Investigating User Perceptions of Conversational Agents for Software-related Exploratory Web Search

Matthew Frazier
University of Delaware
matthew@udel.edu

Shaayal Kumar
University of Delaware
shaayal@udel.edu

Kostadin Damevski
Virginia Commonwealth University
kdamevski@vcu.edu

Lori Pollock
University of Delaware
pollock@udel.edu

ABSTRACT

Conversational agents that respond to user information requests through a natural conversation have the potential to revolutionize how we acquire new information on the Web (i.e., perform exploratory Web searches). Recent advances to conversational search agents use popular Web search engines as a back-end and sophisticated AI algorithms to maintain context, automatically generate search queries, and summarize results into utterances. While showing impressive results on general topics, the potential of this technology for software engineering is unclear.

In this paper, we study the potential of conversational search agents to aid software developers as they acquire new knowledge. We also obtain user perceptions of how far the most recent generation of such systems (e.g., Facebook's BlenderBot2) has come in its ability to serve software developers. Our study indicates that users find conversational agents helpful in gaining useful information for software-related exploratory search; however, their perceptions also indicate a large gap between expectations and current state of the art tools, especially in providing high-quality information. Participant responses provide directions for future work.

CCS CONCEPTS

• **General and reference → Empirical studies**.

KEYWORDS

Conversational search agent, Wizard of Oz study, exploratory search

ACM Reference Format:
Matthew Frazier, Shaayal Kumar, Kostadin Damevski, and Lori Pollock. 2022. Investigating User Perceptions of Conversational Agents for Software-related Exploratory Web Search. In *New Ideas and Emerging Results (ICSE-NIER'22), May 21–29, 2022, Pittsburgh, PA, USA*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3510455.3512778

ICSE-NIER'22, May 21–29, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9224-2/22/05…$15.00
https://doi.org/10.1145/3510455.3512778

1 INTRODUCTION

Written resources that aid software development, (e.g., Q&A sites, tutorials, documentation) are abundant on the Web. Therefore, software developers consider searching for information using popular Web search engines indispensable to their daily work. However, software developers are often unsuccessful in quickly finding high-quality information related to their task. For instance, in a large-scale study of one million Web search sessions by software developers beyond code search and debugging, Rao et al. found that software engineering (SE)-related queries are less effective than other types of queries [12]. SE-related queries resulted in higher rates of query reformulations, fewer clicks, and shorter dwell time compared to non-SE sessions. This is especially the case for exploratory information search (unlike lookup of a previously known document), which formulates search as an iterative process of acquiring new knowledge [11].

Researchers have observed that a natural interface for conducting exploratory information search may be conversations with a conversational search agent [8, 21]. For instance, a growing proportion of difficult Web search queries, which do not result in successful retrieval, are reformulated by users into conversation-starting questions [10]. The advantages of using conversations with a search system, over conventional information retrieval are: 1) question-asking dialogues are a communication modality that humans are well versed in using; 2) the retrieval system can pose clarifying questions to better understand the user need and context; and 3) the user can provide incremental positive or negative (explicit) feedback that can help guide the system in the search process. Relative to interacting with an actual human via public chat platforms or community Q&A, the advantages of automated conversational search systems are in improving the speed and efficiency of obtaining answers and in not having the social burden of asking bad questions or too many questions [15]. Recent work by Zhang et al. demonstrates the promise of conversational agents for helping software developers meet their information needs [19], in their case, for improving question retrieval in software-related Q&A forums.

A new generation of conversational search agents do not use a fixed corpus, but instead operate by querying the Web using a search engine like Google (in the background) to take advantage of advancements in information retrieval algorithms and to ensure that the corpus is kept updated [9]. In effect, these systems perform two summarizing processes, one that takes the latest question and conversational context and generates a search query and a second process that takes the retrieved results and produces a response

Evaluation Task (Investigative): You are a business owner tasked with migrating data to the cloud. Write a short paragraph evaluation of three potential cloud storage solutions.
Planning/Forecasting Task (Investigative): You are giving a presentation on randomized testing in software engineering. Outline the structure of the presentation, including titles of each topic area, and briefly describe the content of each topic area.
Comparison Task (Learning): You are in need of a sorting algorithm for an application you are designing. Write a short paragraph describing similarities and differences between bubble sort and bucket sort. When would you use each?
Knowledge Acquisition Task (Learning): You are an agile Embedded Software Engineer who creates applications for iOT devices. You are tasked with learning as much as possible about creating Linux device drivers. Write a short paragraph summarizing what you have learned.

