Towards a Dataset to Assess Conversational Agent's Efficacy in Mentoring Newcomer Developers

Misan Paul Etchie, Hunter Beach, Katia Felizardo, Igor Steinmacher
Northern Arizona Univerisity (NAU), Flagstaff, Arizona, United States
mpe45@nau.edu, hmb433@nau.edu, katiascannavino@utfpr.edu.br, igor.steinmacher@nau.edu

***Abstract*—Many Open Source Software (OSS) maintainers experience burnout due to the constant need to support and answer questions posed by newcomer developers. This leads to slower response times, and unanswered questions which discourage newcomers and hinder community growth. Research indicates that conversational agents trained on community data can answer common questions, thus reducing maintainers' workload. However, conversational agents should only be used if they are useful to newcomers. To help prove a conversational agent can be useful, we manually curated a dataset of questions posted to the JabRef issue tracker and Gitter, submitted by newcomers. This dataset of questions will be invaluable for conducting studies using LLMs to support newcomers. This dataset paper details our collection process, describes the data, and offers reflections on future research.**

***Index Terms*—Open Source Software, Newcomers, Dataset, Human-Bot Collaboration**

I. Introduction

Open source software (OSS) is integral to modern technology, with 98% of applications using OSS components, comprising about 70% of their code base [1]. As such, ensuring the sustainability of these communities is essential, but many maintainers feel overwhelmed. Surveys show that 48% of maintainers feel unappreciated, and 60% wish to quit or have already [2]. These findings highlight the need for better support to maintain OSS stability.

With rising code volume in popular repositories, maintainers face constant pressure. This strain creates barriers for novice developers who struggle to contribute when maintainers are overworked and unreachable [3]. Delayed or inadequate responses, often due to overwork, reinforce these barriers. Discussions on platforms like Hacker News show overwhelmed maintainers questioning whether to respond promptly, politely, or at all [4]. Addressing this burden is crucial for a more welcoming and sustainable contributor environment.

Conversational agents can support newcomers and reduce maintainers' workload if effectively implemented. Wessel et al. found that 90% of OSS contributors are receptive to bot assistance for completing pull requests [5]. Dominic et al. proposed a conversational agent for newcomer questions, while Serrano et al. explored agents for task selection [6] [7]. Correia et al. found that LLM-generated answers were often preferred over human responses [8]. Despite this potential, the lack of a dataset of real OSS project questions and answers hinders development and evaluation.

To bridge this gap, we collected and curated a corpus of questions posted by newcomer contributors actively developing for JabRef OSS project. This paper presents the data collection and curation method applied, the results obtained, and a discussion of the dataset itself.

II. Method

This section presents the project under analysis and details the method used to collect, classify, and filter the questions. Figure 1 presents an overview of the method.

A. Our Case: JabRef Reference Manager

JabRef is a popular open-source reference manager with over 2.2M downloads. Researchers widely use it to organize bibliographic data. Its active developer community with over 1.2K pull requests and 300 issues in the last 12 months fosters global discussions on contributions and technical issues. Given its popularity and active community, many aspiring OSS contributors see it as a great place to start. Because of this, JabRef receives a significant number of newcomers. The project's GitHub repository currently lists 677 contributors.

B. Data Collection

We manually collected questions from JabRef's GitHub issue tracker and Gitter chat as they are the primary channels for technical inquiries about the project. We focused on issues and comments from August to September 2024. We did this to avoid the pitfalls of cherry-picking, ensure that the questions were not too old, and that our proposed conversational agent would be able to answer them provided with current information. Finally, this dataset of recent questions makes the process of evaluating the agent answers (by maintainers) more effective.

To gather questions, we read through Gitter chat and GitHub issue threads, searched for question marks ("?"), and manually reviewed each post to ensure no questions were missed. We recorded the question content, human responses, and the preceding content on the page as context. We collected 96 questions from the GitHub issue tracker and 23 questions from Gitter. Although we found fewer Gitter questions than GitHub, the questions were more closely related to OSS development, since Gitter is more intended for developers, while GitHub has both developers and users. From this set of 119 questions, we began to categorize the questions to learn more about our data.

