatGPT then provided eight questions, which were, according to P1, “*hard to answer as there were many and they were a bit too abstract.*” However, even after P1 made lots of effort to answer all eight questions, they still felt that the responses were “*a bit too general and not useful.*” Additionally, P1 noted that “*ChatGPT seemed to forget these eight answers soon after in my following chat with it,*” leaving them feeling that their efforts were not valued as ChatGPT failed to maintain and use this context.

**5.3.2 Exploration efficiency.** During the study session, participants explored significantly more plan items using JumpStarter (mean=4.5, SD=1.43) than they did with ChatGPT (mean=2.9, SD=0.74), and they drafted significantly more working solutions with JumpStarter (mean=5.3, SD=1.89) compared to ChatGPT (mean=3.3, SD=1.16).

In ChatGPT, it can be easy for users to become fixated on a single plan item, whereas JumpStarter helps users maintain an overview of the entire plan. For example, P2 went through nine iterations of the survey questions for organizing the weekly game night, stating, “*I got the survey questions I like in the end [after using ChatGPT], but it took me nine iterations on my own. I really hoped ChatGPT would guide me to the right place, but I had to direct myself. Because*

885 *I got too involved in this survey thing, I totally forgot I needed to work on other planning items.*” Similarly, P8 spent all the  
 886 time digging into details like designing the visual content for tutorial slides. They reflected “[With ChatGPT,] I went into  
 887 a detailed level that was not necessary very quickly, and I just forgot to do the general planning.”  
 888

889 Furthermore, JumpStarter’s structure enables users to jump between tasks and focus on one manageable task at a  
 890 time, making it easier for them to arrive at a working solution. P4 summarized it well: “I feel JumpStarter has a more  
 891 flexible structure compared to ChatGPT. I like that I can easily jump between tasks and pick the ones I care about the most.  
 892 Also, I like the visualization for each task—seeing the title, description, and context provides everything I need and helps me  
 893 focus on exploring that task.” P10 echoed this sentiment, commenting, “JumpStarter automatically manages and considers  
 894 my drafts of previous tasks, which is good because then I can focus on the current task.”  
 895

896  
 897 5.3.3 *Perceived quality of outcome.* Plans created with JumpStarter were perceived to be of significantly higher quality  
 898 (mean=5.9, SD=0.88) compared to those created with ChatGPT (mean=3.3, SD=1.49). Moreover, the working solution  
 899 quality was also perceived to be significantly higher with JumpStarter (mean=5.8, SD=0.92) compared to ChatGPT  
 900 (mean=4.7, SD=1.25), based on participants’ self-reported results.  
 901

902 JumpStarter often poses important questions to clarify users’ needs and incorporates that information into its  
 903 planning. While ChatGPT tends to provide more general plans that might not align with specific user situations. For  
 904 example, P9, who wanted to start their YouTube channel, shared, “Identifying the target audience was not listed while I  
 905 was using ChatGPT—I had to ask about it separately. Meanwhile, in JumpStarter, I was asked at the very beginning if I  
 906 already knew my target audience. I said I needed help, and then it listed identifying the target audience as the first step in  
 907 my plan, which was nice.” In addition, P9 mentioned that the plan generated with ChatGPT focused heavily on video  
 908 editing, but as a video editing expert, they did not require much preparation for that task. Instead, “JumpStarter asked  
 909 about my familiarity with video editing at the very beginning, and I told them I was an expert. I feel that this was reflected  
 910 well in the plan—like it assigned a shorter duration for the video editing task, which better suited my situation.”  
 911

912  
 913 5.3.4 *Confidence level in taking the next steps on the goal.* Participants reported significantly higher confidence in  
 914 taking the next steps on their personal projects using JumpStarter (mean=5.6, SD=1.07) compared to using ChatGPT  
 915 (mean=3.9, SD=.99).  
 916

917 Participants reported that ChatGPT helped validate their thoughts with commonsense knowledge and sometimes  
 918 provided surprising or useful tips. As P7 said, “Sitting down and planning things out itself is very helpful. I used ChatGPT  
 919 as a cross-reference, checking to make sure I’m on the right track—thinking similarly to other people. And sometimes answers  
 920 to low-level tasks covered things I did not really know, which is good.” However, they reported that they still always felt  
 921 they might miss something important while using ChatGPT, whereas with JumpStarter, they feel more secure (P3, P4,  
 922 P9), as P4 commented, “I love that I can break things down further if I want, so I don’t feel like I miss anything.”  
 923

924 In addition, JumpStarter can provide very personalized and actionable next steps that greatly increase users’ confidence  
 925 in taking action on their projects. For example, P1, with the goal of starting a side job, shared that “The schedule  
 926 JumpStarter helped me generate is very personalized, and I can directly use it to take real action—before, I felt worried about  
 927 launching this idea as I had very limited time, now I feel like I can really start doing it.” P10 also liked that JumpStarter  
 928 provided them with a specific and personalized itinerary for organizing the family reunion—“I like that it summarizes  
 929 everything I saved in the previous tasks—I can use it in the real world.”  
 930

931  
 932 5.3.5 *Tool preference.* 8 out of 10 participants reported they prefer to use JumpStarter in the future, compared to  
 933 ChatGPT. The main reasons given include that JumpStarter can provide more customized responses with less cognitive  
 934 Manuscript submitted to ACM  
 935  
 936

937 load. The users feel that they do not have to think hard about what information to provide (P1), are guided by the  
938 system (P2), can more easily consume the information (P3) or track the plan (P6), and can get their personal details  
939 efficiently organized, framed, and utilized (P8, P9, P10).

940 The other two participants (P4 and P5) mentioned that their choice depended on how familiar they were with the  
941 project they wanted to work on. If it was a topic they already had a clear understanding of, they preferred the chatbot  
942 interaction to help them figure out the details. Otherwise, they would prefer to use JumpStarter as it offers more  
943 structure. We further our discussion in terms of choices between structure and chat in Section 6.2.  
944

945 5.3.6 *Improvement feedback.* Participants also provided insights on how to improve JumpStarter. The main feedback  
946 included “make the subtask outline and task descriptions editable” (P1, P3, P6, P7, P8, P9), “format the suggested working  
947 solution to be easier to read” (P4, P8, P9), “enable users to add or edit context whenever they want” (P3, P7), and “add a  
948 synthesis button to summarize what has been explored so far” (P1, P2). P8 also suggested embedding a search agent to  
949 collect data and ensure credibility. We discuss limitations and future work further in Section 6.3.  
950

## 951 6 Discussion

### 952 6.1 Adpative Personal Context Curation in Complex Problem Solving

953 JumpStarter’s use of adaptive personal context curation in helping users explore complex goals provides a design  
954 paradigm for future interfaces. Systems that effectively respond to personal context can better collaborate with users in  
955 achieving their goals, as demonstrated in our user studies. In JumpStarter, context curation involves not only saving  
956 necessary information but also incorporates two key components: context elicitation and context selection. We found  
957 that soliciting and integrating relevant context is essential for devising personalized plans and solutions of high quality.  
958

959 Users greatly appreciated the elicitation questions posted by the system. Compared to the “pull” mode in ChatGPT,  
960 where users had to identify the requirements and constraints themselves, they tend to prefer the “push” mode in  
961 JumpStarter, where they can review the posted questions and select which ones to answer. This preference aligns with  
962 cognitive psychology literature, suggesting that recalling information is more cognitively demanding than recognizing  
963 it [12]. Additionally, when users invest more effort into prompting, as observed in ChatGPT, they are more likely to be  
964 critical of the responses they receive. This exemplifies the expectation confirmation theory [42], which suggests that  
965 dissatisfaction arises when results do not meet the standards anticipated based on the effort invested.  
966