Table 1: Task examples.

to the user. BlenderBot2, recently introduced by Facebook, is the most cutting edge incarnation of this paradigm for conversational search [7]. We refer to conversational agents that are backed by a search engine instead of a fixed corpus as Open-Domain Conversational Search Agents (ODCSAs).

To explore the potential of ODCSAs for exploratory search in software engineering and to understand how far the latest generation of such tools have come, we investigate the following two research questions.

RQ1: *Can ODCSAs help software engineers be effective in performing exploratory search?* We answer this question by performing a Wizard-of-Oz (WoZ) study, where software developers interact with a human (i.e., a wizard) acting as an ODCSA. The participant believes that she is interacting with an automated system. The wizards are prepared for the topic of discussion (i.e., exploratory search task), via our study setup, so we can gain understanding of the limits of how helpful an ODCSA can be for a software engineer.
RQ2: *How far does the current generation of Web-based ODCSAs (i.e., BlenderBot2) go in helping software developers with exploratory search?* We answer this question be performing a study parallel to the WoZ study, but where the ODCSA is BlenderBot2. We compare the participants' perceptions of BlenderBot2 to their perceptions of the human wizards along dimensions such as the bot's ability to understand directions and to provide useful information.

The results of our study show that (1) software engineers do find ODCSAs useful at helping with exploratory search, yet (2) software developers believe that the current generation of Web-based ODCSAs are not effective in helping with exploratory search.

2 RESEARCH DESIGN

Tasks: Exploratory Search for SE. Based on researchers observing that conversations are a natural interface for conducting exploratory search [4, 8, 21], we focused our study on participants completing multi-step SE tasks involving exploratory search (sample tasks are shown in Table 1). We created tasks to cover both categories of exploratory searches - Learning and Investigative [11]. Learning through web search involves multiple iterations and interpretation. Investigative search involves analysis, synthesis, and evaluation and requires substantial extant knowledge.

To engage participants in exploratory searches for Learning, we created software-related tasks involving Knowledge Acquisition and Comparison. Knowledge Acquisition tasks have open-ended search goals, because Learning tasks have no clear criteria on when to end the search. They involve developing new knowledge and thus include self-directed life-long learning and professional learning. Comparison tasks involve gathering information about two or more topics to analyze similarities and differences between them.

To engage participants in exploratory searches for Investigation, we created software-related tasks involving Planning/Forecasting and Evaluation. Planning tasks involve gathering overviews of a new area in preparation for a future activity. This task involves consulting many documents with no given straightforward process in doing so. Evaluation tasks involve critically analyzing information and providing objective estimates of the amount of effort and resources required to accomplish the task.

We created two tasks for each of Knowledge Acquisition, Comparison, Planning/Forecasting and Evaluation. We chose these exploratory search tasks because they align with and are representative of common software developer daily tasks that take multiple search iterations over long periods of time or may involve areas to minimize gaps in knowledge such as writing new software programs, testing, debugging, and installing programs. We designed each task by reverse engineering tasks associated with different software developer professions. The following criteria were used for task creation: 1) The tasks were simple enough that knowledge could be learned on the subject easily while complicated enough to take at least 20 minutes to synthesize; 2) The tasks were broad enough to yield a wide variety of supplemental or tangential information on the subject, yet narrow enough to draw a specific conclusion; 3) Wizards could understand the tasks and subject well enough to give meaningful responses to questions.
Participants. We recruited 18 senior undergraduate and graduate student participants. All participants had at least two years prior programming experience in at least one major programming language (e.g., Java, C, Python) in an academic or professional setting.
Open-Domain Conversational Search Agents.
Amelia: WoZ-based ODCSA. Participants communicated via a private Slack workspace created specifically for the study. The wizards consisted of the first two authors - one PhD and one undergraduate Computer Science student. Prior to serving as a wizard, the wizards familiarized themselves with each of the exploratory search tasks used in the Amelia study and surveyed a broad set of the canned responses curated by the first author covering most aspects of each topic. When no prior canned responses were prepared, the wizards aggregated text found on websites obtained from top search engine results to curate their response.

