Designing a Proactive Context-Aware AI Chatbot for People's Long-Term Goals

Brennan Jones
brennanj@meta.com
Meta Reality Labs Research
Redmond, WA, USA

Yan Xu
yanx@meta.com
Meta Reality Labs Research
Redmond, WA, USA

Qisheng Li
qishengli@meta.com
Meta Reality Labs Research
Redmond, WA, USA

Stefan Scherer
stefanscherer@meta.com
Meta Reality Labs Research
Redmond, WA, USA

ABSTRACT

When pursuing new complex goals such as fitness or sustainability, people often seek advice from various sources. Large language models (LLMs) such as ChatGPT have recently emerged as popular sources for information seeking, action discovery, and goal planning. However, such tools require users to provide detailed prompts, are not adaptive to the user's personal attributes or real-time contexts, and are merely reactive to the user's prompts rather than proactively guiding the user at opportune moments. We share the design of an LLM-based chatbot app that proactively recommends actions to the user for their goals based on context factors that can be detected or inferred by the user's smartphone (e.g., location, time, weather) and the user's personal profile. An early pilot field study reveals that participants enjoyed the chatbot as a personal assistant that was adaptable and flexible to their needs and kept them motivated by discovering actions toward their goals.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

human-AI interaction, human-agent interaction, language models, chatbots, context-aware computing

ACM Reference Format:
Brennan Jones, Yan Xu, Qisheng Li, and Stefan Scherer. 2024. Designing a Proactive Context-Aware AI Chatbot for People's Long-Term Goals. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3613905.3650912

1 INTRODUCTION

When people pursue new complex goals like *losing weight* or *being more eco-friendly*, they may not know where to start or what to do. They oftentimes turn to different information sources such as the Internet, friends, family, or experts to seek advice. However, seeking advice can be time consuming, and others may not always be aware of the specific daily circumstances or contexts that the individual experiences, making it harder for them to tailor the suggestions.

Various factors can impact whether, how, and to what extent one succeeds at their goal pursuit, including one's ability to properly plan realistic and feasible actions [13] and the timing of interventions that one receives [20, 22]. In order to break down a long-term goal into realistic actions, one must consider what they can reasonably do within their capabilities and constraints, which relies on an understanding of the user's *context* [6, 8], and if needed, receive effective interventions at contextually-relevant moments [20, 22].

Recently, large language models (LLMs) and LLM-based tools such as ChatGPT [1] and Microsoft Copilot [2] have emerged as popular sources for information seeking and goal planning. However, such tools often require users to craft detailed and effective prompts, which many users are not experts at doing [32]. Furthermore, these tools are not adaptive to the user's individual attributes or real-time contexts, and are merely reactive to the user's prompts rather than proactively guiding the user at opportune moments. Users may not always be aware of when they are in a good opportunity to act or discover a new action toward their goals.

Context-aware computing can understand different facets about a user's context and constraints, and use this understanding to recommend contextually-relevant actions that the user can take toward their goals. Technologies including smartphones, smart glasses, and smartwatches can capture context, which includes *environmental context* (i.e., from one's immediate environmental surroundings, captured via sensors or fetched from the Internet) and personal context (e.g., related to one's goals, interests, availability, personality, preferences, and behaviors), and use this to generate and deliver action recommendations at the right moments, thus providing a *just-in-time* approach [20, 22]. Context-aware recommendations combined with LLMs have the potential to help users discover new actions, especially for new goals and unfamiliar contexts.

We designed an LLM-based chatbot smartphone app that proactively recommends actions to the user for their long-term goals as they go about their day, based on context factors that can be detected or inferred by the user's phone (including location, time, and weather), the user's personal profile, and the explicit questions and feedback that the user gives. The user can converse with the chatbot to ask for action recommendations based on their current context, ask for more details, advice, or clarification on a recommended action, give the chatbot more constraints or details about their needs, set reminders to do an action at a later time or a different context, or log completed actions.

An early pilot field study of this app reveals that participants enjoyed the chatbot as a personal assistant that was adaptable and flexible to their needs and context, and kept them accountable and

[1]ChatGPT: https://chat.openai.com/
[2]Microsoft Copilot: https://copilot.microsoft.com/

CHI EA '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0331-7/24/05.
https://doi.org/10.1145/3613905.3650912

motivated by discovering actions toward their goals. Compared to traditional LLM interfaces like ChatGPT, participants expressed that using this chatbot required less effort from them, given that it was already aware of their goals, attributes, and context. This work contributes the application of LLMs for the novel use case of generating proactive, contextualized, personalized, and feasible action recommendations for individuals' goals, and delivering such recommendations in-situ at the right moments and locations.