967 Users also appreciated the system’s assistance in selecting and previewing context before drafting a solution, giving  
968 them a good sense of transparency and control. However, a few participants expressed a desire for the system to  
969 remember and use all the context they had provided. Tools like ChatGPT are designed to maintain and utilize a memory  
970 of all user-provided context throughout the chat session [45]. However, our observations reveal that ChatGPT did not  
971 effectively use the provided context to deliver more personalized responses, and sometimes even provided conflicting  
972 responses to the user. This presents an interesting challenge [51]—as users progress in their problem-solving journey,  
973 their answers to some context, such as the amount of time they want to allocate each week for LSAT preparation, might  
974 change as they realize the extent of content they need to cover. In such cases, it is crucial for the system to update  
975 the user preferences in the context pool. We aim to explore ways to refine context selection and integration from an  
976 existing context pool in the future.  
977

978 Adaptive personal context curation advances the vision of flexible and personalized systems that actively respond to  
979 user intent. Currently, in JumpStarter, all context curation occurs within the system—users must manually upload files  
980 or input information rather than integrating content from existing tools like email inboxes and desktop files. If more  
981

989 fully integrated into users' daily workflows, JumpStarter could solicit and integrate more relevant context, providing  
 990 even more actionable and useful support to users. In line with earlier explorations [28, 31], it would be ideal if the  
 991 system could perform "in-situ" context capturing, such as an add-on that integrates into users' existing workflows, as  
 992 suggested by the theory of invisible technology [5]. Such an add-on could directly place the email drafts it suggests into  
 993 the email application as editable drafts, ready to be refined and sent. Ultimately, JumpStarter's use of adaptive personal  
 994 context curation provides a design paradigm for future systems, where human insight and computational assistance  
 995 can be more effectively blended together to achieve goals.  
 996

## 998 6.2 Blending Chat and Structure for Interacting with LLMs

1000 Since the advent of ChatGPT, chatbots have become a mainstream method for interacting with LLMs. However, our  
 1001 user study indicates that a structured approach is beneficial for handling more complex tasks. This is in line with  
 1002 findings from recent HCI work [39], which shows that structure aids in generating structured thoughts and personalized  
 1003 responses. Furthermore, a structured format helps users to more easily understand the information provided by GPT,  
 1004 avoiding the overwhelming experience of endless scrolling, as suggested in recent studies [25, 48]. The subtask tree  
 1005 structure in our system additionally aids users in grasping the overall scope of tasks, tracking their progress, managing  
 1006 context, and focusing on one individual task at a time.  
 1007

1008 However, does this mean that we should always prioritize structure over chat when using LLMs for complex problem-  
 1009 solving? During our studies with JumpStarter, we observed that users show an interest in engaging with a chatbot  
 1010 at certain stages of their exploration journey. For example, when users encounter unfamiliar terms such as "research  
 1011 proposal" or "back-end coding" in the plan or working solutions, they prefer to consult a context-aware chatbot to  
 1012 understand these terms. They find the chatbot feels more natural and allows them to quickly ask follow-up questions if  
 1013 anything is unclear. Similarly, when users have strong pre-existing opinions about their plans, they tend to favor the  
 1014 chatbot for its interactivity and immediacy. This allows them to quickly receive feedback that aligns with their own  
 1015 thinking flow. For instance, if they have concerns about aspects such as the market size for their side job product ideas,  
 1016 they can immediately drill into that with the chatbot's assistance.  
 1017

1018 In addition, our system currently uses a "form fill-in" method for users to answer the elicitation questions. Although  
 1019 this structured approach helps the system efficiently gather and save the context, implementing it in a chat interaction  
 1020 might be more natural and friendly. One everyday metaphor is when a doctor wants to learn more about your health  
 1021 condition, they can either give you a long form to fill in or ask in a conversational way that might make you feel more  
 1022 comfortable. They can follow up immediately if you find some questions unclear or if the doctor finds your answers  
 1023 not making sense. However, structured interaction has many benefits that chatbot interaction does not provide, as  
 1024 we discussed in our study findings. It is an interesting future direction to explore how we might blend structure and  
 1025 interaction together to complement their pros and cons and provide a better way to interact with LLMs.  
 1026

## 1027 6.3 Limitations and Future Work

1028 JumpStarter primarily assists users in figuring out "how" to achieve their personal goals, prompting real-world actions  
 1029 by increasing simplicity, as suggested by the Fogg Behavior Model [18]. However, the "why" problem—motivation (such  
 1030 as self-regulation or emotional challenges)—is another very important dimension that JumpStarter does not address. For  
 1031 example, one participant shared that their confidence dropped after exploring the goal, as they realized how much they  
 1032 had to do to make it happen. Emotional support at this time would be very valuable. But as prior studies suggested [6],  
 1033 we should be very careful about using LLMs to provide emotional support, which deserves further investigation.  
 1034

1041 JumpStarter was designed to support people with personal goals that primarily involve cognitive or knowledge work,  
 1042 encompassing a wide range of goals across academia, career, creativity, and life (see examples in Table 4). However, there  
 1043 are many other personal goals that JumpStarter does not cover, such as physical goals (e.g., losing weight), behavioral  
 1044 goals (e.g., overcoming shyness), and spiritual goals (e.g., coming to terms with one’s faith), as outlined in [36]. Many  
 1045 of these goals require motivational support, a topic explored in previous HCI works [11, 17]. Future iterations of  
 1046 JumpStarter could potentially expand to include these areas to accommodate a broader spectrum of personal goals.  
 1047

1048 JumpStarter utilizes GPT-4 as its core engine for providing information. While LLMs like GPT-4 are adept at  
 1049 synthesizing information from the Internet and can sometimes offer valid and useful references, they are also prone to  
 1050 generating inaccurate or hallucinated information [1]. Hence, it is crucial to integrate search agents into our system to  
 1051 enhance the credibility of the information provided. Also, when pursuing creative goals such as starting a YouTube  
 1052 channel, users require support in collecting and analyzing real-world data. Developing a search agent specifically  
 1053 tailored to platforms like YouTube could be a valuable direction to explore.  
 1054

1055 JumpStarter supports users from planning to initiating their projects by creating actionable prototypes. This helps  
 1056 users grasp the full scope of their tasks and make tangible progress. While JumpStarter does not perform tasks like  
 1057 sending emails, it prepares and organizes essential information for action. This initial drafting stage is crucial for  
 1058 overcoming the planning fallacy, where users often underestimate the complexity and time required for tasks, leading to  
 1059 execution difficulties [2]. Moving forward, our focus will be on assisting users throughout their project’s lifespan. Future  
 1060 research will explore strategies for maintaining momentum and adapting to changes during real-world execution.  
 1061

1062 JumpStarter currently requires users to manually add context to the system. As discussed in Section 6.1, adopting a  
 1063 richer and more flexible method for capturing context that integrates into users’ current workflows would be beneficial.  
 1064 Additionally, JumpStarter currently can only handle context in text form, but many context are not in text, such as  
 1065 YouTube channel icons or slide visual layouts. As visual understanding models like GPT-4V [1] evolve, it would be  
 1066 interesting to expand JumpStarter to handle multimodal context.  
 1067

## 1068 7 Conclusion

1069 In this work, we present JumpStarter, a system designed to support people in getting started with their personal goals  
 1070 through adaptive personal context curation. Our technical evaluation indicates that context curation plays a crucial  
 1071 role in generating plans and working solutions of higher quality. Our user study reveals that, compared to ChatGPT,  
 1072 JumpStarter enables users to explore their goals more efficiently and with less mental effort. We discuss the design  
 1073 implications of incorporating adaptive personal context curation into complex problem-solving. We also provide insights  
 1074 into how we might blend structured and conversational approaches to better support people’s interaction with LLMs.  
 1075

