The Journal of Systems and Software 230 (2025) 112541

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

LLMs and Stack Overflow discussions: Reliability, impact, and challenges☆

Leuson Da Silva [a,*], Jordan Samhi [b], Foutse Khomh [a]

[a] Polytechnique Montreal, Montreal, Canada
[b] University of Luxembourg, Luxembourg, Luxembourg

ARTICLE INFO

Dataset link: https://github.com/leusonmario/chat-stack, https://doi.org/10.5281/zenodo.15086541

Keywords:
ChatGPT
LLaMa
Stack overflow
Empirical study
Reliability

ABSTRACT

Since its release in November 2022, ChatGPT has shaken up Stack Overflow, the premier platform for developers' queries on programming and software development. Demonstrating an ability to generate instant, human-like responses to technical questions, ChatGPT has ignited debates within the developer community about the evolving role of human-driven platforms in the age of generative AI. Two months after ChatGPT's release, Meta released its answer with its own Large Language Model (LLM) called LLaMA: *the race was on*. We conducted an empirical study analyzing questions from Stack Overflow and using these LLMs to address them. This way, we aim to ① quantify the reliability of LLMs' answers and their potential to replace Stack Overflow in the long term; ② identify and understand why LLMs fail; ③ measure users' activity evolution with Stack Overflow over time; and ④ compare LLMs together. Our empirical results are unequivocal: *ChatGPT and LLaMA challenge human expertise, yet do not outperform it for some domains*, while a significant decline in user posting activity has been observed. Furthermore, we also discuss the impact of our findings regarding the usage and development of new LLMs and provide guidelines for future challenges faced by users and researchers.

1. Introduction

Practitioners must adopt different methods and approaches to support their work during software development. This involves making technical and non-technical decisions, encompassing the selection of programming languages, frameworks, and methodologies. These decisions have a prominent impact on the software development process (Yli-Huumo et al., 2016; Tamburri et al., 2013; Dias et al., 2020). When it comes to technical support, practitioners have become familiar with leveraging online Question and Answer (Q&A) web forums as instrumental aids playing an essential role in different aspects, like API learning, code compression, and fixing or getting related problem information (Rubei et al., 2020; Squire, 2015). Notably, Stack Overflow[1] stands out as a preeminent online hub within the technology community (Xia et al., 2017). However, owing to the inherent nature of these Q&A platforms, which are contingent upon the accumulation of both individual and collective human knowledge, the potential for errors and the propagation of inaccurate information remains a concern (Verdi et al., 2020; Ragkhitwetsagul et al., 2019; Zhang et al., 2018).

With the advance of Large Language Models (LLMs), practitioners and researchers have explored the potential of leveraging LLMs for different software engineering tasks; for example, pair programming assistants and simulating human behavior as questionnaire respondents (Dakhel et al., 2023; Hämäläinen et al., 2023; Liang et al., 2024; Hou et al., 2023). Such a variety of usage is motivated by the capability of LLMs to provide comprehensive and concise explanations for diverse subjects instantaneously (Touvron et al., 2023). Furthermore, with the release of ChatGPT,[2] publicly available for the general audience, users have prompted LLMs for different purposes, like simulating a Q&A online web forum. As a result, these models, by engaging in human-like tasks, can continuously learn from user feedback (e.g., via fine-tuning) and improve their accuracy (Bakker et al., 2022).

Therefore, practitioners might turn to using more and recurrently LLMs instead of checking truthful and verifiable sources. Nonetheless, exclusively relying on information generated by LLMs is a threat, as LLMs might generate inaccurate information (Goodrich et al., 2019). To overcome the imminent risks, Stack Overflow has proactively announced prohibiting the use of information generated by LLMs (StackOverflow, 2023b). However, considering the difficulty of detecting contents generated by LLMs (Tang et al., 2023; Sadasivan et al., 2023), they later announced OverflowAI, an integration of generative AI into the platform impacting users, products, and IDEs (StackOverflow, 2023a). Even under these new circumstances, it is still necessary to assess the

☆ Editor: Dr Shane McIntosh.

* Corresponding author.
E-mail address: leuson-mario-pedro.da-silva@polymtl.ca (L.D. Silva).

[1] https://stackoverflow.com/.
[2] https://openai.com/chatgpt.

https://doi.org/10.1016/j.jss.2025.112541
Received 11 December 2024; Received in revised form 25 April 2025; Accepted 17 June 2025
Available online 3 July 2025
0164-1212/© 2025 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

generated answers and determine the extent to which LLMs can be effectively utilized.

Such concern is becoming a general and recurrent topic, also targeted by other research areas. For example, Lee et al. (2023) investigate the reliability of content generated by ChatGPT-3.5 in medicine. Based on scenario examples of potential medical use, the authors evaluate the current abilities of ChatGPT under different situations. The model performs satisfactorily, though the authors also observe some hallucinations, like inferring information unrelated to previously shared content. These hallucinations were further explored in ChatGPT 4 when the model could catch them.

In the same way, evaluating the reliability of LLMs in Software Engineering while assisting developers with their daily tasks is a relevant and important topic, bringing the attention of previous studies. Kabir et al. (2023) and Liu et al. (2023) investigate the correctness of ChatGPT by comparing the answers reported with human ones. However, by reliability, we refer to the ability of LLMs to consistently produce correct and accurate behavior across time and varying conditions. While correctness is a key aspect of evaluating LLMs, reliability encompasses more than just factual accuracy. LLMs might provide correct answers, but they could also share content that, although technically correct, may still be misleading or harmful to users in some contexts. Therefore, evaluating such an aspect is crucial for ensuring LLMs do not produce misleading, harmful, or incomplete information, even when their responses are technically correct. Furthermore, previous studies focus only on ChatGPT, while they also do not deeply explore the most challenging domains faced by the model and how users deal with them. Given LLMs' current diversity, different number of parameters supported by models, and features, it is important to consider these factors when investigating and comparing findings across different models.

Researchers have also investigated the impact of LLMs in Stack Overflow. Even before the release of LLMs, a decline in the number of answers to questions related to different programming languages was observed on Stack Overflow (Blanco et al., 2020; Syam et al., 2023; Orosz, 2025). Recently, del Rio-Chanona et al. (2023) report an overall decrease in posting activity on Stack Overflow after the release of ChatGPT. However, whether such a decrease is uniform in different domains is unknown, as different topics involve different groups and their engagement. The decrease in users' activity on Stack Overflow not only affects how developers receive support for their issues but also influences future versions of LLMs, given the need to train new models with updated data.