2 BACKGROUND

"Just-in-time" ([20, 22]) recommendations have been employed for goals such as fitness [5, 11, 14, 23] and health management (e.g., [21]). For example, *WalkMore* [5] provides context-aware reminders to walk at opportunistic times based on phone usage data, the user's physical activity level, and the time of day. RecFit [11] suggests physical activities based off context factors such as risk tolerance, expense, location, time, and weather. Other examples include a system that recommends walking routes based on one's current location and the amount of calories they want to consume that day [23], and a system that recommends fitness activities by matching a user to other users with a similar body-mass index [14]. Other work has proposed utilizing factors such as the presence of other people [21], medical data [9, 21], and calendar/availability data [12]. Our research expands on these works by introducing a system that provides context-aware and adaptive recommendations not just for single goals or even single goal categories (e.g., health, sustainability), but is flexible to a wider variety of goals, contexts, and user profiles.

LLMs can leverage prior knowledge from their training data and adapt to new contexts and goals through in-context learning [2, 18, 26, 28, 29] and prompt engineering (e.g., [27, 30, 31, 35]). LLMs have been used for recommendations (e.g., [7, 15, 16, 25, 33, 34]), and AI more broadly has been used for receiving advice in areas such as investing (e.g., [4]), nutrition (e.g., [24]), medicine [10], and disease screening (e.g., [3, 17]). In these cases, the user actively seeks advice and prompts the system directly to get a response. In our setting however, we focus on agents that fetch the user's context in the background, generate contextualized recommendations proactively, and deliver them 'just-in-time.'

3 DESIGN PROCESS AND RATIONALE

We followed an iterative design process to design, implement and test our context-aware action-recommendation app from lower to higher-fidelity prototypes, with multiple rounds of testing and iteration. Initially, we implemented the app as a traditional mobile app, where different functions are represented as menus or icons. However, user testing revealed that people had a greater variety of needs and requirements, and each user may have different needs. Thus, we experimented with using a chatbot interface to accommodate the flexible user needs. We further tested and iterated with over ten internal users, observing their usage of the chat interfaces and incorporating their feedback. Here, we share the list of design requirements (DRs) that go into the choices of our prototype design:

- **[DR1] Action recommendations should be delivered both per the user's request (user-initiated) and proactively (system-initiated).** Our users' feedback indicated that users do not always know the best moments or contexts to act toward their goals. They expressed that having the app continuously observe their contexts and deliver recommendations at opportune moments can benefit them.
- **[DR2] Support users' inquiries at different levels of detail.** Sometimes users know exactly what they want, and at other times (especially for new goals or unfamiliar contexts) they are less aware of their exact preferences and more open to discovering and exploring new ideas. The chatbot should be flexible to both situations, and be able to answer both detailed and open queries from the user.
- **[DR3] Action recommendations should be specific, detailed, and accomplishable given the user's present context and attributes.** Our users wanted advice that is feasible and specific to them and their contexts. Proper planning and breaking down of actions into feasible steps can lead to more successful goal pursuit [13], and LLMs are well-poised to tap into users' contexts and personal profile attributes to generate personalized recommendations that are detailed, specific, and feasible for the user in their present context.
- **[DR4] Action recommendations should consider a variety of user contexts.** Our users expressed that various context factors influence what actions they can do. Thus, it is beneficial for the app to consider multiple context factors, and as many of these as possible should be automatically fetched or inferred by the system.
- **[DR5] The outputs from the app should be adaptive to the user's feedback and interactions.** Our iterations revealed that users want to work with the app as a partner over time, and converse with it back and forth to tweak and adjust its recommendations. Thus, in addition to its own context observations, the app should adapt its outputs to the user's explicit feedback and strive to learn more about the user over time.
- **[DR6] The app should let the user override any context that the system incorrectly fetches or infers.** The app should be able to account for inaccuracies and missing information, and any information that the user explicitly gives the app should take precedence over information that is observed automatically by the app.
- **[DR7] The user should be able to leverage the chat interface as a versatile, flexible window for various functions.** In addition to conversing with the chatbot, the user should be able to use the chat interface to ask the chatbot to perform a variety of functions such as setting reminders, logging actions, changing or updating their goals, and manually editing their contexts.