## 1076 8 References

- 1077 [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- 1078 [2] Yoana Ahmetoglu, Duncan P Brumby, and Anna L Cox. 2021. To plan or not to plan? A mixed-methods diary study examining when, how and why knowledge work planning is inaccurate. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (2021), 1–20.
- 1079 [3] Icek Ajzen. 1991. The Theory of planned behavior. *Organizational Behavior and Human Decision Processes* (1991).
- 1080 [4] Alan Baddeley. 1992. Working memory. *Science* 255, 5044 (1992), 556–559.
- 1081 [5] Eric Bergman and Donald A Norman. 2000. Making Technology Invisible. *Information Appliances and Beyond: Interaction Design for Consumer Products* (2000).
- 1082 [6] Ananya Bhattacharjee, Yuchen Zeng, Sarah Yi Xu, Dana Kulzhabayeva, Minyi Ma, Rachel Kornfield, Syed Ishtiaque Ahmed, Alex Mariakakis, Mary P Czerwinski, Anastasia Kuzminikh, et al. 2024. Understanding the Role of Large Language Models in Personalizing and Scaffolding Strategies to Combat Academic Procrastination. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–18.

- 1093 [7] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
- 1094 [8] Carrie J. Cai, Philip J. Guo, James Glass, and Robert C. Miller. 2014. Wait-learning: leveraging conversational dead time for second language  
1095 education. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (*CHI EA '14*). Association for Computing  
1096 Machinery, New York, NY, USA, 2239–2244. <https://doi.org/10.1145/2559206.2581183>
- 1097 [9] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional  
1098 interpolation. *arXiv preprint arXiv:2306.15595* (2023).
- 1099 [10] Kenneth Conley and James Carpenter. 2007. Towel: Towards an Intelligent To-Do List. In *AAAI Spring Symposium: Interaction Challenges for  
1100 Intelligent Assistants*. 26–32.
- 1101 [11] Sunny Consolvo, Predrag Klasnja, David W McDonald, and James A Landay. 2009. Goal-setting considerations for persuasive technologies that  
1102 encourage physical activity. In *Proceedings of the 4th international Conference on Persuasive Technology*. 1–8.
- 1103 [12] Fergus IM Craik and Robert S Lockhart. 1972. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal  
1104 behavior* 11, 6 (1972), 671–684.
- 1105 [13] Allen Cypher. 1986. The structure of users' activities. *User centered system design* (1986), 243–263.
- 1106 [14] Edward L Deci and Richard M Ryan. 2012. Self-determination theory. *Handbook of theories of social psychology* 1, 20 (2012), 416–436.
- 1107 [15] Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv  
1108 preprint arXiv:2305.02750* (2023).
- 1109 [16] Adele Diamond. 2013. Executive functions. *Annual review of psychology* 64, 1 (2013), 135–168.
- 1110 [17] Tina Ekhtiar, Armağan Karahanoglu, Rúben Gouveia, and Geke Ludden. 2023. Goals for goal setting: a scoping review on personal informatics. In  
1111 *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 2625–2641.
- 1112 [18] Brian J Fogg. 2009. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology*. 1–7.
- 1113 [19] Peter Gollwitzer. 1996. The volitional benefits of planning. In *The psychology of action: Linking cognition and motivation to behavior*. Guilford Press,  
1114 287–312.
- 1115 [20] Peter M Gollwitzer. 1990. Action phases and mind-sets. *Handbook of motivation and cognition: Foundations of social behavior* 2, 53–92 (1990), 2.
- 1116 [21] Peter M Gollwitzer. 1999. Implementation intentions: strong effects of simple plans. *American psychologist* 54, 7 (1999), 493.
- 1117 [22] Kunal Handa, Yarin Gal, Ellie Pavlick, Noah Goodman, Jacob Andreas, Alex Tamkin, and Belinda Z Li. 2024. Bayesian preference elicitation with  
1118 language models. *arXiv preprint arXiv:2403.05534* (2024).
- 1119 [23] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*,  
1120 Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.
- 1121 [24] Martin Heidegger. 2010. *Being and time*. SUNY press.
- 1122 [25] Peiling Jiang, Jude Rayan, Steven P Dow, and Haijun Xia. 2023. Graphologue: Exploring large language model responses with interactive diagrams.  
In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–20.
- 1123 [26] Philip Nicholas Johnson-Laird. 1983. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Number 6. Harvard  
1124 University Press.
- 1125 [27] William Jones, Predrag Klasnja, Andrea Civan, and Michael L Adcock. 2008. The personal project planner: Planning to organize personal information.  
In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 681–684.
- 1126 [28] Victor Kaptelein. 2003. UMEA: translating interaction histories into project contexts. In *Proceedings of the SIGCHI conference on Human factors in  
1127 computing systems*. 353–360.
- 1128 [29] Harmanpreet Kaur, Alex C Williams, Anne Loomis Thompson, Walter S Lasecki, Shamsi T Iqbal, and Jaime Teevan. 2018. Creating better action  
1129 plans for writing tasks via vocabulary-based planning. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–22.
- 1130 [30] Andruid Kerne, Andrew M Webb, Steven M Smith, Rhema Linder, Nic Lupfer, Yin Qu, Jon Moeller, and Sashikanth Damaruju. 2014. Using metrics of  
1131 curation to evaluate information-based ideation. *ACM Transactions on Computer-Human Interaction (ToCHI)* 21, 3 (2014), 1–48.
- 1132 [31] Mik Kersten and Gail C Murphy. 2006. Using task context to improve programmer productivity. In *Proceedings of the 14th ACM SIGSOFT international  
1133 symposium on Foundations of software engineering*. 1–11.
- 1134 [32] Nicolas Kokkalis, Thomas Köhn, Johannes Huebner, Moontae Lee, Florian Schulze, and Scott R Klemmer. 2013. Taskgenies: Automatically providing  
1135 action plans helps people complete tasks. *ACM Transactions on Computer-Human Interaction (TOCHI)* 20, 5 (2013), 1–25.
- 1136 [33] Michel Krieger, Emily Margarete Stark, and Scott R Klemmer. 2009. Coordinating tasks on the commons: designing for personal goals, expertise and  
1137 serendipity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1485–1494.
- 1138 [34] Yash Kumar Lal, Li Zhang, Faeze Brahman, Bodhisattwa Prasad Majumder, Peter Clark, and Niket Tandon. 2024. Tailoring with Targeted Precision:  
1139 Edit-Based Agents for Open-Domain Procedure Customization. In *Findings of the Association for Computational Linguistics ACL 2024*. 15597–15611.
- 1140 [35] Belinda Z Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2023. Eliciting human preferences with language models. *arXiv preprint  
1141 arXiv:2310.11589* (2023).
- 1142 [36] Brian R Little. 1983. Personal projects: A rationale and method for investigation. *Environment and behavior* 15, 3 (1983), 273–309.
- 1143 [37] Chengzhong Liu, Zeyu Huang, Dingdong Liu, Shixu Zhou, Zhenhui Peng, and Xiaojian Ma. 2022. PlanHelper: Supporting Activity Plan Construction  
1144 with Answer Posts in Community-Based QA Platforms. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–26.