At the start of each session with Amelia, the wizard logged into Slack from two accounts simultaneously, one account for the simulated ODCSA Amelia and another with their personal account. Using the personal Slack account, the wizard sent the participant instructions on how to access Amelia using Slack and instructions for the exploratory search task. Then, the participant worked on completing the task by asking Amelia (i.e., the wizard) questions intended to gather the necessary information.
BlenderBot2 Extended with SE Word Embeddings. For the state of the art ODCSA, we utilized the BlenderBot 2.0 pretrained model developed by Facebook AI Research [7]. The cornerstone of the model's novelty is its long term memory component and its ability

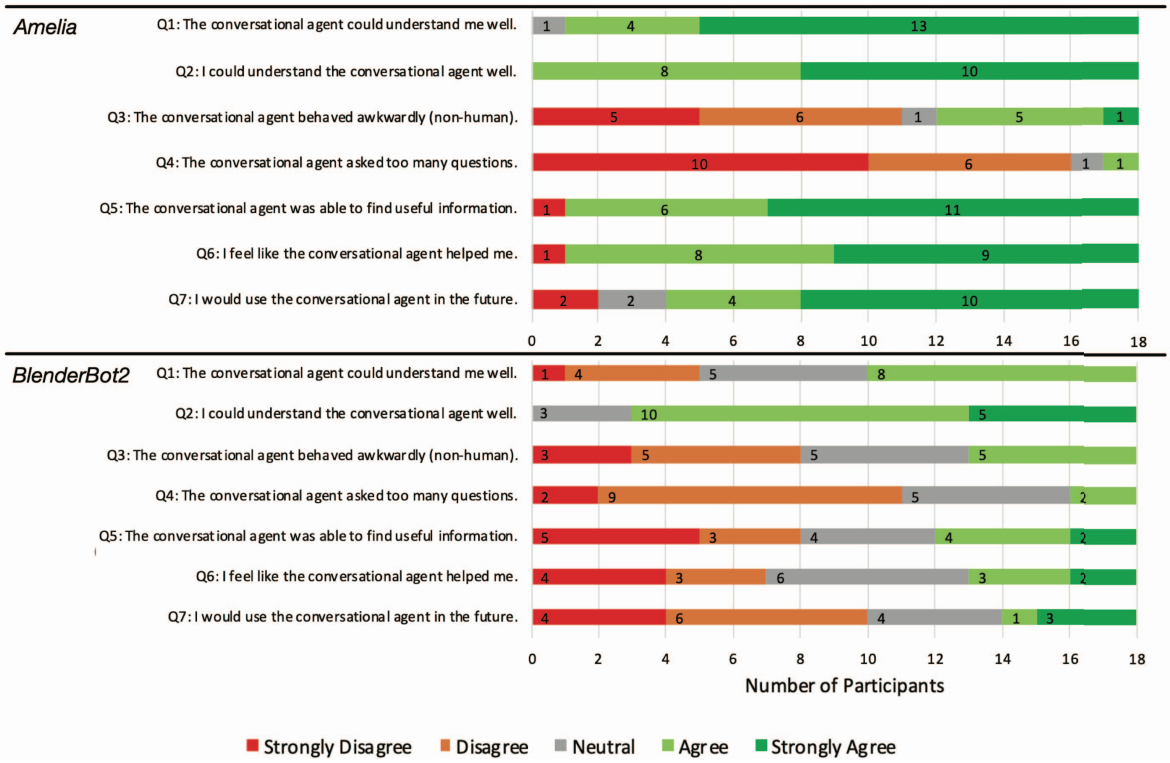

Figure 1: Survey results for Amelia (top) and BlenderBot2 (bottom).

to access the Internet and dynamically generate responses [9, 18]. Since it is known that general-purpose NLP algorithms do better by customizing to the SE domain, we adapted the search component of BlenderBot2 with SE query expansion. Specifically, we added the top 3 most semantically-related terms to the BlenderBot2 generated query, based on GloVe v1.2 word embeddings pre-trained on the entirety of Stack Overflow (using the Stack Overflow data dump as of June, 2020) with default parameters. Participants interacted with the model through the web socket chat interface hosted on an AWS EC2 instance.

Procedure. Prior to participant recruitment, we conducted four rounds of pilot sessions with students who did not participate in the study. These sessions helped us identify problems with our study design, improve our tasks to make them clearer, and fully test the interfaces and how well users could understand the ODCSAs.