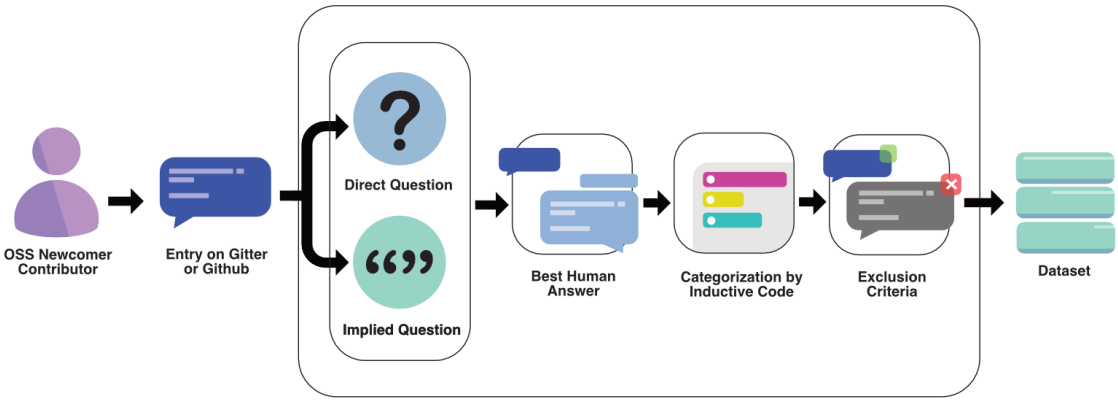

Fig. 1. Question Collection Methodology

C. Categorization

We started our analysis by classifying all 119 questions with multiple labels drawn from distinct categories. The categories were designed to capture various facets of each question from its type and nature to the context in which it was asked, as well as its purpose and the user's role. These labels fell into three broad categories: Question-Related, Response-Related, and Content-Specific. This structured approach guided our analysis process and ensured a comprehensive analysis of how questions are formed, how they are answered, and the broader context that shapes these interactions.

Within each category, we applied inductive coding to build a set of labels that emerged from our data. This approach enabled us to capture some details of each question more effectively, however, it did come with some difficulties which will be noted in the section **Lessons Learned**.

Each question was labeled by two team members, who worked independently, to capture its key aspects. For instance, labels such as "Follow-up Question," "Requesting More Info," and "Seeking Confirmation" indicate cases where the user seeks additional clarification. Once all 119 questions were labeled, we collaborated to ensure agreement between the labelers. We reviewed each question and its assigned labels, keeping those that matched, deciding on the best label if they had a similar meaning, and resolving discrepancies through discussion before deciding to retain or remove a label.

Table I details the categories related to the question type, nature, context, user information, and purpose. These categories help identify the main concerns or topics, providing insights into the typical questions raised. Table II highlights labels related to responses, detailing how they are characterized—whether they offer solutions, point to resources, or remain unanswered. Lastly, Table III covers the contextual elements that provide additional insights into the question.

We established our filtering criteria to build a robust and relevant dataset of developer questions for our research. Our dataset encompassed various question types, levels of detail, and topics. We systematically excluded entries that did not meet the necessary criteria for meaningful analysis, ensuring that only relevant questions were retained.

TABLE I
CATEGORIZATIONS FOR QUESTION RELATED LABELS

Category	Relevant Labels
Question Type	Implied Question, Rhetorical Question, Hypothetical Question, Several Similar Questions, Not a Question
Nature of Question	UI/UX Issue, Code/Coding Related, Architecture/Scope Question, Bug Reporting, Usability Issue
Question Context	Follow-up Question, Context Needed, Repeat the Question, Similar to the Previous Question
User Type	OSS User, Dev User, Newcomer Onboarding, Program User
Question Purpose	Requesting Issue Creation/Completion/Assignment, Requesting More Info, Requesting Help with Features, Locating Code, Seeking Starting Advice, Seeking Confirmation, Seeking Code Review

TABLE II
CATEGORIZATIONS FOR RESPONSE RELATED LABELS

Category	Relevant Labels
Response Type	Pointed to Resource, Outlines Fix, Robot Response, Workaround Provided, Brainstorming Fix
Response Context	Spans Responses, Quotes Asker

TABLE III
CATEGORIES OF CONTENT-SPECIFIC LABELS

Category	Relevant Labels
Content Type	Uses Large Text, Multimedia Attached, Good First Issue
Platform Specificity	Mac, Windows, Linux
Resolution	Unanswered, Closed as Completed

Our primary goal was to compile a dataset of technical newcomer questions related to JabRef, specifically focusing on recent OSS development inquiries. To achieve this, we excluded questions related to the following:

1) ***Project management questions.:*** We excluded project management-related questions, such as *"Can I work on this issue?"* or *"Can someone fix this issue?"*, since they do not contribute to newcomer engagement in OSS development or provide valuable insights into development challenges.