- 1145 [39] Xiao Ma, Swaroop Mishra, Ariel Liu, Sophie Ying Su, Jilin Chen, Chinmay Kulkarni, Heng-Tze Cheng, Quoc Le, and Ed Chi. 2024. Beyond chatbots:  
 1146 ExploreLlm for structured thoughts and personalized model responses. In *Extended Abstracts of the CHI Conference on Human Factors in Computing  
 1147 Systems*. 1–12.
- 1148 [40] James E Maddux. 1995. Self-efficacy theory: An introduction. In *Self-efficacy, adaptation, and adjustment: Theory, research, and application*. Springer,  
 1149 3–33.
- 1150 [41] Chaitanya Malaviya, Joseph Chee Chang, Dan Roth, Mohit Iyyer, Mark Yatskar, and Kyle Lo. 2024. Contextualized Evaluations: Taking the  
 1151 Guesswork Out of Language Model Evaluations. *arXiv preprint arXiv:2411.07237* (2024).
- 1152 [42] Richard L Oliver. 1980. A cognitive model of the antecedents and consequences of satisfaction decisions. *Journal of marketing research* 17, 4 (1980),  
 1153 460–469.
- 1154 [43] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *arXiv  
 1155 preprint arXiv:2309.00071* (2023).
- 1156 [44] Sajjadur Rahman, Pao Sianglihue, and Adam Marcus. 2020. Mixtape: Mixed-initiative team action plan creation through semi-structured notes,  
 1157 automatic task generation, and task classification. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–26.
- 1158 [45] Partha Pratim Ray. 2023. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope.  
*Internet of Things and Cyber-Physical Systems* 3 (2023), 121–154.
- 1159 [46] Maria A Rickers-Ovsiankina. 1928. *Die wiederaufnahme unterbrochener handlungen*. Ph. D. Dissertation. Verlagsbuchhandlung Julius Springer.
- 1160 [47] Paschal Sheeran and Thomas L Webb. 2016. The intention–behavior gap. *Social and personality psychology compass* 10, 9 (2016), 503–518.
- 1161 [48] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling multilevel exploration and sensemaking with large language  
 1162 models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–18.
- 1163 [49] John Sweller. 2011. Cognitive load theory. In *Psychology of learning and motivation*. Vol. 55. Elsevier, 37–76.
- 1164 [50] Jaime Teevan, Shamsi T Iqbal, Carrie J Cai, Jeffrey P Bigham, Michael S Bernstein, and Elizabeth M Gerber. 2016. Productivity decomposed: Getting  
 1165 big things done with little microtasks. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*.  
 3500–3507.
- 1166 [51] Tiannan Wang, Meiling Tao, Ruoyu Fang, Huilin Wang, Shuai Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2024. AI PERSONA: Towards  
 1167 Life-long Personalization of LLMs. *arXiv preprint arXiv:2412.13103* (2024).
- 1168 [52] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting  
 1169 elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
- 1170 [53] Deena Skolnick Weisberg, Kathy Hirsh-Pasek, Roberta Michnick Golinkoff, and Bruce D McCandliss. 2014. Mise en place: Setting the stage for  
 thought and action. *Trends in Cognitive Sciences* 18, 6 (2014), 276–278.
- 1171 [54] Alex C Williams, Harmanpreet Kaur, Shamsi Iqbal, Ryen W White, Jaime Teevan, and Adam Fournier. 2019. Mercury: Empowering Programmers'  
 1172 Mobile Work Practices with Microproductivity. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 81–94.
- 1173 [55] T Winograd. 1986. Understanding computers and cognition: A new foundation for design.
- 1174 [56] Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for  
 1175 real-world planning with language agents. *arXiv preprint arXiv:2402.01622* (2024).
- 1176 [57] Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-context language modeling with parallel context encoding. *arXiv preprint arXiv:2402.16617*  
 1177 (2024).
- 1178 [58] Xuan Zhang, Yang Deng, Zifeng Ren, See-Kiong Ng, and Tat-Seng Chua. 2024. Ask-before-Plan: Proactive Language Agents for Real-World Planning.  
*arXiv preprint arXiv:2406.12639* (2024).
- 1179 [59] Xuanming Zhang, Rahul Divekar, Rutuja Ubale, and Zhou Yu. 2023. GrounDialog: A Dataset for Repair and Grounding in Task-oriented Spoken  
 1180 Dialogues for Language Learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*.  
 300–314.
- 1181 [60] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user  
 1182 respond. In *Proceedings of the 27th acm international conference on information and knowledge management*. 177–186.
- 1183 [61] Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V Le, Ed H Chi,  
 1184 et al. 2024. NATURAL PLAN: Benchmarking LLMs on Natural Language Planning. *arXiv preprint arXiv:2406.04520* (2024).
- 1185
- 1186
- 1187
- 1188
- 1189
- 1190
- 1191
- 1192
- 1193
- 1194
- 1195
- 1196

**A GPT-4 Prompts used in JumpStarter**

**A.1 Context Elicitation for the Goal**

See Figure 8.

**A.2 Context Elicitation for Drafting Working Solutions**

See Figure 9.

**A.3 Context Selection**

**A.3.1 Context Selection for Drafting.** See Figure 10.

**A.3.2 Context Selection for Forking.** The prompt is as follows:

- My user has a main purpose: {main\_purpose}. My user is working on the task {task\_name}: {task\_description}. My user needs to break down the task into sub-tasks. Here is the current context history from the user: {context\_history}. Please select the most relevant context key from the current context history that can be used to better decompose the current task into several sub-tasks for the user to get started. Do not help the user to break down the task. Please also provide explanations. Format the response like this: <context\_key>: <reasons>. Replace the context\_key with the actual key in the context history.

**A.4 Subtask Suggestion**

See Figure 11.

**A.5 Subtask Detection**

See Figure 12.

**A.6 Fork Detection**

See Figure 13.

**A.7 Working Solution Drafts**

The prompt to generate the working draft is shown as follows:

- My user has a main purpose: {main purpose}. Please consider the following context information from my user: {user\_context}. My user needs help with the current task {current task}: {task description}

**B Prompts for Technical Evaluation of Subtask Detection**

**B.1 Zero-shot Prompting**

The prompt for zero-shot for the task of subtask detection is demonstrated below:

- System prompt: You are a useful assistance to detect if the current task needs to be further decomposed if it is not actionable and the primary goal of the task can not be viewed as a singular, distinctive deliverable. Based on the user prompt, please output Yes if it needs to be decomposed; No otherwise meaning it is actionable and does not require task decomposition.

1249  
 1250     **System prompt**

1251 You are a helpful assistant in generating at most three questions to elicit more context from the  
 1252 user in order to accomplish a task the user is involved. You will be given the main purpose of the  
 1253 task, and the current context history from the user. You need to judge if the current context  
 1254 information is enough to do this task. If not, what existing doc do you think they already have  
 1255 and can be provided as the context document for this task? Alternatively, what existing most  
 1256 important extra information do you think they should provide as the context information for this  
 1257 task? Please try your best to start by asking the potential existing doc first, and then ask for the  
 1258 potential info. Do not ask the question that can be possibly answered from the suggested doc in  
 1259 the first question. You can also just ask for the potential info if no doc is needed from the user.  
 1260 Please directly generate the questions for users to answer, provide the reason for the question,  
 1261 identify if it is DOC or INFO, and provide a name for the question. Otherwise, output "Ready".  
 1262

1263 Here are two examples:

1264 Input: My user has a main purpose: Apply for a drivers license by 12/15/2024. The current  
 1265 context history from the user is empty.