Aiming to address the previous gaps, in this paper, we explore LLMs' reliability, challenges, and impact on Stack Overflow, focusing on exploring different models and users' activity regarding different topics and groups. To that end, we perform an empirical study analyzing questions and associated answers mined from Stack Overflow. First, we mine questions and related information from Stack Overflow (before and after ChatGPT's release). Then, we analyze the overall and by topics impact of posting activity. Second, following previous studies conducted on Stack Overflow, we select a representative number of the previously selected questions to prompt LLMs, further comparing the generated answers with the original ones provided on Stack Overflow. To mitigate previous studies' diversity issues regarding LLMs under analysis, we investigate two models: ChatGPT-3.5 and LLaMA-2-7b. Next, based on the answers provided, we further manually analyze the ones that LLMs present low similarity compared with human answers, reporting the most challenging topics they faced.

We observe that ChatGPT-3.5 significantly outperforms LLaMA-2-7b regarding answers' textual similarity. Although LLaMA-2-7b does not present significantly better results than ChatGPT, this model represents an alternative, free, and public option for the tech community. About the challenges faced by the LLMs, overall, they performed well in most domains, with some challenges regarding questions targeting *Frameworks and Libraries*, leading to different observed impacts on specific topics. Although ChatGPT presents a more neutral sentiment when generating answers, LLaMA presents a more positive sentiment. Finally, regarding the impact on Stack Overflow, our results confirm previous findings about a continuous significant decline in posting, answering, and commenting activity after ChatGPT's release. Such a decline is not uniform initially, as we observed no statistical decline for some topics, like specific frameworks and libraries. However, after one year of ChatGPT's release, we observe a constant and significant drop in posted content in Stack Overflow.

Our paper makes the following contributions:

- We report an empirical study evaluating the reliability of answers provided by ChatGPT and LLaMA in comparison to human-generated answers on Stack Overflow.
- We identify and further explore the domains where LLMs fall short and show that for some topics, like frameworks and libraries.
- We compare ChatGPT with LLaMA and show that despite the structural difference between these models, they share similar challenges.
- We reinforce previous findings about a decline in posting activity on Stack Overflow, though our results also show that such a decline is not uniform on different topics and groups.
- We provide a dataset of questions from Stack Overflow and associated scripts available through our online Appendix (Online Appendix, 2025) and Zenodo as well.[3]

The rest of the paper is organized as follows: Section 2 presents details about the LLMs evaluated here, while Section 3 explains our study setup and the steps performed. In Section 4, we present the results, which are further discussed in Section 5. Threats to the validity of our study and related work are discussed in Sections 6 and 7 respectively. Finally, in Section 8, we present our conclusions.

2. Background

This section introduces the necessary concepts and terminology used in our study.

Stack Overflow: Stack Overflow (SO) is a platform where developers can ask questions and share their knowledge with others (Barua et al., 2014). It is designed to provide accurate answers to specific programming problems, fostering a community of learning and sharing among developers. Over time, developers have adopted this platform to discuss different topics, while building general knowledge with the tech community.

Large Language Model: A large language model (LLM) is a machine learning model trained on extensive sets of types of data, like textual, images, audio, and structured data. Recently, LLMs trained on textual data brought the attention of users, scientists, and maintainers due to their support of generating human-like text (Xu et al., 2022). With millions to billions of parameters, LLMs are used for various natural language processing tasks. Different LLMs have been proposed for tasks related to software engineering, like StarCoder (Li et al., 2023a), Copilot (GitHub, 2024), and CodeBERT (Feng et al., 2020).

ChatGPT: ChatGPT is a language model developed by OpenAI. It is designed to generate human-like text based on the prompts it receives. Built on the GPT architecture, it can be used for a variety of applications, among which are code generation, summarization, translation, testing, and documentation (Zheng et al., 2023; Ozkaya, 2023). Currently, OpenAI offers different models, but in this study, we focus on ChatGPT-3.5. The motivation for this decision is driven by the recurrent adoption of this version by related work; furthermore, it is the default

[3] https://doi.org/10.5281/zenodo.15086541.

version that users have access to through the app option without further costs.

LLaMA: LLaMA, released by Meta AI, is a series of large language models that come with different sizes (e.g., 7, 13, and 70 billion parameters for LLaMA-2) (Touvron et al., 2023). Similar to ChatGPT, it spans various use cases, such as interacting with humans. In our study, we relied on the latest LLaMA version, i.e., LLaMA-2. This version has similarities with ChatGPT, like its architecture based on transformers and its generative capabilities, leading to generate coherent and contextually relevant text based on input prompts. Although this model has not been explored like ChatGPT, we aim to investigate how these models differ and complement each other. Specifically for this study, we consider LLaMA-2-7b as it represents the most accessible option for users, based on the required resources to load and use the model locally.

Although there are newer versions of the selected LLMs in this study, like GPT-4 and LLaMA 4, not evaluated here, we consider the selected versions based on their popularity and availability for the general users. Our study was conducted in two phases: first, we analyzed data before and after the release of ChatGPT, then, one year later, we collected data for the one-year milestone. When we conducted our study for the first phase in May 2023, GPT-3.5-turbo was the publicly available model. GPT-4 was released in March 2023 but was only accessible to (i) ChatGPT Plus subscribers or (ii) API users with a waitlist. This way, our analysis was necessarily based on ChatGPT-3.5, the model available to the general public at that time through its GUI without additional costs compared with ChatGPT-4. Regarding LLaMA, newer versions are known for handling more parameters and, consequently, dealing with more complex tasks. LLaMA API shares the same costs previously discussed, though their models can be locally loaded by users in their environments. However, it requires advanced resources due to hardware and software resource constraints, which might be challenging for some users.

3. Study setup

Our experimental methodology investigates the reliability, challenges, and impact of LLMs on Stack Overflow. To this end, our study aims to answer the following research questions:

- **RQ1.** *How does the reliability of answers generated by ChatGPT and LLaMA-2 compare to those provided by Stack Overflow users?*
  Previous studies have investigated the adoption of LLMs to address Stack Overflow questions by evaluating the potential of ChatGPT for such a task (Kabir et al., 2023; Liu et al., 2023). Knowing the diversity of LLMs available and their different configurations, we take a different route by evaluating the reliability of different LLMs while comparing them. Reliability in Software can be understood as the probability of having failure-free software execution for a timely and spacely specified period (Association et al., 1990). In this study, we investigate reliability based on the capability of LLMs to report answers matching the specifications reported in questions. Knowing that different answers can fix a single question in Stack Overflow, we focus on the answer previously accepted by the requester. To evaluate the reliability of LLM-generated answers, we use the accepted answer on Stack Overflow as a proxy, since it is typically considered the most reliable solution by the community, based on user feedback. We aim to investigate the capability of LLMs to report trustworthy and valuable content to users, considering the occurrence of wrong answers due to the LLM faults (Lyu, 2007). This way, we can evaluate how far or close LLMs are to the most accurate information that properly fixes the problem in that specific context. Following previous studies (Yazdaninia et al., 2021; Asaduzzaman et al., 2013), we randomly select a representative number of previously collected questions and prompt the models to answer them. Next, we compute the reliability by checking the textual and semantic similarity of the generated and original answers. Finally, knowing that answers avoiding negative comments are more likely to be accepted by Stack Overflow users (Calefato et al., 2015), we aim to assess how close the generated answers fit with general users' preferences. For that, we perform a sentiment analysis of the generated answers.
- **RQ2.** *What specific domains pose challenges for ChatGPT and LLaMA-2 when generating reliable and accurate answers?* When evaluating the adoption of LLMs to address Stack Overflow questions, previous studies' findings report how well LLMs perform (Kabir et al., 2023; Liu et al., 2023). However, there is a gap regarding how LLMs perform in different domains. Aiming to explore such a perspective, in this RQ, we investigate how to leverage LLMs' adoption for practitioners. Based on the results of RQ1, for a subset of questions (low and high text similarity), we analyze and classify their associated domains, based on the categorization proposed by Barua et al. (2014), reporting the domains that LLMs performed well and the challenging ones. Second, we explore the possible structural factors of the questions that correlate with high or low similarity.
- **RQ3.** *What is the impact of ChatGPT's public release on Stack Overflow posting activity?*
  - **RQ3.1.** *How are different domains from Stack Overflow impacted by the release of ChatGPT?*

  del Rio-Chanona et al. (2023) and Burtch et al. (2023) investigate the impact of ChatGPT in Stack Overflow. Although they report a decline in user posting activities and Stack Overflow visits, it is unknown whether such a decline is uniform for different domains, like programming languages, frameworks, and libraries. Furthermore, they investigate the immediate impact without checking the evolution over time. Burtch et al. (2023) even provide an analysis by individual topics but do not group related ones and analyze them together. Aiming to bring light to these concerns, we first mine questions and related information from Stack Overflow for five months before and after the release of ChatGPT (30th of November 2022), and after one year of the release (30th of November 2023). Second, we statistically assess the impact on the usage/engagement by comparing the number of posted questions, answers, comments, and related information. Third, to investigate the impact of ChatGPT on specific domains of Stack Overflow, we filter previously mined questions, which represent challenges for LLMs, and compare the frequency of posted questions referring to the associated tags. Based on our findings, we discuss possible factors that might influence the results and the different strategies for adopting/combining Stack Overflow and LLMs from now on.
- **RQ4.** *How has the release of ChatGPT impacted the activity patterns of Stack Overflow users?* When investigating the impact of ChatGPT in Stack Overflow, del Rio-Chanona et al. (2023) mostly focus on the posting activity, which does not cover how such an impact affects the activity of users in Stack Overflow. Burtch et al. (2023) and Xue et al. (2023) even investigate the impact on users, focusing on the access traffic and the way users deal with Stack Overflow, respectively. In this RQ, we take a different route by investigating the impact of active users in Stack Overflow. First, we compare users' activity level on Stack Overflow pre- and post-release of ChatGPT, considering posting questions, answers, and comments. Second, based on the different groups of users observed (reminiscent and new users), we further explore how users combine the adoption of ChatGPT and Stack Overflow. For that, we manually analyze questions not initially addressed by the LLM and later posted on the web forum. Based on our findings, we discuss the challenges Stack Overflow faces regarding their new users while understanding their new current needs.

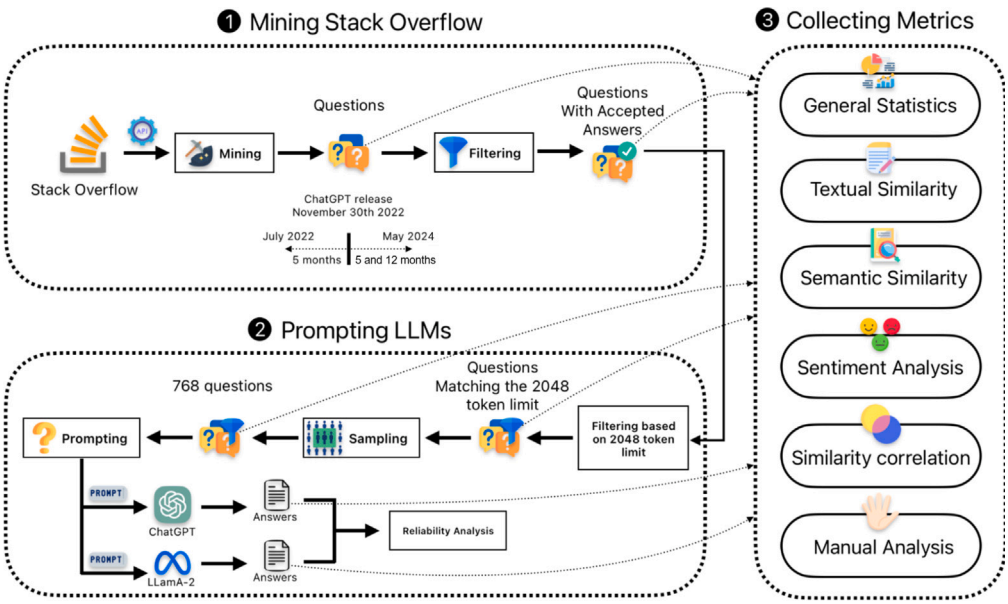


Fig. 1. Overview of our empirical setup.

The remainder of this section describes the empirical setup of our study. We structure our study in three main steps, as presented in Fig. 1: ❶ we mine SO questions using its API from July 2022 till May 2023, and Nov 2023 till May 2024; then, we filter the questions with accepted answers and sort them based on their score. ❷ we select a representative number of questions from this subgroup and use them to prompt ChatGPT and LLaMA-2; and ❸ we further collect metrics regarding question contents and associated information.

3.1. Mining SO data

To investigate the current impact of ChatGPT on Stack Overflow usage, following previous studies' methodological steps (del Rio-Chanona et al., 2023), we mined SO posts and collected the data using SO's API[4] from 5 months before and after the ChatGPT release (1st July till 29th November 2022, and 30th November 2022 till 30th April 2023, respectively). Knowing that adopting the release date of ChatGPT as a cutoff may introduce bias in our results, we also mine posts dated one year after the release (30th November 2023 till 30th April 2024, respectively). This way, we can evaluate the long-term impact, considering immediate reactions and the more sustained engagement over time. This approach ensures a more comprehensive analysis, capturing shifts in discussions, challenges, or resolutions that might emerge only after extended use or broader adoption. From now on, when we say data after ChatGPT's release, we mean the data posted just after the release. To refer to the data posted after one year of the release, we say one-year milestone. Stack Overflow's API limits the returned data for a single request to 50 000 rows. Hence, we performed as many requests as needed to gather all relevant data.[5] For example, to get the `posts` posted between November 29-30th, we used the following query:

```
select * from Posts where Posts.CreationDate
>'2022-11-28 00:00:00' and Posts.CreationDate
<'2022-11-30 00:00:00'
```

[4] https://data.stackexchange.com/stackoverflow/query/new.