2) ***Software use-only questions:*** We excluded questions focused solely on usability issues, as they did not provide insight into the challenges faced by newcomer developers in OSS. For example, questions like *"When I do this, the*

program breaks" were excluded, but those referencing code or development strategies were included.

3) ***Conversational/off-topic questions****:* We excluded casual conversation questions, such as *"How are you today?"* commonly found on Gitter as they did not contribute to technical discussions or assist novice OSS developers.

4) ***User Type Exclusion****:* We excluded questions posted by users who were not newcomer developers, as our focus is on understanding the challenges faced by those new to the project. Questions originating from experienced users or long-term contributors were omitted, ensuring that our dataset accurately reflects the issues encountered by novice OSS developers.

Of the 119 questions collected, 27 were deemed relevant and accepted based on these criteria, meaning that 92 questions were removed for failing to meet our requirements. This selective approach ensured that our dataset was precisely tuned to the experiences and challenges of new developers, making the insights derived from it directly applicable to improving the efficacy of a conversational agent.

III. Results

We categorized each of the 119 questions collected initially, based on the three dimensions presented in the previous section. In this section, we present an analysis of the questions included and excluded.

Figure 2 illustrates that specific categories were more prevalent than others in accepted questions. Some examples are Code/Coding Related, Seeking Confirmation, and Usability Questions. This trend suggests that many queries revolve around technical challenges, usability, and clarifications for users who are still adjusting to the project.

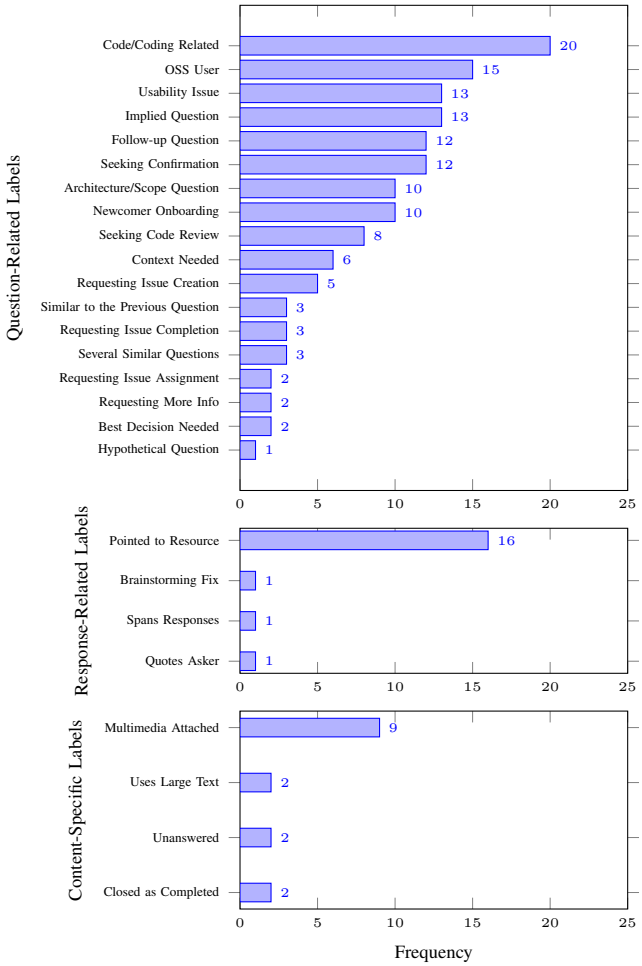

Fig. 2. Combined Distribution of Labels by Category in Their Current Order

A. Excluded Questions

We analyzed the 92 questions that did not align with our criteria. The analysis reveals that questions classified with labels such as "UI/UX Issues" and "Rhetorical Questions" were frequently dismissed, as we were not focusing on user questions. Further analysis on some tags which rarely appear can be more seen in **Lessons Learned** The aforementioned labels commonly coincided with questions posted by the software users and not new contributors. This is shown in **Figure 3**.