4 SYSTEM DESIGN AND IMPLEMENTATION

Figure 1 illustrates the design of our chatbot app. We designed the system as an 'always-on' app that can be used in the foreground (through the chat interface) and in the background as the user goes about their day.

The user first creates a profile by completing a Qualtrics survey, where they provide the following information:

- **Name**, **age**, and **gender** (all optional)

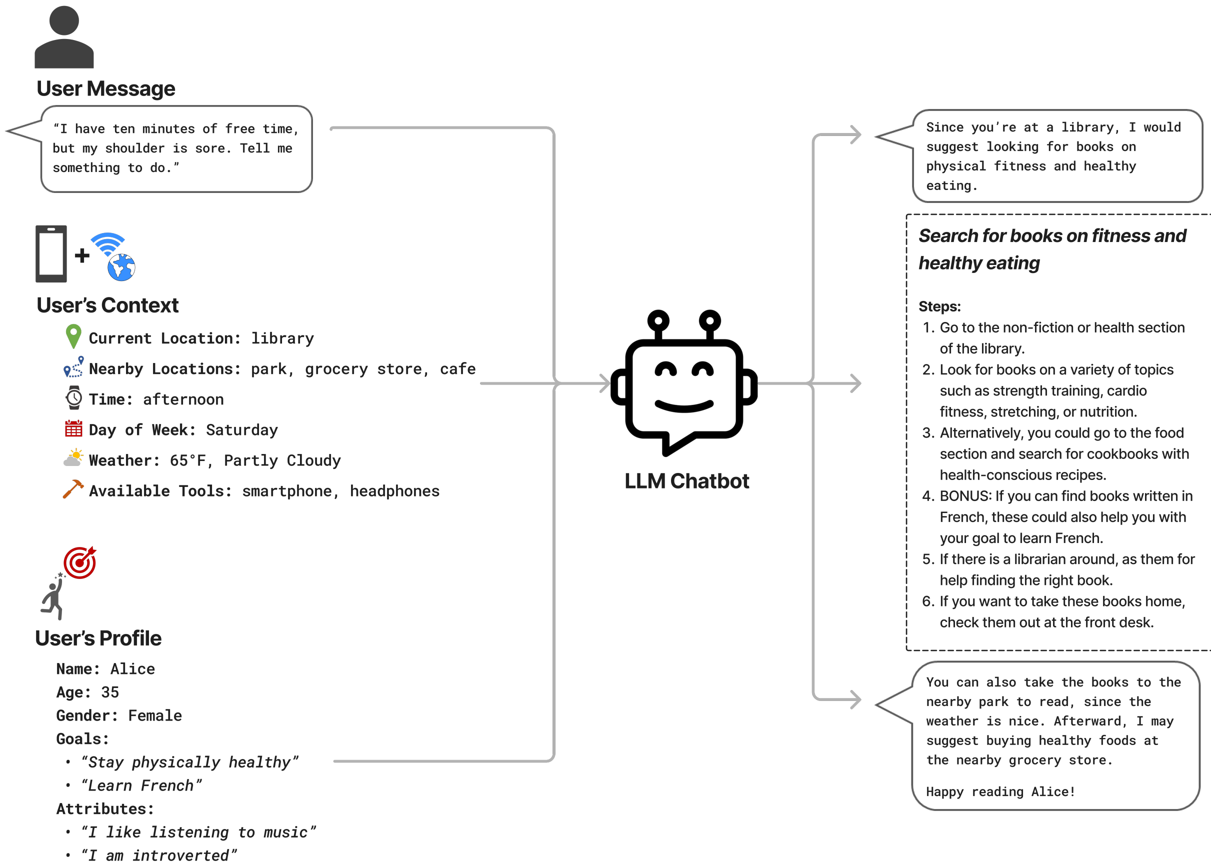

Figure 1: An example illustration of the flow of information into the chatbot LLM, and the chatbot's outputs. In this example, the user types a message into the chat interface asking for an action recommendation. The user's realtime context (including location, time, weather, etc.) and personal profile (including goals and attributes) are both fed into the LLM alongside the user's message. The chatbot uses this information to generate an action recommendation for the user that is detailed, specific, personalized, and adapted to the user's context. This action recommendation is broken down into a series of steps that are accomplishable for the user given their current context and profile attributes. The chatbot also acts as a polite and motivational assistant to the user.