During the study, each participant completed two different tasks of the same exploratory search intent [2, 11]. Each participant used Amelia on one task and Blenderbot on another task of the same search intent. We varied the ordering of the use of the ODCSAs and tasks among participants to avoid learning bias.

Working remotely, all participants were asked to close all applications while participating in the study with the exception of a text editor. The goal was to avoid any potential distractions as well as allow participants the ability to brainstorm and perform notetaking while performing each exploratory search task.

Participants were free to use each ODCSA in any way of their choice. They were not given any instructions on how to interact with each ODCSA, except to avoid jargon. Participants were given up to 20 minutes to solve each task. After each of their two tasks, they completed a short post-study survey. The post-study survey starts with a set of 5-point Likert Scale questions designed to gather information on the participant's perception of the ODCSA's helpfulness and conversational interface. Through open questions, we ask what they like and dislike about the ODCSAs and any other comments they want to offer. Finally, we conclude with demographic information questions.

3 RESULTS AND DISCUSSION

Figure 1 presents the quantitative results from our post-survey. We answer RQ1: *"Can ODCSAs help software engineers be effective in performing exploratory search?"* using the Amelia chart of results as well as the responses to three open-ended questions. 17 of the 18 participants agreed or strongly agreed with both the statements "The conversational agent was able to find useful information." and "I feel like the conversational agent helped me." Furthermore, 14 of 18 agreed or strongly agreed with "I would use the conversational agent in the future."

The positive perception of the helpfulness of ODCSAs for software engineers is additionally supported by the participants' open-ended responses. When asked what they disliked about Amelia,

almost all participants said they did not dislike anything; one participant mentioned the somewhat vague answers at times. When asked what they liked about Amelia, several different reasons were given. For instance, P7 stated "relating the discussion topic with previous messages, providing useful resources, code snippets."

Not surprisingly, participants responded positively to their interactions with Amelia, where they interacted unknowingly with a human wizard. All agreed or strongly agreed with "I could understand the conversational agent well.", and all but one (who was neutral) agreed or strongly agreed with "The conversational agent could understand me well." Interestingly, 5 of 18 agreed and 1 strongly agreed that Amelia behaved awkwardly (non-human)!

To answer RQ2: *"How far does the current generation of Web-based ODCSAs (i.e., BlenderBot2) go in helping software developers with exploratory search?"*, we compared the responses about experiences using Amelia to responses about experiences using BlenderBot2. Our results indicate that there is a large gap between the current generation of ODCSAs and what participants want in such conversational agents for software-related exploratory search. In contrast to the overwhelming positive perceptions of the usefulness of information provided and helpfulness of Amelia, only 6 of 18 participants responding about their BlenderBot2 experience agreed or strongly agreed with "The conversational agent was able to find useful information." and only 5 of the 18 participants agreed or strongly agreed with "I feel like the conversational agent helped me." Only 4 of 18 agreed or strongly agreed with "I would use the conversational agent in the future."

The results with BlenderBot2 were mixed. While 15 of the participants agreed or strongly agreed they could understand the ODCSA, only 8 of 18 agreed that the ODCSA could understand them. Only 5 thought that the ODCSA behaved awkwardly and only 2 thought it asked too many questions. However, the open-ended responses include: "A real conversation - much more human-like than Amelia," "Relatively fast," and "Human-like syntax".

Dissatisfaction with the helpfulness and usefulness of the information from BlenderBot2 is revealed in the open-ended responses. Many indicate there is a need to customize the ODCSA to the SE domain. For example, some responses include "No understanding of simple jargon," "Very limited in getting useful information to complete the task," "Some contradictory information," "Vague, generic answers," "not detailed enough," "Not able to answer any of my questions," and "Would love to use this bot if more database was added to it that could give me clearer answers to my questions."

Other open-ended responses about what they liked about Amelia provide further insight into the features users want in an ODCSA for exploratory search in SE. These include: "timely response," "handles typos," "familiar interface (through Slack)," "personable," and "felt like a real conversation with a human."

In summary, our study indicates that users of a conversational agent backed by a wizard find the agent helpful in gaining useful information from exploratory search involved in software-related tasks. However, their perceptions as evidenced through a post-survey after using both agents indicate there is a large gap between a representative current ODCSA and the wizard-backed agent. The open-ended responses indicate likes and dislikes that suggest that customization for software engineers could help address. We provide a replication package at https://tinyurl.com/2c7t3wwp.