B. Analysis of Accepted Questions

Our dataset analysis revealed some challenges developers face when contributing to the JabRef OSS project. These insights are important for evaluating the implementation of a conversational agent and understanding how it can be tailored to effectively facilitate the onboarding of newcomer developers. In this section, we will reference the label names alongside their corresponding counts, using the format "Label Name" (Label-Count) for clarity and consistency.

1) Need for Enhanced Communication: From our analysis, we observed that some of the most common labels such as "Follow-Up Question" (68), "Scope Question" (43), "Seeking Confirmation" (34), and "Requesting More Info" (20) are related to requiring additional context, clarification, or information from the initial post. Therefore, we hypothesize that there is ample space for a conversation agent to provide this kind of information. For a conversational agent to be valuable and viable in open-source software contexts, it needs enhanced communication mechanisms, including the ability to properly reference and provide relevant documentation, offer targeted clarifications, and bridge knowledge gaps for new contributors.

2) How Maintainers Answer Questions: On the side of the answers, our analysis showed that pointing users to existing resources is a common behavior. The label aptly named "Pointed to Resource" (53), represents nearly half 119 of the answers. Given that this is existing information, the model we will use to build the conversational agent may emulate this behavior or provide the specific chunk of information that would answer the question. On one end, providing the link to the information is a good practice to provide breadcrumbs to the newcomers; on the other end, providing piecemeal

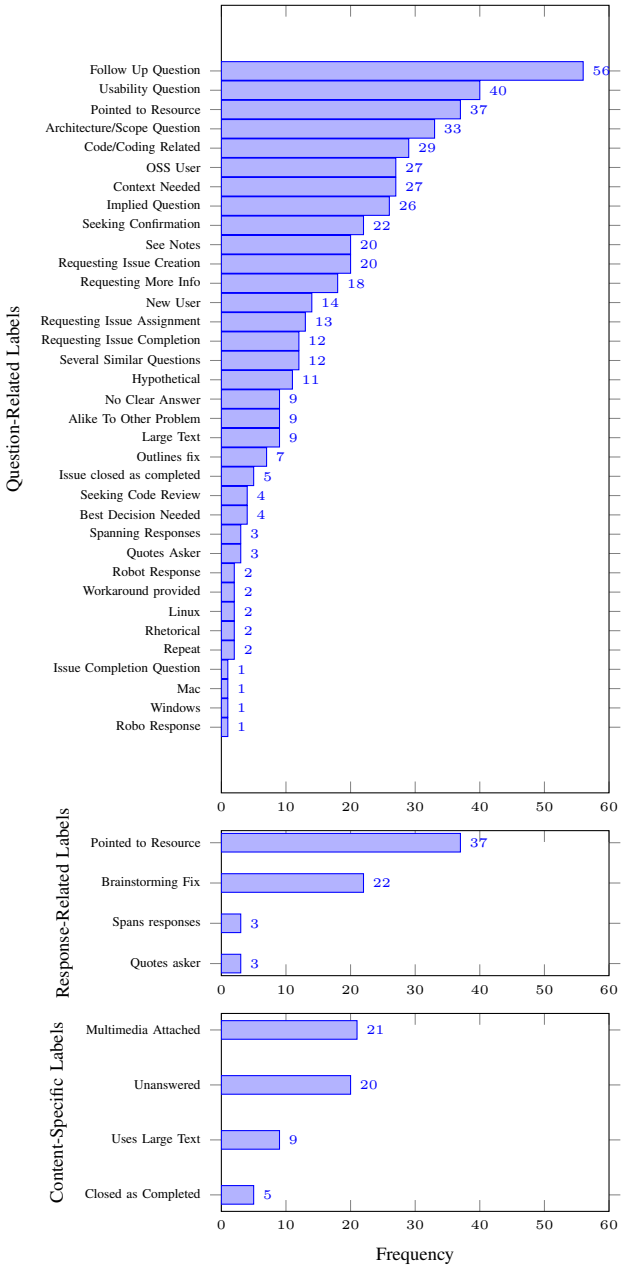

Fig. 3. Combined Distribution of Rejected Labels by Category (All Labels, Sorted in Decreasing Order)

information may be interesting for those who prefer objective information to work with.