[5] Stack Overflow offers its archived dump dataset quarterly. By the time of our mining, we needed up-to-date information, leading us to use the provided API.

Table 1
Sample data description.

	Pre-release	Post-release	One-year milestone
Questions	674 425	578 115	296 718
Questions with answers	424 808	329 412	167 303
Question with accepted answers	212 775	157 187	78 262
Questions without answers	249 617	248 703	129 415
Answers	543 533	425 391	202 326
Comments	1 673 906	1 348 982	718 496
Tags	36 837	37 328	31 867

This way, we make sure the contents posted were created before or after the release of ChatGPT.

Initially, we collected the posts and their related comments (as returned by Stack Overflow' API). However, in Stack Overflow's data, questions and answers are interchangeably treated as regular posts, differentiating from each other based on their type. Furthermore, while questions can receive multiple answers, only one can be targeted as the accepted one; posts can be associated with multiple comments. We collected all information available for each element under investigation, including the creation and last modification date, title, description, score, views, etc. The creation and modification dates are essential information since a question might have been added before the ChatGPT release, but its accepted answer might have been dated after ChatGPT's release. Next, we organize the data by associating the questions with their related answers and comments. This step is required since answers and comments for questions not dated from the interval evaluated here might introduce bias in our results. For example, a question posted before the release of ChatGPT may be answered by a user that used the LLM to address the initial reported issue.

This way, we adopt one particular constraint to investigate the overall impact of posting activity on Stack Overflow. For the questions posted before ChatGPT release, we only consider valid answers and comments, the ones also dated before the release date (30th November 2022). Finally, we were able to collect 674 425, 578 115 and 296 718 questions before, after, and one-year milestone ChatGPT's release, respectively. Table 1 presents an overview of our dataset. Overall, we notice a drop of almost 100 000 between the before and after the release of ChatGPT (for the same amount of time, i.e., five months). One year

after its release, the drop becomes even more pronounced, decreasing by almost threefold (approximately 300 000 questions).

On the other hand, to investigate the impact of questions addressing specific topics (programming languages, frameworks, and libraries), we first properly filtered the questions focusing on these topics. When users ask about these topics, they commonly add the name of the target topic as one of the question tags; for example, `Java`, `Angular`, and `Pandas`. Since multiple tags can be used for a single question, we counted the number of times a single tag was adopted. Then, we organize the frequency of questions associated with each topic daily. Although users may consider variations or abbreviations when targeting their subjects, we observe a consistent adoption of the official topic names; for example, `Pandas` can be abbreviated to `pd`.

Finally, to investigate the impact on the user's activity patterns in Stack Overflow, we evaluate the user information associated with the final set of questions, answers, and comments collected previously. For that, we group the users associated with each content into three groups: *questioners*, *respondents*, and *commentators*. This way, we could compare the number of users involved in each type of activity in Stack Overflow. To answer RQ4, we collected further user information, including their account creation date, allowing us to identify *reminiscent* and *new* users after the release. We request each user's information using their IDs on the Stack Overflow API.

3.2. Prompting LLM models

To investigate LLMs' capabilities to generate high-quality answers compared to humans' answers, we selected questions from our dataset with *accepted answers only* (refer to Table 1). To adhere to the 2048 token limitation for the LLMs under study, questions exceeding this token count were excluded (6998 questions spanning before ChatGPT's release were filtered, and 11 659 after ChatGPT's release). Token requirements were calculated based on the question title, description, and associated tags.[6] As a result, the dataset comprised 205 777 questions before (i.e., 212 775 - 6998) and 145 528 questions after the release of ChatGPT (i.e., 157 187 - 11 659). For subsequent analysis, we randomly selected 384 questions from each set (before and after ChatGPT release), aiming for a 95% confidence level and a 5% margin of error.

This study focuses on two LLMs: ChatGPT 3.5 and LLaMA-2-7b. We used the GPT-3.5-turbo variant for our research, interacting with it through its API. Unlike ChatGPT, we call LLaMA by locally loading it. For our experiments, we used the 7B-sized LLaMA-2 model (`Llama-2-7b-chat-hf` checkpoint in its base configuration), which is the 7-billion-parameter chat-optimized variant, ensuring alignment with conversational tasks. The model was loaded via Hugging Face support,[7] on a machine equipped with 40 GB RAM. When prompting the LLMs, we follow a standardized approach that simulates a prior conversation to establish context. For that, we consider the tags associated with the questions as the foundation for instructing the LLM to adopt the persona of an expert with extensive knowledge in the relevant subject matter (White et al., 2023). Next, we consider the question's title to properly ask the LLM to explain how to fix the problem. Additionally, we argue that further context will be given by informing the question description. Fig. 2 presents the adopted approach with more details. Regarding the usage of these models, we consider their default configurations. For example, we prompt the models with their default temperature and context length (maximum token limit) (Wagner et al., 2024). By making this decision, we aim to avoid arbitrary bias, considering the diversity of tasks addressed in this study. Here, we focus on evaluating the models in similar ways users would, such as respecting their default configurations and interacting with them under typical usage scenarios without additional hyperparameter adjustments. Furthermore, related studies did not report specific hyperparameters, leading us to believe they used default configurations as well. Finally, for each question, we prompt each model once.

Fig. 2. Prompt approach adopted to prompt the LLMs. Question contents were given as input, replacing the elements informed highlighted with ≪ and ≫, respectively.

3.3. Research questions: Evaluation design

This section presents the methodology adopted to address each RQ and the associated metrics adopted when running our study.

3.3.1. RQ1: Comparing human and LLM-generated answers for Stack Overflow questions

Textual Similarity: To assess the similarity between answers from Stack Overflow and those generated by LLMs, we employed the cosine similarity metric (θ) (Salton and Buckley, 1988; Lahitani et al., 2016). To compute this metric, we relied on a pre-trained Sentence Transformers model (`all-MiniLM-L6-v2`).[8] First, we take the accepted and LLM-generated answers for a given question and compute their embeddings. Second, we used the function `cos_sim` from the same model, passing the previous embeddings and computing the metric.