4 THREATS TO VALIDITY

To enhance construct validity, we constructed our SE tasks based on Athukorala et al. which lend themselves to exploratory search [2]. We tested the study procedure and survey on several non-authors and made improvements. Initial test sessions of BlenderBot2 revealed the chatbot hallucinated on some responses within the SE domain, which we mitigated by automatically extending BlenderBot2's search queries. Results could vary with a well-tuned BlenderBot2 model trained on an SE corpus. Internal threats due to possible task/tool use ordering bias were minimized by assigning the same tasks to different participants in different orders and changing the tool use order. Interpretation bias in the qualitative analysis of open-ended survey questions was minimized by multiple researchers conducting the analysis. The main external threat is generalizability beyond our participant set. We recruited both undergraduates and graduate students from various courses in two universities. Replication to broader demographics is encouraged. We focused on exploratory search where more query reformulation is common; our results may not generalize to other kinds of search.

5 RELATED WORK

The most related studies to this paper are WoZ experiments in dialogue strategies by Eberhart et al. [5] and speech act detection by Wood et al. [16]. Both help to address the need for more experimental datasets for SE virtual assistants by making available their conversations, annotations, participant survey results, and recommendations for improving future software engineering conversational agent studies. However, in both studies, participants complete tasks that initiate lookup search intents backed by a corpus (e.g. API documentation) whereas our experiment's tasks foster learning and investigative exploratory search intents while aggregating Internet sources, synthesizing, and summarizing knowledge.

Within the SE domain, researchers have developed virtual assistant prototypes that automate API documentation question answering [6, 14], perform Git operations [3], refactor projects [1], or answer developers' technical questions [13, 17]. Our work most closely relates to GitterAns and Chatbot4QR. GitterAns is a Gitter bot that automatically detects when a developer asks a technical question in a chat and provides links to Stack Overflow of possible answers[13]. Chatbot4QR performs interactive query refinement for question retrieval on Stack Overflow by assisting users in recognizing and clarifying technical details missed in queries [20]. Both systems use highly canned responses and do not aim to foster multi-turn conversations with developers.

6 FUTURE PLANS

Based on the promising results of this study on the perceived helpfulness of ODCSAs for exploratory search in SE, we plan research towards closing the gap between the current state of the art and wizard-based agent. We will focus on features that participants liked that were perceived inadequate in the state of the art. This includes: customizing existing agents for software engineers towards improving the usefulness of the agent responses, reducing response time, including examples with answers, and examining the use of an agent's long-term memory.

REFERENCES

[1] Vahid Alizadeh, Mohamed Amine Ouali, Marouane Kessentini, and Meriem Chater. 2019. RefBot: Intelligent Software Refactoring Bot. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 823–834. https://doi.org/10.1109/ASE.2019.00081

[2] Kumaripaba Athukorala, Dorota Głowacka, Giulio Jacucci, Antti Oulasvirta, and Jilles Vreeken. 2016. Is exploratory search different? A comparison of information search behavior for exploratory and lookup tasks. *Journal of the Association for Information Science and Technology* 67, 11 (2016), 2635–2651. https://doi.org/10.1002/asi.23617 arXiv:https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.23617

[3] Nick C. Bradley, Thomas Fritz, and Reid Holmes. 2018. Context-Aware Conversational Developer Assistants. In *Proceedings of the 40th International Conference on Software Engineering* (Gothenburg, Sweden) *(ICSE '18)*. Association for Computing Machinery, New York, NY, USA, 993–1003. https://doi.org/10.1145/3180155.3180238

[4] Paul A. Crook, Alex Marin, Vipul Agarwal, Samantha Anderson, Ohyoung Jang, Aliasgar Lanewala, Karthik Tangirala, and Imed Zitouni. 2018. Conversational Semantic Search: Looking Beyond Web Search, Q&A and Dialog Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 763–766. https://doi.org/10.1145/3159652.3160590

[5] Zachary Eberhart, Aakash Bansal, and Collin Mcmillan. 2020. A Wizard of Oz Study Simulating API Usage Dialogues with a Virtual Assistant. *IEEE Transactions on Software Engineering* (2020), 1–1. https://doi.org/10.1109/TSE.2020.3040935

[6] Hamza Ed-Douibi, Gwendal Daniel, and Jordi Cabot. 2020. *OpenAPI Bot: A Chatbot to Help You Understand REST APIs*. 538–542. https://doi.org/10.1007/978-3-030-50578-3_40

[7] https://parl.ai/projects/blenderbot2/. [n.d.]. BlenderBot 2.0: An open source chatbot that builds long-term memory and searches the internet. Accessed: 2021-10-07.