1266 Output:

- 1267 1. Reason: Driver's license requirements vary significantly depending on the location. Knowing  
 1268 the specific state or country would allow for tailored advice regarding local rules, tests, and  
 1269 documentation required -> Question: Which state or country are you applying for? -> Type:  
 1270 INFO -> Name: State or Country
- 1271 2. Reason: Different age groups may have different requirements or steps in the licensing  
 1272 process. For example, minors often have to go through graduated license programs. ->  
 1273 Question: How old are you -> Type: INFO -> Name: Age of the User

1274 Input: My user has a main purpose: Apply for a PhD program by 12/15/2024. The current  
 1275 context history from the user is empty.

1276 Output:

- 1277 1. Reason: Curriculum Vitae (CV) or Resume could be helpful as it would likely contain detailed  
 1278 information about their educational background and any research experiences or academic  
 1279 achievements, which are critical for applying to PhD programs. -> Question: What CV or  
 1280 Resume can you provide for the PhD application? -> Type: DOC -> Name: CV or Resume

1281 Now, start prediction:

1282  
 1283     **User Prompt**

1284 Input: My user has a main purpose: {task\_input}. The current context history from the user is  
 1285 empty.

1286 Fig. 8. The prompt for Context Elicitation for the goal.

- 1287 • User Prompt: *My user is working on the task {task title}: {task description}. My user needs to know if the current*  
*1288 task needs to be decomposed.*

```

1301 My user has a main purpose: {main_purpose}. My user is working on the task {task_name}:
1302 {task_description}. The current context history from the user is {context_history}. Please judge if
1303 the current context information is enough to do this task. If not, what existing most important
1304 extra information do you think they should provide as the context information for this task?
1305 Please directly generate questions for users to answer as extra info. Otherwise, output "Ready"
1306 Format the response like this: 1. <question 1> : <reason for asking question 1> -> title of
1307 question 1
1308 2. <question 2> : <reason for asking question 2> -> title of question 2
1309 3. <question 3> : <reason for asking question 3> -> title of question 3
1310
1311
1312 Fig. 9. The prompt for Context Elicitation for drafting working solutions.
1313
1314
1315 System prompt
1316 Given the user's main purpose and the task they are working on, select the most relevant
1317 context keys from the current context history that can be used to draft good responses for the
1318 user to complete the task. Please also provide explanations. \n The current context history is
1319 shown as one or more key-value pairs. Please select only the keys from the 'key' part of the
1320 context history. Do not select the keys from the 'value' part of the context history \n Format the
1321 response like this: number. <context_keys>: <reasons for selecting context_keys>. Replace the
1322 context_keys with the actual keys as shown in the context history. Please directly give the
1323 answers and do not provide extra summarization sentences at the end.
1324
1325
1326
1327 User Prompt
1328 My user has a main purpose: {main_purpose}. My user is working on the task {task_name}:
1329 {task_description}. Here is the current context history in JSON format (with 'key':'value' pairs)
1330 from the user: {context_history}
1331
1332
1333
1334 Fig. 10. The prompt for Context Selection for drafting working solutions.
1335
1336
1337 B.2 Few-shot Prompting
1338 The prompt for few-shot-only prompting is shown in Figure 14. Note that we used three in-context examples in the
1339 prompt.
1340
1341
1342 B.3 Few-shot + CoT
1343 We constructed the prompt in a Chain-of-Thought fashion, where GPT-4 is instructed to first generate the reasoning
1344 and then the answer. The prompt is shown in Figure 15.
1345
1346
1347 B.4 Few-shot + CoT + Draft
1348 We experimented with incorporating both CoT and the initial working solution draft into the prompt. The system
1349 prompt is shown in Figure 16. For the user prompt, before detecting subtasks. we first generated the initial working
1350 solution draft for the current task. The user prompt is shown below:
1351
1352 Manuscript submitted to ACM

```

Fig. 10. The prompt for Context Selection for drafting working solutions.

1353 My user has a main purpose:{main purpose}.  
 1354 Please consider the following context information from my user: {user\_context}  
 1355  
 1356 Please break down the task below into three to six manageable subtasks: {current task}  
 1357 The existing step structure is shown as follows: {existing tree step structure}.  
 1358  
 1359 Please directly give the response that fills in the current subtask: {current task} in the provided  
 1360 task structure.  
 1361  
 1362 Format the response like this: 1. [Duration for subtask1] {subtask1 title}: {subtask1 detailed  
 1363 description} 2. [Duration for subtask2] {subtask2 title}: {subtask2 detailed description} 3.  
 1364 [Duration for subtask3] {subtask3 title}: {subtask3 detailed description}.  
 1365 Please specify the duration for each subtask in terms of days, weeks or months. For example,  
 1366 [1 week], [2-4 weeks], [1 month], and [1-2 months]. Please do not include other texts for  
 1367 duration such as [Ongoing]. Please do not include \*\* in the subtask title. Please directly give the  
 1368 response and do not start with "{current subtask title}":  
 1369  
 1370  
 1371  
 1372 Fig. 11. The prompt for Subtask Suggestion.

- 1373  
 1374 • *My user is working on the task {task title}: {task description}. The GPT response to the task is: {Draft}. My user needs*  
 1375 *to know if the current task is specific and actionable.*

### 1377 B.5 Few-shot + CoT + Tree + Draft

1379 To construct the system prompt for this setting, we incorporate the tree level of each task into the prompt. The prompt  
 1380 is shown in Figure 17. Additionally, for the current task at hand, its tree-level information is also presented in the user  
 1381 prompt, as shown below:

- 1382  
 1383 • *My user is working on the task {task title}: {task description}. The current node level of the task is {level}. The GPT*  
 1384 *response to the task is: {Draft}. My user needs to know if the current task is specific and actionable.*

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

## System prompt

You are a useful assistance to detect if the current task needs to be further decomposed if it is not actionable and the primary goal of the task can not be viewed as a singular, distinctive deliverable. Specifically, an overall goal is decomposed into a tree of subtasks. Each of these tasks is positioned on the tree structure and is associated with a level. Based on the user prompt, and the current level of the task node on the tree, please output Yes if it needs to be decomposed; No otherwise meaning it is actionable and does not require task decomposition. Please also provide explanations for your choice.

Here are some examples:

User: My user is working on the task Research on Prospective Companies and Positions: Conduct a comprehensive search on potential companies and specific research scientist internship positions in the NLP field. Understand what each role entails, identify skill requirements, and evaluate how they align with your research interests. The current node level of the task is 1. My user needs to know if the current task is specific and actionable

Reason: this task needs to be further decomposed as it is positioned on the first level of the tree and it involves more than one deliverables: search on companies and search on positions. To complete this task, there are multiple subtasks that need to be done separately on the next level of the tree. These include conducting a comprehensive search on potential companies, searching for specific research scientist internship positions, an analysis of what each role entails, identifying skill requirements, and evaluation of alignment with the user's research interests.

Answer: Yes

User: My user is working on the task Identify Potential Universities: Create a list of universities that offer PhD programs in HCI. The selection can be based on factors such as reputation, HCI research focus, published HCI research papers, faculty expertise etc. The current node level of the task is 2. My user needs to know if the current task is specific and actionable.

Reason: This task does not need to be further decomposed as it is positioned on the second level of the tree and it just involves one deliverable: create a list of schools that offer PhD programs in HCI. Although it may require several steps to create the list, the end goal of this task is to get a list. Therefore the task is actionable.

Answer: No

User: My user is working on the task Identify Required Documents: Research and confirm all the necessary documents required for the non-driver ID application, ensuring to list all forms of acceptable proofs such as a birth certificate or passport for identity, Social Security Card or W-2 form for Social Security number, and utility bills or lease agreement for proof of residency. The current node level of the task is 1. My user needs to know if the current task is specific and actionable.