Semantic Similarity: Knowing that the cosine similarity is sensitive and does not compute the semantics between two given answers, we might misclassify similar answers due to low textual similarity. Motivated by previous studies (Liang et al., 2024), we decided to compute the semantic similarity relying on the judgment capability of the LLMs evaluated in this study. Our goal is not to evaluate the reliability of a particular semantic similarity metric, but rather to analyze how the LLMs themselves assess semantic similarity between answers. Specifically, we investigate the degree of consistency in their judgments and how closely they align with human evaluation. First, we simulate a prior conversation to establish the context, using the questions' tags again to instruct the LLM to adopt the persona of an expert. Second, we ask the LLM to compare the original and LLM-generated answers, reporting an evaluation using a VERY LOW to VERY HIGH scale. By combining both answers in one single prompt, we reached the maximum number of tokens by the LLMs; ChatGPT did not classify one answer, while LLaMA did not classify 72 answers due to resource constraints, resulting in 1464 answers that could be analyzed. Third, to evaluate the consistency and level of agreement among different runs for the same model, we prompted ChatGPT to classify the 1535 answers three times. In the end, we randomly selected a subsample of 91 questions from each group (ensuring a 95% confidence level with a 10% margin of error). The primary author manually labeled these questions, following the same scale previously adopted by the LLMs, by comparing the contents of the human-reported and LLM-generated answers. Since the categories adopted present an ordinal

[6] https://pypi.org/project/tiktoken/.
[7] https://huggingface.co/meta-llama/Llama-2-7b.
[8] https://github.com/UKPLab/sentence-transformers.

structure, we must consider such aspect when assessing the level of inter-rater agreement between our manual analysis and the assessments provided by LLMs, and the agreements between the LLMs themselves. As categories have a defined order, the distance between them may not be equal; for example, disagreeing between VERY HIGH and HIGH is less severe than between VERY HIGH and VERY LOW. As a result, we compute Krippendorff's alpha coefficient (Krippendorff, 2011) with an ordinal distance function, as this coefficient handles ordinal weighting by default.

Sentiment Analysis. In light of research suggesting that avoiding negative comments can improve answer acceptance on Stack Overflow (Calefato et al., 2015), we aim to investigate whether LLMs' responses align with this observation. To do so, we perform a sentiment analysis on answers generated by the LLMs using a pre-trained Transformers model (`sentiment-analysis`).[9] Subsequently, we statistically compared the sentiment outcome achieved by each LLM using the Wilcoxon test.

3.3.2. RQ2: Extracting meta information and associated domains from questions

To answer this RQ, we perform a manual analysis focusing on the questions linked to answers reporting low and high textual similarity for each LLM evaluated. For that, we select the questions associated with the first and fourth quartile of each LLM violin plot (low and high similarity, respectively; we explain in detail when discussing our results in Section 4). Since the main focus here is to report the domains that challenge LLMs, we adopt the taxonomy provided by Barua et al. (2014) as a starting point for our categorization. When analyzing the proposed categories, we observe that they were too specific, limiting their application to our sample. This way, we decided to group similar categories into single ones. For example, Barua et al. (2014) report *UI Development*, *Website Design/CSS*, *Web Development*, and *Web Service/Application* as individual categories. For our study, we group all of them into *Web Development*, as they share a similar goal. As a result, we define nine categories: *Database/SQL*, *Framework/Library*, *General Programming Concepts*, *Mobile*, *Networking*, *Operating Systems*, *Programming Languages*, *Tools/IDEs*, and *Web Development*. However, when performing the analysis, we observe that some questions could not be classified using the previous categories. By analyzing them, we additionally report two categories: *DevOps* and *ML/Datascience*. Finally, our analysis is based on 11 categories, nine derived from previous work and two new ones reported by us.

Aiming to get more understanding of these questions, we decided to further explore them based on the meta information we could extract. Initially, we explored some questions to identify the information shared among them. We adopted a simple process, by randomly selecting questions previously prompted to LLMs (no more than 20 questions). As our main goal was to specify the types of information shared by users when posting their questions, when we observed that no new type of information was reported (saturation), we stopped the exploration. As a result, we defined a spreadsheet, incorporating the information we could extract and their predefined values, providing a guide for the manual analysis. Table 2 provides the information we extracted. For this analysis, we focus only on the aspects associated with the questions.

Similarity Correlation. To explore possible factors contributing to the higher reliability of answers generated by LLMs, we extended our analysis to include the context in which the models were prompted. Overall, we performed a correlation analysis between (i) the textual similarity achieved by a given LLM answer, previously computed, and (ii) the number of words associated with the different types of context we have. Since we have three different types of context in our study (tags, title, and description), we adopted distinct strategies for each, as reported below. For tags, we computed the number of tags associated with each question, as this directly reflects the context provided by the tags. For both the title and description, we first computed the number of words. Then, we split them into intervals: a 2-word interval for the title and a 25-word interval for the description. The rationale behind this segmentation was to capture the varying levels of context provided by different text lengths. For titles, a 2-word interval was chosen because we did not observe any questions with only a single word. For descriptions, we opted for 25-word intervals, as there was a consistent rise in the number of questions at each interval length (e.g., 25, 50, etc.). Our goal with this analysis was to quantify the impact of context size on model performance, revealing how the amount of input text influences the quality of the LLM's responses. Since our dataset was not drawn from a normally distributed population, as confirmed by a *Shapiro–Wilk* test, we used the non-parametric Spearman test. Initially, we verify whether the context used to prompt the LLMs correlates with the textual similarity level previously computed (for that, we rely on the Spearman test).

3.3.3. RQ3: Analyzing general and domain-specific impact of posting activity on Stack Overflow

To assess the overall impact of ChatGPT's release on question, answer, and comment frequency, we grouped the frequency of each element by the day they were posted and then performed our analysis. To evaluate the impact on domains, we adopt a similar approach by grouping the questions based on the challenging domains reported in RQ2 (*programming languages*, and *frameworks and libraries*). For each domain, we manually searched for the top 10 most cited tags, which were the same before and after ChatGPT's release. Then, for each tag, we compute the frequency at which they were used daily. Although *Frameworks and Libraries* are classified as one single domain, we explore them individually, selecting the ten most cited frameworks and libraries.

Knowing that our data did not follow a normal distribution, as verified by the Shapiro–Wilk test (Shapiro and Wilk, 1965),[10] we employed the non-parametric Wilcoxon test to compare the distributions of each evaluated group, using a significance level of $\alpha = 0.05$ (Mann and Whitney, 1947).

3.3.4. RQ4: Checking impact on user's activity patterns in Stack Overflow

Based on the number of users active before and after the release of ChatGPT, we first investigate the impact on the different types of users (questioners, respondents, and commentators). Different from the previous RQ, in this one, we focused only on the data before and after release. Knowing that our primary goal is to evaluate the impact of ChatGPT's release on user activity, to ensure a focused comparison, we analyzed data from five months before and up to one year after its release (after and one-year milestone). While we acknowledge that external factors could influence user activity both before and after the release, restricting the pre-ChatGPT period helps minimize historical biases unrelated to ChatGPT, such as long-term platform trends, policy changes, technologies released, or shifts in users' activity patterns. While post-release activity may also be affected by other factors, by adopting a one-year window, we believe we can ensure that we analyze both immediate and sustained changes, making it more likely that observed trends are linked to ChatGPT's introduction rather than unrelated fluctuations.