- List of the user's **long-term goals**
- Advantages and disadvantages (i.e., **attributes**) in pursuing long-term goals. Examples could include:
  - *"I am very good at time management."*
  - *"I have children, and so tend to get busy as a parent."*
- List of **tools** that the user has access to at *home*, *work*, and *on the go* that are relevant to the user's goals

The user can open the app to ask the chatbot questions at any time. The chatbot continuously tracks the user's context (including their location, time, weather, etc.), and always considers this context and the user's profile when responding to the user and generating recommendations. This context tracking also runs when the app is in the background, so that the chatbot can proactively message the user with recommendations at opportune moments, even when the user is not actively using the app.

The app UI is composed of two primary views. The **chat view** (Figure 2, left) is the space where the user converses back and forth with the chatbot. In this view, action recommendations from the chatbot appear with hyperlinks, which when clicked direct the user to the **action details view** (Figure 2, right). This view provides a space for the user to view 'structured details' about an action recommended to them by the chatbot, including (i) a list of *steps* to complete the action, (ii) the *goals* that the action supports, (iii) a description of which of the user's current *context factors* enable the action to be completed and how they enable the action, and (iv) a description of which of the user's *profile attributes* enable the action to be completed and how they enable the action, and (v) a ranking of the amount of *effort* expected for the user to complete the action.

Using the chat view, the user can converse with the chatbot as they would with a traditional LLM-based chatbot. In addition, the user can perform the following actions directly in the chat interface:

- Ask the chatbot to provide a **new action recommendation**. In this case, the chatbot responds with a new recommended

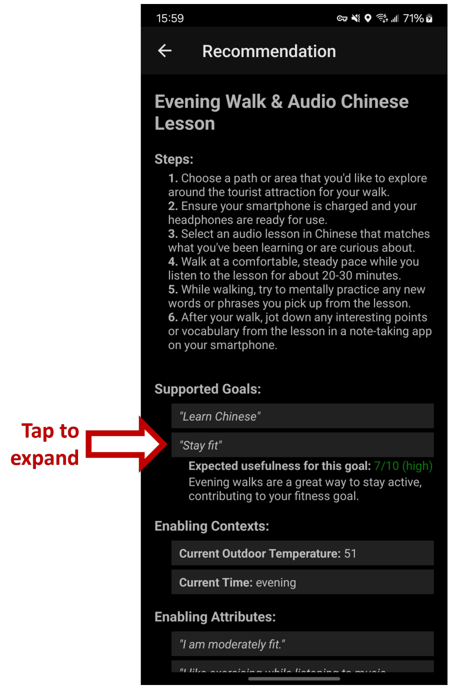

Figure 2: The UI of the chatbot app, composed of two primary views: the *chat* view (left); and the *action details* view (right).

action based on any constraints that the user gives to the chatbot.

- Specify to the chatbot any **changes in their goals**, including new goals or changes to existing goals. In this case, the chatbot updates the user's goals in their profile.
- **Change their current context** (e.g., tell the chatbot where they are located or what the weather conditions are). In this case, the context that the user specifies would override any context that the system detects or infers.
- **Log an action** as 'completed', 'will consider', 'liked', or 'disliked'. In this case, the chatbot saves the action to the user's personal action log, which they can then view on a separate page in the app at any time.
- Ask the chatbot to **remind them about an action** at a later specified time or when the user is in a specified context (e.g., when the user is at home). In this case, the chatbot flags the action, and when it detects that the specified time arrived or the user is in the specified context, it sends a reminder about the action to the user.

4.1 Implementation

The client app runs on Android and was implemented using React Native. The server was implemented using Node.js, stores the user's profile, and uses GPT-4 Turbo as the chosen LLM, running on a Microsoft Azure cloud instance.