Reason: Although the task is positioned on the first level of the tree, the task does not need to be further decomposed as the primary goal of the task - identify necessary documents for the non-driver ID application - can be viewed as a singular, cohesive deliverable. Despite involving various types of documents, the task is focused on compiling a comprehensive list, which makes it actionable as a single unit. The distinction lies in the focus on gathering all necessary documentation, a clear and direct objective.

Answer: No

Now, let's start prediction:

## User Prompt

User: My user is working on the task {task\_title}: {task\_descriptions}. The current node level of the task is {tree\_level}. My user needs to know if the current task needs to be decomposed

Fig. 12. The prompt for Subtask Detection.

**System prompt**

Given the queried task, determine if a "for" loop is needed to complete the task. You will be given a question Q. Please provide the reasoning and then respond with "Yes" or "No".

Here are some examples:

Q: Research the specific HCI PhD programs at each university from the initial list. Focus on aspects such as program curriculum, research opportunities, faculty expertise, and available resources.

Reason: The task requires a "for" loop to complete as there already exists an initial list of entities (i.e. universities) to research. Specifically, the goal of this task is to research the program curriculum, research opportunities, faculty expertise, and available resources for each university from the initial list. It is not possible to complete the task directly without a "for" loop.

A: Yes

Q: Make a list of potential recommenders including former supervisors, academic advisors, and professors who are familiar with your academic and research abilities.

Reason: The task does not require a "for" loop to complete as there does not exist a list of potential recommenders. The goal of this task therefore is to construct the list of recommenders based on certain criteria.

A: No

Q: Reach out to the individuals on your list via email or phone, providing them with the necessary documents and details about the HCI programs, and formally request their letters of recommendation

Reason: This task needs a "for" loop to complete as you have already obtained your list of individuals. You should reach out to each entity (i.e. individuals) on the list to complete the task.

A: Yes

Q: Gather information on different universities offering PhD programs in Human-Computer Interaction. Create an initial list based on general information such as program recognition, location, and basic offerings.

Reason: The task does not require a "for" loop to complete as there does not exist a list of universities offering PhD programs in HCI that can be used to iterate on.

A: No

**User Prompt**

Q: {task\_description}

Fig. 13. The prompt for Fork Detection.

1509  
1510

### System prompt

1511  
1512  
1513  
1514  
1515

You are a useful assistance to detect if the current task needs to be further decomposed if it is not actionable and the primary goal of the task can not be viewed as a singular, distinctive deliverable. Based on the user prompt, please output Yes if it needs to be decomposed; No otherwise meaning it is actionable and does not require task decomposition

1516  
1517  
1518  
1519  
1520  
1521

Here are some examples:

1522  
1523  
1524  
1525  
1526  
1527

User: My user is working on the task Research on Prospective Companies and Positions: Conduct a comprehensive search on potential companies and specific research scientist internship positions in the NLP field. Understand what each role entails, identify skill requirements, and evaluate how they align with your research interests. My user needs to know if the current task is specific and actionable

1528

Answer: Yes

1530  
1531  
1532  
1533  
1534

User: My user is working on the task Identify Potential Universities: Create a list of universities that offer PhD programs in HCI. The selection can be based on factors such as reputation, HCI research focus, published HCI research papers, faculty expertise etc. My user needs to know if the current task is specific and actionable.

1535

Answer: No

1536  
1537  
1538

User: My user is working on the task Identify Required Documents: Research and confirm all the necessary documents required for the non-driver ID application, ensuring to list all forms of acceptable proofs such as a birth certificate or passport for identity, Social Security Card or W-2 form for Social Security number, and utility bills or lease agreement for proof of residency. My user needs to know if the current task is specific and actionable.

1539  
1540  
1541

Answer: No

1542  
1543  
1544

Now, let's start prediction:

1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559

### User prompt

My user is working on the task {task title}: {task description}. My user needs to know if the current task needs to be decomposed.

Fig. 14. The few-shot-only prompt for Subtask Detection.

1560

## 1561 System prompt

1562 You are a useful assistance to detect if the current task needs to be further decomposed if it is not  
1563 actionable and the primary goal of the task can not be viewed as a singular, distinctive deliverable. Based  
1564 on the user prompt, please output Yes if it needs to be decomposed; No otherwise meaning it is  
1565 actionable and does not require task decomposition. Please also provide explanations for your choice.

1566 Here are some examples:

1567 User: My user is working on the task Research on Prospective Companies and Positions: Conduct a  
1568 comprehensive search on potential companies and specific research scientist internship positions in the  
1569 NLP field. Understand what each role entails, identify skill requirements, and evaluate how they align with  
1570 your research interests. My user needs to know if the current task is specific and actionable  
1571 Reason: this task needs to be further decomposed as it involves more than one deliverables: search on  
1572 companies and search on positions. To complete this task, there are multiple subtasks that need to be  
1573 done separately. These include conducting a comprehensive search on potential companies, searching  
1574 for specific research scientist internship positions, an analysis of what each role entails, identifying skill  
1575 requirements, and evaluation of alignment with the user's research interests.  
1576

1577 Answer: Yes

1578 User: My user is working on the task Identify Potential Universities: Create a list of universities that offer  
1579 PhD programs in HCI. The selection can be based on factors such as reputation, HCI research focus,  
1580 published HCI research papers, faculty expertise etc. My user needs to know if the current task is specific  
1581 and actionable.

1582 Reason: This task does not need to be further decomposed as it just involves one deliverable: create a  
1583 list of schools that offer PhD programs in HCI. Although it may require several steps to create the list, the  
1584 end goal of this task is to get a list. Therefore the task is actionable.  
1585

1586 Answer: No

1587 User: My user is working on the task Identify Required Documents: Research and confirm all the  
1588 necessary documents required for the non-driver ID application, ensuring to list all forms of acceptable  
1589 proofs such as a birth certificate or passport for identity, Social Security Card or W-2 form for Social  
1590 Security number, and utility bills or lease agreement for proof of residency. My user needs to know if the  
1591 current task is specific and actionable.

1592 Reason: The primary goal of the task is to identify necessary documents for the non-driver ID application,  
1593 which can be viewed as a singular, cohesive deliverable. Despite involving various types of documents,  
1594 the task is focused on compiling a comprehensive list, which makes it actionable as a single unit. The  
1595 distinction lies in the focus on gathering all necessary documentation, a clear and direct objective.  
1596

1597 Answer: No

1598 Now, let's start prediction:

## 1600 User prompt

1601 My user is working on the task {task title}: {task description}. My user needs to know if the current task  
1602 needs to be decomposed.