Moving on, we first compute the IDs of the users associated with the questions and then compare which users were active or inactive after ChatGPT's release. Second, we further compare the active users after release based on the creation date of their accounts. Third, based on the different types of users we observe (reminiscent and new users),

[9] https://github.com/huggingface/transformers.

[10] *p-value* < 0.01.

Table 2
Structural aspects extracted from Stack Overflow questions and associated accepted answers. For a given aspect, we present its definition and possible associated values.

	Structural aspects	Definition	Associated values	Target RQs
Question	Context adequacy	The level of information provided in the question.	Shallow, medium, strong	RQ2, RQ4
	External reference	External content linked with the question	Another SO question, documentation, link	
	No textual content	Adoption of content different than text	Image	
	Illustrative context	Supportive information provided to illustrate the target problem	Code, error message, outcome, expected outcome	
	Goal	The approach adopted to ask for assistance	Question, Explanation, Open	
	Domain	The domain associated with the question	Categories adapted from Barua et al. (2014)	
	Tag concordance	The level of concordance among the informed tags	Shallow, medium, strong	
Answer	No textual content	Adoption of content different than text	Image	RQ4
	External reference	External content linked with the answer	Another SO question, documentation, link	
	Illustrative content	Supportive information provided to illustrate the solution	Code, outcome, expected outcome	

we explore how users appeal for support on Stack Overflow, regarding questions previously asked to ChatGPT. This way, we mine all questions that refer to ChatGPT and its varying terminologies on the question's title and body (`chat-gpt`, `chat gpt`, and `gpt`).[11]

Initially, by analyzing all the 578 115 questions after ChatGPT release (see Table 1), which also include questions without answers, we observe that 2193 questions mention ChatGPT in their contents (title or body). Next, checking the creation date of the questioners' accounts, we observe that 1439 questions were asked by reminiscent users, while 731 questions were asked by new ones. The remaining questions were asked anonymously or by users no longer with active accounts in Stack Overflow, so we could not check whether the questions were associated with reminiscent or new users. For each group, we randomly select a subsample of questions to be manually analyzed (91 and 86 questions, respectively), aiming for a 95% confidence level and a 10% margin of error. Then, we manually analyze each question, extracting the information reported in Table 2. For the current analysis, we focus on the question and the associated accepted answer, as we are also interested in how the respondents would behave when facing a problem that LLMs were previously asked.

4. Empirical findings

This section presents the results of our empirical study.

4.1. How does the reliability of answers generated by ChatGPT and LLaMA-2 compare to those provided by Stack Overflow users?

In our first research question, after selecting Stack Overflow questions and prompting LLMs to address them, we aim to check their reliability based on textual (cosine metric) and semantic similarity.

Textual similarity analysis

Regarding the textual similarity, Fig. 3 shows the distribution of the cosine metric across all prompted questions for both ChatGPT and LLaMA-2. Specifically for ChatGPT, overall, both plots present similar densities (Fig. 3(a), i.e., before and after ChatGPT release), indicating that there is no significant difference regarding the similarity between Stack Overflow and ChatGPT answers. By computing the mean for each time interval, we could validate our initial insight (0.644 and 0.640, pre- and post-release, respectively). Further comparing them, we report no significant statistical difference between both distributions (*Wilcoxon*, *p-value* = 0.59). By reporting such overall similar answers, ChatGPT shows its capability to address tech questions while adopting a similar behavior compared to humans at the textual level. Furthermore, we may highlight that such observation must consider that ChatGPT was trained on publicly available data until September 2021, eliminating possible associated bias.[12]

For LLaMA, the plots also present a similar density, observed by the average reported (Fig. 3(b), 0.616 and 0.619, before and after ChatGPT release, respectively), with a slightly more uniform similarity distribution for the "after ChatGPT release" plot. We also observed no significant statistical difference between both distributions (*Wilcoxon* test, *p-value* = 0.71), indicating that LLaMA-2 presented the same behavior for both sets of questions. Different from ChatGPT, this result might be unexpected considering that, though most of the data used to train LLaMA-2 were dated until September 2022, some tuning data was further required, dated up to July 2023, which might introduce bias in our results due to memorization (Carlini et al., 2022).[13]

Since no significant difference was observed for time interval metrics based on each LLM, we decided to compare the LLMs by aggregating their data. As a result, LLaMA-2 presented an overall lower mean when compared to ChatGPT (0.617 and 0.642, respectively). A *Wilcoxon* test reports that ChatGPT was statistically superior to LLaMA-2 (*p-value* < 0.01) but with a small effect size (Cliff's delta, 0.1). Hence, we conclude that though ChatGPT was statistically superior, LLaMA-2 was close to achieving similar results, representing an alternative, free, and public option for the tech community. To delve into the capabilities of the LLMs evaluated, we discuss some examples as follows.

Before ChatGPT's release. Considering the question with the highest similarity by ChatGPT (θ = 0.92), the user places a question about a programming task in Java, asking about *How to get all keys whose values are null in Java 8 using Map*.[14] An initial code snippet is provided to illustrate the task, presenting a `HashMap` with five `keys` representing colors associated with their HTML codes (two of which are *null*). Figs. 4(a), 4(b), and 4(c) present the answers provided by a human on Stack Overflow, and generated by ChatGPT and LLaMA-2, respectively.

Both Stack Overflow and ChatGPT answers address the user's request but adopt slightly different approaches. Initially, they adopt a `stream` object to process the `HashMap` elements; while the human answer works only with the set of `keys` (line 4), ChatGPT opts to work with the entire `set` (line 4). Next, to check whether the `value` associated with a `key` is `null`, the human answer first requests the value associated with a key to `h` and then checks whether the returned `value` is `null` by calling the method `Objects.isNull` (line 6). ChatGPT directly asks for the `value` associated with each set element, and for the applicable cases, it gets the associated `keys` (line 6). Finally, the selected `keys` are grouped similarly in a `List` for both cases (line 7).