[8] Johannes Kiesel, Arefeh Bahrami, Benno Stein, Avishek Anand, and Matthias Hagen. 2018. Toward Voice Query Clarification. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. 1257–1260.

[9] Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-Augmented Dialogue Generation. *CoRR* abs/2107.07566 (2021). arXiv:2107.07566 https://arxiv.org/abs/2107.07566

[10] Qiaoling Liu, Eugene Agichtein, Gideon Dror, Yoelle Maarek, and Idan Szpektor. 2012. When Web Search Fails, Searchers Become Askers: Understanding the Transition. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Portland, Oregon, USA) *(SIGIR '12)*. Association for Computing Machinery, New York, NY, USA, 801–810. https://doi.org/10.1145/2348283.2348390

[11] Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Commun. ACM* 49, 4 (2006), 41–46.

[12] Nikitha Rao, Chetan Bansal, Thomas Zimmermann, Ahmed Hassan Awadallah, and Nachiappan Nagappan. 2020. Analyzing Web Search Behavior for Software Engineering Tasks. In *IEEE International Conference on Big Data, Big Data 2020, Atlanta, GA, USA, December 10-13, 2020*, Xintao Wu, Chris Jermaine, Li Xiong 0001, Xiaohua Hu 0001, Olivera Kotevska, Siyuan Lu, Weija Xu, Srinivas Aluru, ChengXiang Zhai, Eyhab Al-Masri, Zhiyuan Chen 0003, and Jeff Saltz 0001 (Eds.). IEEE, 768–777. https://doi.org/10.1109/BigData50022.2020.9378083

[13] Ricardo Romero, Esteban Parra, and Sonia Haiduc. 2020. Experiences Building an Answer Bot for Gitter. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (Seoul, Republic of Korea) *(ICSEW'20)*. Association for Computing Machinery, New York, NY, USA, 66–70. https://doi.org/10.1145/3387940.3391505

[14] Yuan Tian, Ferdian Thung, Abhishek Sharma, and David Lo. 2017. APIBot: Question answering bot for API documentation. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 153–158. https://doi.org/10.1109/ASE.2017.8115628

[15] Alexandra Vtyurina, Denis Savenkov, Eugene Agichtein, and Charles L. A. Clarke. 2017. Exploring Conversational Search With Humans, Assistants, and Wizards. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 2187–2193.

[16] Andrew Wood, Paige Rodeghero, Ameer Armaly, and Collin McMillan. 2018. Detecting Speech Act Types in Developer Question/Answer Conversations during Bug Repair. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Lake Buena Vista, FL, USA) *(ESEC/FSE 2018)*. Association for Computing Machinery, New York, NY, USA, 491–502. https://doi.org/10.1145/3236024.3236031

[17] Bowen Xu, Zhenchang Xing, Xin Xia, and David Lo. 2017. AnswerBot: Automated generation of answer summary to developers' technical questions. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 706–716. https://doi.org/10.1109/ASE.2017.8115681

[18] Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond Goldfish Memory: Long-Term Open-Domain Conversation. arXiv:2107.07567 [cs.CL]

[19] N. Zhang, Q. Huang, X. Xia, Y. Zou, D. Lo, and Z. Xing. 2020. Chatbot4QR: Interactive Query Refinement for Technical Question Retrieval. *IEEE Transactions on Software Engineering* (2020), 1–1. https://doi.org/10.1109/TSE.2020.3016006

[20] Neng Zhang, Qiao Huang, Xin Xia, Ying Zou, David Lo, and Zhenchang Xing. 2020. Chatbot4QR: Interactive Query Refinement for Technical Question Retrieval. *IEEE Transactions on Software Engineering* (2020), 1–1. https://doi.org/10.1109/TSE.2020.3016006

[21] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. New York, NY, USA, 177–186.