1603 Fig. 15. The prompt for few-shot + CoT for Subtask Detection.  
1604

**1613 System prompt**  
**1614** You are a useful assistance to detect if the current task needs to be further decomposed if it is not actionable and the primary goal of the task can not be viewed as a singular, distinctive deliverable. Based on the user prompt, and the GPT response, please output Yes if it needs to be decomposed; No otherwise meaning it is actionable and does not require task decomposition. Please also provide explanations for your choice.  
**1615** Here are some examples:  
**1616** User: My user is working on the task Research on Prospective Companies and Positions: Conduct a comprehensive search on potential companies and specific research scientist internship positions in the NLP field. Understand what each role entails, identify skill requirements, and evaluate how they align with your research interests. The GPT response to the task is:  
**1617** Sure, I will list some promising companies that offer research scientist internship positions in Natural Language Processing (NLP) and briefly describe what each role entails:  
**1618** 1. \*\*Google AI\*\*:  
**1619** - \*Role\*: As a research intern, you can work in areas such as machine intelligence, machine perception, natural language processing, and quantum computing. Google typically looks for research experience, ability to design and execute a research agenda, and coding skills.  
**1620** 2. \*\*Facebook AI Research (FAIR)\*\*:  
**1621** - \*Role\*: Interns at FAIR involve developing novel techniques for NLP tasks which include generating human-like text. They require experience with machine learning, AI, deep learning, dialogue management, and computer vision.  
**1622** 3. \*\*Amazon Alexa\*\*:  
**1623** - \*Role\*: Interns in this role generally work on language understanding capabilities of Alexa, advancing state-of-the-art on tasks like entity resolution or text classification. Knowledge in machine learning, especially deep learning for NLP, is key for this role.  
**1624** 4. \*\*OpenAI\*\*:  
**1625** - \*Role\*: Research internships at OpenAI are highly selective. Interns are expected to have published in machine learning or AI and can spend a significant portion of their time on original research.  
**1626** 5. \*\*IBM Watson\*\*:  
**1627** - \*Role\*: Watson Interns can work on various AI/NLP tasks from fact extraction, relation extraction, disambiguation, to summarization, based on the team you join. A strong AI/ML foundation and coding skills are required.  
**1628** 6. \*\*Apple AI/ML\*\*:  
**1629** - \*Role\*: Apple offers internships where you can work on Siri's language understanding capabilities. They typically require machine learning knowledge and experience, especially in deep learning.  
**1630** 7. \*\*Microsoft Research\*\*:  
**1631** - \*Role\*: Interns in this role could work on a variety of NLP challenges, including developing algorithms that enable, for example, conversational interactions with AI. Strong machine learning knowledge, particularly in deep learning, is expected.  
**1632** 8. \*\*Baidu Research\*\*:  
**1633** - \*Role\*: Interns can work on various novel NLP tasks such as language understanding, language generation, and more. Baidu looks for individuals with robust knowledge in machine learning, deep learning, and coding abilities.  
**1634** 9. \*\*NVIDIA - AI Research\*\*:  
**1635** - \*Role\*: Interns have an opportunity to work on projects in the realms of machine learning, deep learning, computer vision, and natural language processing. They seek candidates with strong ML and DL backgrounds.  
**1636** When selecting a specific position, I recommend considering the following:  
**1637** - Does the role align with your career objectives?  
**1638** - Does the role play to your strengths or offer opportunities to develop in areas of interest?  
**1639** - Does the role match your skill set? If not, can you acquire those skills?  
**1640** Remember, a role can be an opportunity to learn and grow, so don't be deterred if you lack a few skills. If the role excites you, go for it!  
**1641** My user needs to know if the current task is specific and actionable  
**1642** Reason: this task needs to be further decomposed as the GPT response is not good enough to help user complete the task. It involves more than one deliverables: search on companies and search on positions. To complete this task, there are multiple subtasks that need to be done separately. These include conducting a comprehensive search on potential companies, searching for specific research scientist internship positions, an analysis of what each role entails, identifying skill requirements, and evaluation of alignment with the user's research interests.  
**1643** Answer: Yes  
**1644** User: My user is working on the task Identify Potential Universities: Create a list of universities that offer PhD programs in HCI. The selection can be based on factors such as reputation, HCI research focus, published HCI research papers, faculty expertise etc. The GPT response to the task is:  
**1645** Here is a list of universities that offer PhD programs in HCI:  
**1646** 1. Stanford University, USA  
**1647** 2. Massachusetts Institute of Technology (MIT), USA  
**1648** 3. Carnegie Mellon University (CMU), USA  
**1649** 4. University of California, Berkeley, USA  
**1650** 5. University of Washington, USA  
**1651** 6. University of Michigan, USA  
**1652** 7. University of California, San Diego, USA  
**1653** 8. Cornell University, USA  
**1654** 9. University of Toronto, Canada  
**1655** 10. University of College London, UK  
**1656** My user needs to know if the current task is specific and actionable.  
**1657** Reason: the GPT response is good enough and it provides direct answer to complete the task. Additionally, this task does not need to be further decomposed as it just involves one deliverable: create a list of schools that offer PhD programs in HCI. Although it may require several steps to create the list, the end goal of this task is to get a list. Therefore the task is actionable.  
**1658** Answer: No  
**1659** User: My user is working on the task Identify Required Documents: Research and confirm all the necessary documents required for the non-driver ID application, ensuring to list all forms of acceptable proofs such as a birth certificate or passport for identity, Social Security Card or W-2 form for Social Security number, and utility bills or lease agreement for proof of residency. The GPT response to the task is:  
**1660** To apply for a non-driver ID in [REDACTED] you need to provide several documents to prove your identity, Social Security Number (SSN), and residency. Below, I have listed the acceptable documents for each category based on the latest guidelines provided by the [REDACTED] Department of Motor Vehicles (DMV).  
**1661** \*\*1. Proof of Identity and Date of Birth (must total 6 points according to [REDACTED] DMV's point system):\*\*  
**1662** - U.S. passport or U.S. passport card (valid or expired within the past five years) – 4 points  
**1663** - Birth certificate (U.S., Puerto Rican, Canadian, or Mexican) - 2 points  
**1664** - Employment Authorization Card (I-766) – 3 points  
**1665** - Permanent Resident Card (I-551) – 3 points  
**1666** - Foreign passport with valid U.S. Visa and I-94 – 3 points  
**1667** \*Note: At least one document must show your date of birth and at least one document must have your full Social Security Number.\*  
**1668** \*\*2. Proof of Social Security Number:\*\*  
**1669** - Social Security card  
**1670** - W-2 form  
**1671** - SSA-1099 form  
**1672** - Non-SSA-1099 form  
**1673** - Pay stub with your name and SSN on it  
**1674** \*\*3. Proof of [REDACTED] Residency (at least two documents):\*\*  
**1675** - State or federal tax return  
**1676** - Utility bill (must be dated within the last 90 days)  
**1677** - Lease agreement or mortgage statement  
**1678** - Bank statement (must be dated within the last 90 days)  
**1679** - Any mail from any government agency (dated within the last 90 days)  
**1680** \*\*Additional Notes:\*\*  
**1681** - All documents must be originals or certified copies.  
**1682** - If your name has changed due to marriage, divorce, or court order, you will need to provide documents that link your current name to your former name.  
**1683** - You must also complete an "Application for Permit, Driver License or Non-Driver ID Card" (Form MV-44).  
**1684** Before going to the DMV, it's recommended to use the [REDACTED] DMV Document Guide which provides a personalized checklist of the required documents based on the specifics of your application. This can be found on the official [REDACTED] DMV website.  
**1685** Make sure to double-check with the DMV for the latest requirements or any specific circumstances that may affect your application to ensure a smooth process.  
**1686** My user needs to know if the current task is specific and actionable.  
**1687** Reason: The task does not need to be further decomposed for two reasons. The first reason is that the GPT response provides direct answer to the task and generates a list of all forms of acceptable proofs for the documents required to get a non-driver ID. The second reason is that the primary goal of the task - identify necessary documents for the non-driver ID application - can be viewed as a singular, cohesive deliverable. Despite involving various types of documents, the task is focused on compiling a comprehensive list, which makes it actionable as a single unit. The distinction lies in the focus on gathering all necessary documentation, a clear and direct objective.  
**1688** Answer: No  
**1689** Now, let's start prediction:  
**1690**

Fig. 16. The prompt for few-shot + CoT + Draft for Subtask Detection.

1665 You are a useful assistance to detect if the current task needs to be further decomposed if it is not actionable and the primary goal of the task can not be viewed as a singular, distinctive deliverable. Specifically, the task is positioned on a tree structure and is associated with a level. Based on the user prompt, the current node level of the task on the tree, and the GPT response, please output Yes if it needs to be decomposed; No otherwise meaning it is actionable and does not require task decomposition. Please also provide explanations for your choice.