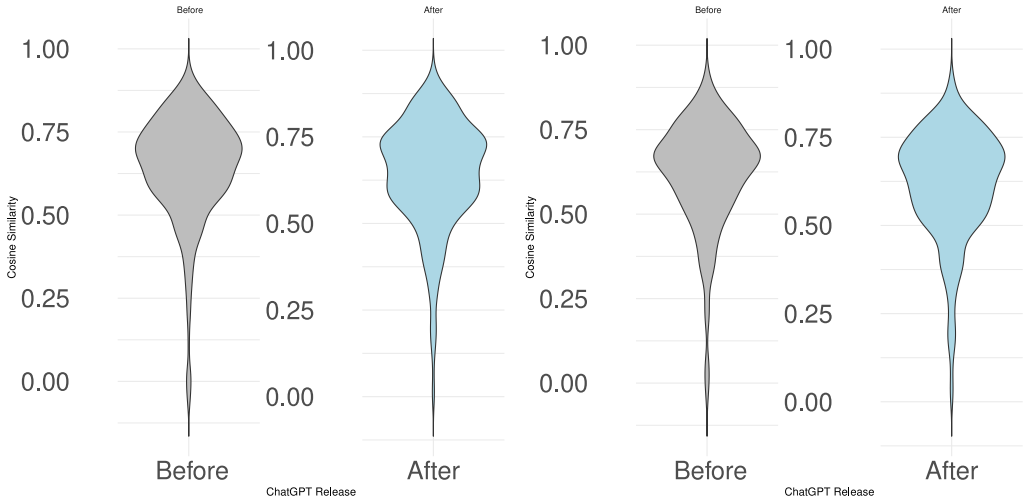
However, LLaMA-2's solution does not work correctly (θ = 0.89). Indeed, the solution returns the `values` associated with the `keys` instead of just the `keys` themselves. It occurs because instead of initially getting the `HashMap` keys, the proposed solution gets the set of `values` (line 4). Then, for each `value`, it is checked whether they are `null` (line 6). For the valid cases, they are stored in the `list` used to save the results (line 7). Such a result shows that though LLaMA-2

[11] We did the same for LLaMA-2, but we did not observe questions targeting it.

[12] https://platform.openai.com/docs/models/gpt-3.5-turbo.

[13] https://github.com/facebookresearch/llama/blob/main/MODEL_CARD.md#training-data.

[14] https://stackoverflow.com/questions/73687017/.



(a) ChatGPT (b) LLaMA

Fig. 3. Distribution of Cosine Similarity computed by each Large Language Model. The metric ranges from −1 to 1, representing dissimilarity and similarity, respectively.

```
        Map<String, String> h;
        h = new HashMap<>();

        List<String> collect = h.keySet()
        .stream()
        .filter(e -> Objects.isNull(h.get(e)))
        .collect(Collectors.toList());

```

(a) Answer generated by a human on SO.

```
        Map <String, String> h;
        h = new HashMap<>();

        List<String> collect = h.entrySet()
        .stream()
        .filter(e -> e.getValue() == null).map(Map.Entry::
getKey)
        .collect(Collectors.toList());

```

(b) Answer generated by ChatGPT.

```
        Map <String, String> h;
        h = new HashMap<>();

        List<String> collect = h.values()
        .stream()
        .filter(e -> e == null)
        .collect(Collectors.toList());

```

(c) Answer generated by LLaMA.

Fig. 4. Three different versions of answers to a Stack Overflow question.

was close to addressing the requested task, the reported answer requires further adjustments. An experienced user could run the code and detect the misbehavior. Next, with a few updates, the code could be fixed and behave as expected.

After ChatGPT's release. Now consider another question with the highest similarity achieved by ChatGPT dated after its release (θ = 0.92). In this example, the user asks for assistance regarding responsive design.[15] For that, the user provides some CSS code, defining different screen and font sizes, and based on screen size, it expects the elements to be adjusted. The Stack Overflow respondent provides four possible solutions for the problem[16]:

> ① *Make sure that the .my-element class is being applied to the correct element in your HTML.* ② *Check that there are no other styles elsewhere in your CSS that might be overriding the font size changes made by the media queries.* ③ *Try adding the !important declaration to the font-size property in each media query.* ④ *Verify that your browser window size is within the range specified.*

ChatGPT's answer overlaps three: ① double-checking whether the expected HTML element is correctly targeting by the specified property; ② possible style conflicts overriding the expected font sizes; and ③ checking whether the browser window size is valid. Furthermore, the Stack Overflow respondent advises the user to use the `!important` declaration, ensuring that property is prioritized over others (style conflicts). ChatGPT adopts a more factual solution, asking the user to check whether the CSS file is correctly linked to the expected HTML file.

LLaMA-2 takes a different route compared to ChatGPT and the original answer, by exploring solutions involving exclusively the code given as input (θ = 0.72). The solution proposed is related to possible properties being overridden by others, which ChatGPT and Stack Overflow answers also briefly discussed. To address the issue, LLaMA-2 suggests replacing the usage of the property `max-width` for `min-width`, or the previous two properties for just `break-point`. These examples show the ability of ChatGPT and LLaMA-2 to address different subjects, with a high similarity when compared to human answers, leading us to observe a good level of reliability.

Semantic similarity analysis

Evaluating reliability relying exclusively on textual similarity might introduce biases in our results, as we briefly discuss when presenting the question addressed by LLaMA-2, which was not properly correct. The opposite scenario can also occur; for example, a user asks for instructions about exporting a fillable textbox PDF file using PowerBI. The accepted answer reports that the required action is not possible to be done.[17] ChatGPT correctly informs the required task is currently impossible to be done. However, the LLM provides further alternatives that might support the user, resulting in a low textual similarity (θ = 0.10). We may conclude that textual similarity represents a good metric for reporting high similarity between the answers but not for reporting low similarity, as false negatives might take place. To address this threat, we also relied on LLMs' judgments to compute the semantic similarity between answers and analyzed the level of agreement in these judgments. Overall, LLaMA-2 reported a more positive evaluation with only 7.4% of the answers with low similarity, whereas ChatGPT reported a frequency of 22.1%. This result goes the opposite of our manual classification, where the frequency of low similarity was around 44% of our sample.

When computing the Krippendorff alpha between ChatGPT and LLaMA-2, we observe a coefficient of 0.02. Although we have used the LLMs with their default configurations, we observe that their ratings are not in good agreement, leading us to conclude that their ratings may not be reliable for the given data. When computing the Krippendorff alpha for multiple runs using ChatGPT, we observed more consistent results but still had a low coefficient (0.34). This result shows that the LLMs, even on different runs of the same model, do not present consistent and stable results regarding judging the semantic similarity of the answers. We believe that the temperature and advanced approaches might play a role in our findings. However, even in such conditions, we might expect some inconsistency when judging semantic similarity. Finally, we also observe the same low similarity when evaluating the agreement of the LLMs with our manual analysis (Krippendorff alpha 0.26 and −0.21, LLaMA and ChatGPT, respectively). As a result, we can conclude that possible conclusions we draw here based on this semantic similarity would be biased by the random behavior of the LLMs (see Section 6).