Whenever the user sends a message to the chatbot using the phone app, the app sends to the server a JSON object containing the message text, the phone's GPS location, the phone's system time, and a flag that indicates if the user is currently located at their 'home' location, 'work' location, or neither. The server then uses the user's GPS location to fetch the current weather conditions at the user's location using the US National Weather Service (NWS) API [3]. This API returns the current temperature and a short natural-language description of the current weather conditions at the user's location (e.g., *"Sunny"*, *"Overcast with a chance of rain"*, *"Thunderstorms"*). Next, if the user is not currently located at their 'home' or 'work' addresses, the server then uses the OpenStreetMap (OSM) Nominatim API [4] to fetch a natural-language description of the user's location using reverse geocoding (e.g., *"library"*, *"coffee shop"*, *"gym"*, *"shopping mall"*, etc.). The server also uses the OSM Overpass API [5] to fetch *nearby 'place types'* (e.g., *"cafe"*, *"library"*, *"shopping mall"*) within a 400-meter radius of the user's current location.

The server then compiles a prompt and sends the following information to the LLM, structured as JSON inputs:

(1) The user's profile
(2) A list of the previous actions that the user logged (as 'completed', 'will consider', 'liked', or 'disliked')
(3) High-level daily summaries of the conversations between the user and the chatbot for each previous day that the user used the chatbot:
   - These are generated for every user once per day at midnight and stored on the server. When generating these summaries, the LLM is instructed to *"focus primarily on*

[3]US National Weather Service API: https://www.weather.gov/documentation/services-web-api
[4]OpenStreetMap Nominatim API: https://nominatim.org/
[5]OpenStreetMap Overpass API: https://wiki.openstreetmap.org/wiki/Overpass_API

what the user asked for, which actions the chatbot recommended, and how the user reacted/responded to these recommendations."

(4) The most recent previous messages (from within the last 24 hours) sent between the user and the chatbot
(5) The text contents of the user's current message to the chatbot
(6) The user's current context, which includes:
   - The natural-language description of the user's location (*"home"*, *"work"*, or the third-place description fetched from the OSM Nominatim API)
   - A list of 'place types' (e.g., *"cafe"*, *"library"*, *"shopping mall"*) near the user's current location, fetched from the OSM Overpass API
   - The user's current time as a text representation (*"morning"*, *"afternoon"*, *"evening"*, or *"night"*)
   - The current day of the week for the user (*"Monday"*, *"Tuesday"*, *"Wednesday"*, etc.)
   - The current outdoor temperature at the user's location, in Fahrenheit, fetched from the NWS API
   - The description of the current weather conditions at the user's location, fetched from the NWS API
   - A list of the tools or objects that the user listed in their profile that they have access to at 'home', 'work', or 'on the go', depending on what their current location is

The LLM is instructed to respond to the user by playing the role of an *"expert in the user's high-level goals, a personal coach and mentor to the user, and an expert in multi-tasking"* (from the system prompt). It is instructed to respond politely, address the user by their name (contained in their profile) with the intent of adding a layer of personalization, and to make use of a variety of function calls [6] it has available to it to recommend new actions, edit the user's goals, edit the user's context, log actions, or set reminders for the user, based on the user's request. Further, the LLM is instructed to make as effective use of the conversation history and the user's personal profile details as it can, and to try to learn about the user over time. Lastly, the LLM is instructed to always be polite, not to give specialized health advice (although general, high-level health advice is okay), and to avoid discussing topics related to *"violence, crime, sex, politics, uncouth behavior, or other highly-sensitive topics"*.

4.1.1 Background Updates and Proactive Recommendations. The user's phone periodically (about once every 15 minutes) sends background updates of the user's context to the server. These background updates contain all the information that is normally sent with a chat message to the server except for a chat message itself. The server then determines whether to prompt the LLM to generate an action recommendation for the user using this context. The LLM is prompted to generate a recommendation if the last recommendation sent to the user was sent more than three hours ago, it is not currently night time, and *at least* one of the following is true:

- The user has not received a new recommendation for more than eight hours, or
- The user is at a location that is not 'home' or 'work', and they have been at that location for at least 15 minutes

[6] OpenAI function calling: https://platform.openai.com/docs/guides/function-calling

5 PILOT FIELD STUDY AND FUTURE WORK

We ran an early pilot study where two participants (P-A1 and P-A2) installed the prototype app onto their phones and used it for two and a half weeks in the contexts of their day-to-day lives and their real goals. At the beginning of the study, each participant created their user profile by filling out the Qualtrics survey, and participated in a one-on-one interview with one of the researchers, where they were asked to discuss their goals, the motivation(s) behind them, and how they pursue their goal(s) today.