1666 Here are some examples:

User: My user is working on the task Research on Prospective Companies and Positions: Conduct a comprehensive search on potential companies and specific research scientist internship positions in the NLP field. Understand what each role entails, identify skill requirements, and evaluate how they align with your research interests. The current node level of the task is 1. The GPT response to the task is:  
Sure, I will list some promising companies that offer research scientist internship positions in Natural Language Processing (NLP) and briefly describe what each role entails:

1669 1. \*\*Google AI\*\*:  
- "Role": As a research intern, you can work in areas such as machine intelligence, machine perception, natural language processing, and quantum computing. Google typically looks for research experience, ability to design and execute a research agenda, and coding skills.

1670 2. \*\*Facebook AI Research (FAIR)\*\*:  
- "Role": Natural Language Processing Internship roles at FAIR involve developing novel techniques for NLP tasks which include generating human-like text. They require experience with machine learning, AI, deep learning, dialogue management, and computer vision.

1671 3. \*\*Amazon Alexa AI\*\*:  
- "Role": Interns in this role generally work on language understanding capabilities of Alexa, advancing state-of-the-art on tasks like entity resolution or text classification. Knowledge in machine learning, especially deep learning for NLP, is key for this role.

1672 4. \*\*OpenAI\*\*:  
- "Role": Research internships at OpenAI are highly selective. Interns are expected to have published in machine learning or AI and can spend a significant portion of their time on original research.

1673 5. \*\*IBM Watson\*\*:  
- "Role": Watson Interns can work on various AI/NLP tasks from fact extraction, relation extraction, disambiguation, to summarization, based on the team you join. A strong AI/ML foundation and coding skills are required.

1674 6. \*\*Apple AI/ML\*\*:  
- "Role": Apple offers internships where you can work on Siri's language understanding capabilities. They typically require machine learning knowledge and experience, especially in deep learning.

1675 7. \*\*Microsoft Research\*\*:  
- "Role": Interns in this role could work on a variety of NLP challenges, including developing algorithms that enable, for example, conversational interactions with AI. Strong machine learning knowledge, particularly in deep learning, is expected.

1676 8. \*\*Baidu Research\*\*:  
- "Role": Interns can work on various novel NLP tasks such as language understanding, language generation, and more. Baidu looks for individuals with robust knowledge in machine learning, deep learning, and coding abilities.

1677 9. \*\*NVIDIA - AI Research\*\*:  
- "Role": Interns have an opportunity to work on projects in the realms of machine learning, deep learning, computer vision, and natural language processing. They seek candidates with strong ML and DL backgrounds.

1678 When selecting a specific position, I recommend considering the following:  
- Does the role align with your career objectives?  
- Does the role play to your strengths or offer opportunities to develop in areas of interest?  
- Does the role match your skill set? If not, can you acquire those skills?

1679 Remember, a role can be an opportunity to learn and grow, so don't be deterred if you lack a few skills. If the role excites you, go for it!

1680 My user needs to know if the current task is specific and actionable  
Reason: this task needs to be further decomposed as the GPT response is not good enough to help user complete the task. Moreover, it is positioned on the first level of the tree and it involves more than one deliverables: search on companies and search on positions. To complete this task, there are multiple subtasks that need to be done separately on the next level of the tree. These include conducting a comprehensive search on potential companies, searching for specific research scientist internship positions, an analysis of what each role entails, identifying skill requirements, and evaluation of alignment with the user's research interests.  
Answer: Yes ■

1681 User: My user is working on the task Identify Potential Universities: Create a list of universities that offer PhD programs in HCI. The selection can be based on factors such as reputation, HCI research focus, published HCI research papers, faculty expertise etc. The current node level of the task is 2. The GPT response to the task is:  
Here is a list of universities that offer PhD programs in HCI:  
1. Stanford University, USA  
2. Massachusetts Institute of Technology (MIT), USA  
3. Carnegie Mellon University (CMU), USA  
4. University of California, Berkeley, USA  
1691 5. University of Washington, USA  
6. University of Michigan, USA  
1692 7. University of California, San Diego, USA  
8. Cornell University, USA  
9. University of Toronto, Canada  
10. University of College London, UK  
1693 My user needs to know if the current task is specific and actionable.  
Reason: The GPT response is good enough and it provides direct answer to complete the task. Additionally, this task does not need to be further decomposed as it is positioned on the second level of the tree and it just involves one deliverable: create a list of schools that offer PhD programs in HCI. Although it may require several steps to create the list, the end goal of this task is to get a list. Therefore the task is actionable.  
Answer: No

1694 User: My user is working on the task Identify Required Documents: Research and confirm all the necessary documents required for the non-driver ID application, ensuring to list all forms of acceptable proofs such as a birth certificate or passport for identity, Social Security Card or W-2 form for Social Security number, and utility bills or lease agreement for proof of residency. The current node level of the task is 1. The GPT response to the task is:  
To apply for a non-driver ID in [REDACTED] you need to provide several documents to prove your identity, Social Security Number (SSN), and residency. Below, I have listed the acceptable documents for each category based on the latest guidelines provided by the [REDACTED] Department of Motor Vehicles (DMV).  
\*\*1. Proof of Identity and Date of Birth (must total 6 points according to [REDACTED] DMV's point system):\*\*  
- U.S. passport or U.S. passport card (valid or expired within the past five years) – 4 points  
- Birth certificate (U.S., Puerto Rican, Canadian, or Mexican) - 2 points  
- Employment Authorization Card (I-766) – 3 points  
- Permanent Resident Card (I-551) – 3 points  
- Foreign passport with valid U.S. Visa and I-94 – 3 points  
1700 \*Note: At least one document must show your date of birth and at least one document must have your full Social Security Number.\*  
\*\*2. Proof of Social Security Number:\*\*  
- Social Security card  
- W-2 form  
- SSA-1099 form  
- Non-SSA-1099 form  
- Pay stub with your name and SSN on it  
1705 \*\*3. Proof of [REDACTED] Residency (at least two documents):\*\*  
- State or federal tax return  
- Utility bill (must be dated within the last 90 days)  
- Lease agreement or mortgage statement  
- Bank statement (must be dated within the last 90 days)  
- Any mail from any government agency (dated within the last 90 days)  
1706 \*\*Additional Notes:\*\*  
- All documents must be originals or certified copies.  
- If your name has changed due to marriage, divorce, or court order, you will need to provide documents that link your current name to your former name.  
- You must also complete an "Application for Permit, Driver License or Non-Driver ID Card" (Form MV-44).  
1711 Before going to the DMV, it's recommended to use the [REDACTED] DMV Document Guide which provides a personalized checklist of the required documents based on the specifics of your application. This can be found on the official [REDACTED] DMV website.  
1712 Make sure to double-check with the DMV for the latest requirements or any specific circumstances that may affect your application to ensure a smooth process.  
1713 My user needs to know if the current task is specific and actionable.  
Reason: Although the task is positioned on the first level of the tree, the task does not need to be further decomposed for two reasons. The first reason is that the GPT response provides direct answer to the task and generates a list of all forms of acceptable proofs for the documents required to get a non-driver ID. The second reason is that the primary goal of the task - identify necessary documents for the non-driver ID application - can be viewed as a singular, cohesive deliverable. Despite involving various types of documents, the task is focused on compiling a comprehensive list, which makes it actionable as a single unit. The distinction lies in the focus on gathering all necessary documentation, a clear and direct objective.  
Answer: No

1714 Now, let's start prediction:

Fig. 17. The prompt for few-shot + CoT + Tree + Draft for Subtask Detection.

Manuscript submitted to ACM