Given that we are developing a conversational agent aimed at delivering responses that are equal to or better than those provided by human maintainers, our model should emulate these behaviors. Therefore we will focus on providing mechanisms that will retrieve relevant information at the same time that it facilitates navigation of resources.

The dataset is available at https://doi.org/10.6084/m9.figshare.28406402.

IV. Closing Remarks

A. Lessons Learned

We experimented with various platforms, initially collecting questions from Gitter, GitHub, SourceForge, and JabRef-Discourse, but ultimately focused on Gitter and GitHub. These platforms host a larger majority of development issues, making it crucial for research aimed at improving developer experiences to select the right communication channels and gather appropriate questions.

Although we reviewed all questions for dataset robustness, we recommend implementing filtering mechanisms upfront to exclude unsuitable questions more efficiently. Labels like "Requesting Issue Assignment/Completion/Creation" (53 occurrences, 10 inclusions) had low inclusion rates as they pertained more to project management than development. Future studies should carefully use these labels to avoid irrelevance.

Also, certain validity issues stemmed from our inductive labeling approach, where labels were created for management during analysis leading to labels that were irrelevant in the final data like "Best Decision Needed" (2 occurrences), intended more for internal communication. These extraneous labels diluted dataset quality by adding minimal value and extra complications. To enhance data integrity, we recommend a structured deductive labeling approach, initially analyzing a subset of questions to establish relevant criteria aligned with research goals.

B. Future Work

We aim to develop an agent leveraging JabRef's documentation, source code, and developer interactions to address the corpus's identified needs. The agent will generate responses to compare with human answers. By analyzing our labeled dataset, we will refine labels and improve agent performance. We plan to use an LLM-based conversational agent to retrieve relevant sections for question answering. Integration with JabRef for real-time interactions will follow, with surveys assessing the agent's effectiveness in aiding newcomers.

References

[1] F. Nagle, J. Dana, J. Hoffman, S. Randazzo, and Y. Zhou, "Census II of Free and Open Source Software — Application Libraries," 2022, access: Oct-24. [Online]. Available: https://www.linuxfoundation.org/research/census-ii-of-free-and-open-source-software-application-libraries

[2] Tidelift, "The 2024 tidelift state of the open source maintainer report," *Tidelift*, 2024, access: Oct-24. [Online]. Available: https://tidelift.com/open-source-maintainer-survey-2024

[3] I. Steinmacher, M. A. G. Silva, M. A. Gerosa, and D. F. Redmiles, "A systematic literature review on the barriers faced by newcomers to open source software projects," *Inf and Softw Technol*, vol. 59, pp. 67–85, 2015.

[4] Transpute, "Xz: A microcosm of the interactions in open source projects," https://news.ycombinator.com/item?id=39879710, 2024, access: Oct-24.

[5] M. Wessel, B. M. de Souza, I. Steinmacher, I. S. Wiese, I. Polato, A. P. Chaves, and M. A. Gerosa, "The power of bots: Characterizing and understanding bots in oss projects," *Proc ACM Hum Comput Interact*, vol. 2, no. CSCW, pp. 1–19, 2018.

[6] J. Dominic, J. Houser, I. Steinmacher, C. Ritter, and P. Rodeghero, "Conversational bot for newcomers onboarding to open source projects," in *BotSE 2020*, 2020, pp. 46–50.

[7] L. P. Serrano Alves, I. S. Wiese, A. P. Chaves, and I. Steinmacher, "How to find my task? chatbot to assist newcomers in choosing tasks in oss projects," in *International Workshop on Chatbot Research and Design*. Springer, 2021, pp. 90–107.

[8] J. Correia, M. C. Nicholson, D. Coutinho, C. Barbosa, M. Castelluccio, M. Gerosa, A. Garcia, and I. Steinmacher, "Unveiling the potential of a conversational agent in developer support: Insights from mozilla's pdf. js project," in *AIWARE 2024*, 2024, pp. 10–18.