During the first week of the deployment, the chatbot feature on the app was disabled, and the app was instead configured to send reminders (as push notifications) to the user to pursue their goals four times per day (once in the early morning, once near noon, once in the late afternoon, and once in the evening). Each notification would mention one of the user's goals from their profile at random, and pair it with a motivational message. For example:

> ***Remember your goal to "Learn French":***
> *Look around. Can you think of something, anything, to do toward your goal in your current context?*

These served as *non-context-aware* reminders for the user to act toward their goal(s). The purpose of this was to provide participants with a baseline to compare the experience of using the chatbot and receiving context-aware recommendations to, as this experience is close to that provided by notifications from some consumer apps (e.g., Duolingo). Starting in the second week of the study, the chatbot feature of the prototype was enabled, and this experience replaced the non-context-aware reminder notifications that the participants used in the first week.

During the deployment period, participants were asked to use the app at least once per day (i.e., to open the app to chat with the chatbot, or at least read its recommendation(s) to the user), and to fill out a short survey at least four times per week (but preferably every day) where they reported on one or more actions that they completed toward their goal(s) that day. At the end of the two and a half weeks, each participant took part in a final one-on-one interview with the researcher, where they discussed their experiences using the chatbot for pursuing their goals.

5.1 Preliminary Findings

Overall, participants found the chatbot to be useful for their goals. Participants described the chatbot as being similar to a *"coach"* (P-A1) or *"personal assistant"* (P-A2) that would *"just meet you wherever you are"* (P-A1) and remove the *"administrative element"* (P-A2) from one's goal pursuit. Participants found the chatbot to be useful for **action discovery**, for its ability to **adapt its suggestions** to their needs and circumstances, and for its **proactiveness**, both in the sense that it proactively collected and made use of the user's context and user profile and that it proactively gave the user recommendations on its own without the user asking. P-A1 valued that the chatbot **required less effort** to produce more detailed, personalized, and contextualized advice, compared to more traditional LLM interfaces like ChatGPT, given that it already had his goals, attributes, and context at its disposal.

5.2 Limitations and Future Work

While users found value in the context-aware chatbot, there are still limitations that need to be addressed. For example, LLMs have been known to produce ineffective (e.g., untruthful, inaccurate, or low-quality) outputs [1], and thus our system could produce such outputs as well, especially when given incomplete, inaccurate, or low-quality inputs either as contextual data collected by the device (e.g., an inaccurate location or weather forecast) or incomplete information from the user (e.g., incomplete profile data or missing information in the chat messages). Some users may also have privacy concerns, and thus may not want to give the chatbot all of the information that it is capable of collecting. Such situations would also lead to the chatbot having limited context information, which could affect the quality of its outputs. Thus, for future work, we are interested in evaluating the effectiveness of the chatbot's outputs, assessing users' reactions when receiving lower-quality outputs, exploring how the chatbot's outputs can be improved through more effective prompting and use of context and user inputs, and exploring more ways for users to intervene when they receive suggestions from the chatbot that are inaccurate or undesirable.

We also plan to run more field studies of this prototype with more participants over longer time periods to learn more about how individuals make use of such agents over time as they make progress in their goals and as their life situations change. As part of future work, we are interested in further exploring the **types of roles** that such chatbot agents can play in individuals' goal pursuits, beyond just the roles of *"coach"* (P-A1) or *"personal assistant"* (P-A2) mentioned by our participants. For example, both of our participants expressed a desire for the chatbot to assist with *longer-term goal planning* (e.g., coming up with weekly plans), *progress tracking* (e.g., checking in with the user on their progress), and *providing accountability* (e.g., following up with the user to see if they did an action). We are also interested in improving the proactiveness of the chatbot and the timing and delivery of its interventions, for example, by employing more sophisticated machine-learning approaches (e.g., [19]) to determine the best moments to intervene.

6 CONCLUSION

With LLMs, action recommendation systems that are adaptable to various goals and contexts, flexible around a diversity of user needs and characteristics, and proactive in their suggestions become more feasible. Smartphones that are capable of using real-time context information are already ubiquitous in people's daily lives, and devices such as smart glasses and smartwatches, which are capable of collecting and making use of even more context cues, are becoming increasingly prevalent. This work has begun to explore the use of LLMs as proactive agents for giving users contextualized and personalized guidance for their goals, and delivering such guidance at the right times and locations. We introduced an early prototype design for a chatbot app running on a smartphone, presented preliminary findings from its use in the field, and provided future directions for the design and research of such agents.

ACKNOWLEDGMENTS

We thank our colleagues Mary Anne Hood, Mohammad Shahidul Kader, Hamid Eghbalzadeh, and Hannah Nguyen from Meta Reality Labs Research for their extensive design feedback and help putting together the prototype and study protocol, recruiting participants, and executing the study. We would also like to thank the anonymous reviewers for their valuable feedback on this work.

REFERENCES

[1] Razvan Azamfirei, Sapna R. Kudchadkar, and James Fackler. 2023. Large language models and the perils of their hallucinations. *Critical Care* 27, 1 (March 2023), 120. https://doi.org/10.1186/s13054-023-04393-x

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[3] Muhammad E. H. Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar Reajul Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al Emadi, Mamun Bin Ibne Reaz, and Mohammad Tariqul Islam. 2020. Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access* 8 (2020), 132665–132676. https://doi.org/10.1109/ACCESS.2020.3010287 Conference Name: IEEE Access.

[4] Alton Y. K. Chua, Anjan Pal, and Snehasish Banerjee. 2023. AI-enabled investment advice: Will users buy it? *Computers in Human Behavior* 138 (Jan. 2023), 107481. https://doi.org/10.1016/j.chb.2022.107481

[5] Xiang Ding, Jing Xu, Honghao Wang, Guanling Chen, Herpreet Thind, and Yuan Zhang. 2016. WalkMore: promoting walking with just-in-time context-aware prompts. In *2016 IEEE Wireless Health (WH)*. 1–8. https://doi.org/10.1109/WH.2016.7764558

[6] George T Doran. 1981. There's a SMART way to write management's goals and objectives. *Management review* 70, 11 (1981), 35–36.

[7] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2023. Recommender Systems in the Era of Large Language Models (LLMs). https://doi.org/10.48550/arXiv.2307.02046 arXiv:2307.02046 [cs].

[8] BJ Fogg. 2009. A behavior model for persuasive design. In *Proceedings of the 4th International Conference on Persuasive Technology (Persuasive '09)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/1541948.1541999

[9] Andreas Hamper. 2015. A Context Aware Mobile Application for Physical Activity Promotion. In *2015 48th Hawaii International Conference on System Sciences*. 3197–3206. https://doi.org/10.1109/HICSS.2015.386 ISSN: 1530-1605.

[10] Claudia E. Haupt and Mason Marks. 2023. AI-Generated Medical Advice—GPT and Beyond. *JAMA* 329, 16 (April 2023), 1349–1350. https://doi.org/10.1001/jama.2023.5321

[11] Qian He, Emmanuel Agu, Diane Strong, and Bengisu Tulu. 2014. RecFit: a context-aware system for recommending physical activities. In *Proceedings of the 1st Workshop on Mobile Medical Applications (MMA '14)*. Association for Computing Machinery, New York, NY, USA, 34–39. https://doi.org/10.1145/2676431.2676439

[12] Kazi Sinthia Kabir and Jason Wiese. 2022. The Challenge for Just-in-Time Adaptive Interventions: Incomplete or Missing Data. (2022).

[13] Kathrin Krause and Alexandra M. Freund. 2014. How to Beat Procrastination. *European Psychologist* 19, 2 (Jan. 2014), 132–144. https://doi.org/10.1027/1016-9040/a000153 Publisher: Hogrefe Publishing.

[14] Jong Won Lee, Han Kil Kim, and Hoe Kyung Jung. 2016. User Analysis Mechanisms based Mobile Fitness System. *International Journal of Electrical and Computer Engineering (IJECE)* 6, 6 (2016), 3154–3160.

[15] Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation. https://doi.org/10.48550/arXiv.2304.03879 arXiv:2304.03879 [cs].

[16] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. https://doi.org/10.48550/arXiv.2306.05817 arXiv:2306.05817 [cs].

[17] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg S. Corrado, Ara Darzi, Mozziyar Etemadi, Florencia Garcia-Vicente, Fiona J. Gilbert, Mark Halling-Brown, Demis Hassabis, Sunny Jansen, Alan Karthikesalingam, Christopher J. Kelly, Dominic King, Joseph R. Ledsam, David Melnick, Hormuz Mostofi, Lily Peng, Joshua Jay Reicher, Bernardino Romera-Paredes, Richard Sidebottom, Mustafa Suleyman, Daniel Tse, Kenneth C. Young, Jeffrey De Fauw, and Shravya Shetty. 2020. International evaluation of an AI system for breast cancer

screening. *Nature* 577, 7788 (Jan. 2020), 89–94. https://doi.org/10.1038/s41586-019-1799-6 Number: 7788 Publisher: Nature Publishing Group.

[18] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? https://doi.org/10.48550/arXiv.2202.12837 arXiv:2202.12837 [cs].

[19] Varun Mishra, Florian Künzler, Jan-Niklas Kramer, Elgar Fleisch, Tobias Kowatsch, and David Kotz. 2021. Detecting Receptivity for mHealth Interventions in the Natural Environment. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (June 2021), 74:1–74:24. https://doi.org/10.1145/3463492

[20] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. 2018. Just-in-time adaptive interventions (JITAIs) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine* 52, 6 (2018), 446–462. Publisher: Oxford University Press US.

[21] Shriti Raj, Kelsey Toporski, Ashley Garrity, Joyce M. Lee, and Mark W. Newman. 2019. "My blood sugar is higher on the weekends": Finding a Role for Context and Context-Awareness in the Design of Health Self-Management Technology. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300349

[22] Donna Spruijt-Metz and Wendy Nilsen. 2014. Dynamic Models of Behavior for Just-in-Time Adaptive Interventions. *IEEE Pervasive Computing* 13, 3 (July 2014), 13–17. https://doi.org/10.1109/MPRV.2014.46 Conference Name: IEEE Pervasive Computing.

[23] Yasufumi Takama, Wataru Sasaki, Takafumi Okumura, Chi-Chih Yu, Lieu-Hen Chen, and Hiroshi Ishikawa. 2017. Walking Route Recommender for Supporting a Walk as Health Promotion. *IEICE TRANSACTIONS on Information and Systems* E100-D, 4 (April 2017), 671–681. https://search.ieice.org/bin/summary.php?id=e100-d_4_671&category=D&year=2017&lang=E&abst= Publisher: The Institute of Electronics, Information and Communication Engineers.

[24] Chun-Hua Tsai, Sathvik Kadire, Tejesvi Sreeramdas, Matthew VanOrmer, Melissa Thoene, Corrine Hanson, Ann Berry, and Deepak Khazanchi. 2023. Generating Personalized Pregnancy Nutrition Recommendations with GPT-Powered AI Chatbot. In: 20th International Conference on Information Systems for Crisis Response and Management. *20th International Conference on Information Systems for Crisis Response and Management (ISCRAM)* 2023 (May 2023), 263–271. https://digitalcommons.unomaha.edu/isqafacpub/127

[25] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023. RecMind: Large Language Model Powered Agent For Recommendation. https://doi.org/10.48550/arXiv.2308.14296 arXiv:2308.14296 [cs].

[26] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. https://doi.org/10.48550/arXiv.2109.01652 arXiv:2109.01652 [cs].

[27] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. https://doi.org/10.48550/arXiv.2201.11903 arXiv:2201.11903 [cs].

[28] Sang Michael Xie and Sewon Min. 2022. How does in-context learning work? A framework for understanding the differences from traditional supervised learning. http://ai.stanford.edu/blog/understanding-incontext/

[29] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. https://doi.org/10.48550/arXiv.2111.02080 arXiv:2111.02080 [cs].

[30] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. https://doi.org/10.48550/arXiv.2305.10601 arXiv:2305.10601 [cs].

[31] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).

[32] J Zamfirescu-Pereira, Richmond Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems (CHI'23)*.

[33] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language models as recommender systems: Evaluations and limitations. In *NeurIPS 2021 Workshop on I (Still) Can't Believe It's Not Better.* https://www.amazon.science/publications/language-models-as-recommender-systems-evaluations-and-limitations

[34] Xianglin Zhao, Li Chen, and Yucheng Jin. [n. d.]. Using ChatGPT as a recommender system: A case study of multiple product domains. https://pilab-hkbu.github.io/blog/Blog/Using%20ChatGPT%20for%20recommendations_%20A%20case%20study%20of%20multiple%20product%20domains.html

[35] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. https://doi.org/10.48550/arXiv.2205.10625 arXiv:2205.10625 [cs].