Sentiment analysis

Regarding the sentiment analysis performed, we observe that most of the answers provided by Stack Overflow users and LLMs present a neutral sentiment associated (96%, 97%, and 79%, Stack Overflow, ChatGPT, and LLaMA, respectively). Unlike other web forums, Stack Overflow users are used to keeping a formal, neutral, and direct way of speaking, which could be misinterpreted as demanding (negative sentiment), especially when listing a sequence of steps to be followed. When comparing the answers, we observe that ChatGPT reports a more neutral sentiment compared to the original answers from Stack Overflow users (*Wilcoxon* test, *p-value* < 0.01). Furthermore, ChatGPT did not report any answer with a negative sentiment, while 13 user answers were classified with negative sentiment. Such a result reinforces the ability of ChatGPT to adapt itself when answering questions, specifically in a tech context evaluated here.

Comparing LLaMA answers with those generated by ChatGPT and Stack Overflow users, we observe a significant statistical difference, revealing that LLaMA-2 report a less neutral sentiment (*Wilcoxon* test, *p-value* < 0.01 and 0.01, respectively). However, unlike the previous answers, LLaMA-2 reports more answers with positive sentiment (20%). These results show that, in its current version, LLaMA-2 consistently adopts a positive-neutral sentiment when prompted by users.

RQ1 answer: Checking the reliability based on textual similarity is statistically consistent. Both before and after ChatGPT's release, the generated answers show a high degree of similarity to the accepted answers on Stack Overflow, as evidenced by the textual similarity (cosine metric). Although textual similarity seems to be a good metric for reporting answers with high similarity, this metric is not applicable for reporting low similarity, as false negatives might occur. Finally, the adopted semantic similarity metric based on LLMs classification also does not present an applicable metric for checking similarity among the answers. Although ChatGPT presents a more neutral sentiment when generating their answers, LLaMA-2 presents a more positive sentiment.

4.2. RQ2: What specific domains pose challenges for ChatGPT and LLaMA-2 when generating reliable and accurate answers?

This research question focuses on understanding the associated challenges faced when generating these answers and the factors that might lead the LLMs to generate more accurate answers. Next, we discuss each aspect individually.

[15] https://stackoverflow.com/questions/75494817/.
[16] https://stackoverflow.com/a/75494849/5141439.
[17] https://stackoverflow.com/questions/74639660/.

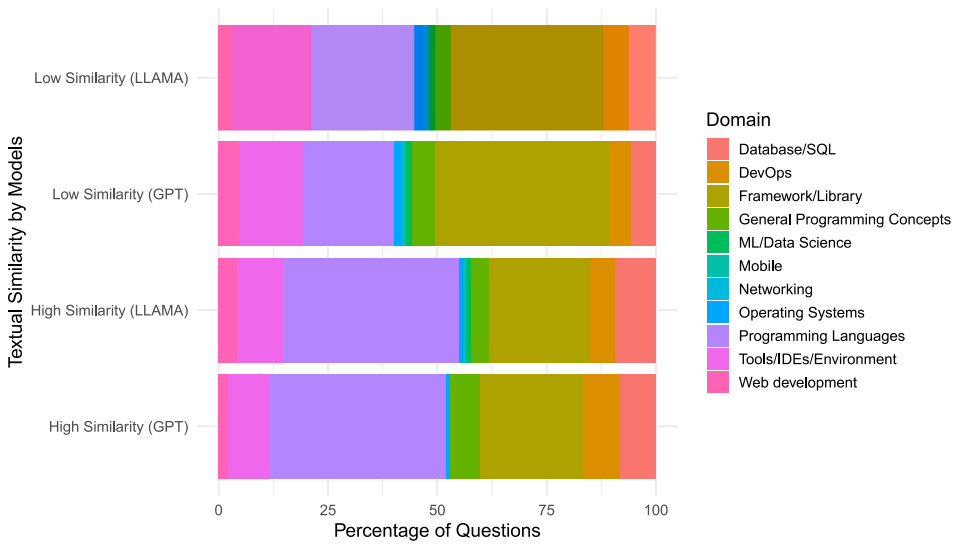


Fig. 5. Distribution of question domains for high- and low-similarity answers generated by LLMs (ChatGPT and LLaMA-2).

Challenges for LLMs

To report the challenges faced by LLMs, we manually analyzed the questions associated with high and low textual similarity (see Section 3.3.2). Regarding the domain of the questions classified with low textual similarity for LLMs (Fig. 5), we reach the same conclusions for both LLMs. *Frameworks and Libraries* represent the most challenging domain for the LLMs evaluated here (39.6% and 34.9%, ChatGPT and LLaMA, respectively), followed by *Programming Languages* (20.8% and 23.4%). When looking at the different groups of questions with the low textual similarity associated with each LLM, we observe that they share 64% of the same questions, revealing that despite the structural differences between the models, they partially share the same limitations when addressing these domains. Furthermore, less than 20% of all generated answers by both models presented a similarity of less than 0.5 (14% and 16%, for ChatGPT and LLaMA, respectively). This small frequency shows that even when the LLMs fail to create a more similar answer, they do not deviate from the original direction of the accepted answers.

On the other hand, for the questions with high textual similarity, the most promising domains are Programming Languages (40.5% and 40.3%), followed by *Frameworks and Libraries* (23.7% and 23%). Although the *Frameworks and Libraries* domain is previously reported as a challenging domain for LLMs, we must discuss how the type of questions associated with each group plays an important role here. When users ask for specific, popular, and standard features of these dependencies, the LLMs achieve a high similarity; however, when the questions address general conventions, standard practices or common knowledge adopted by the community which may not be explicitly documented, or unpopular features about using these external dependencies, a low similarity is observed. For example, a user asks for support when failing to authenticate to an application using valid credentials. The solution accepted guides the user to hash the password when creating a new user in the application[18]:

> *You need to hash the password before creating the user [...] You will need to create the user again since you will create a user with a* ***hashed*** *password to let the authentication succeed.*

In its generated answer, ChatGPT even warns about the user's credential and object creation steps without mentioning the actual cause.

Further exploring the questions, we could observe some structural aspects that might bring some light to our observations. For example, when we look for external references linked to questions regarding framework and dependencies, we observe that the LLMs performed poorly for questions with more external references. While ChatGPT and LLaMA had to deal with 10 and 13 questions with external references when reporting low similarity, they had to deal with only 2 and 4 questions, respectively, when reporting higher similarity. For example, consider a user facing an issue when logging out in a VueJS app using Supertokens for authentication. Even after clicking the Logout button, the session data remained (`userInfo`), and the user still had an active session. Attached to the question, the user also reports a link holding a code snippet provided by SuperTokens.[19] When answering the question, the respondent clarifies that the browser was not saving session cookies because the frontend was running on localhost while the API was on 127.0.0.1, and cross-domain requests require HTTPS for browsers to store cookies properly. Without these extra external details, the LLMs take a different route, mainly guiding the user to check the setup of the environment and sign-out options.

Although the advanced version of ChatGPT can access external links using the web tool (`GPT-4-turbo`), allowing the model to perform searches and open URLs when needed, the evaluated models in this study do not have such a capability. By linking external content, the questions provide a less self-contained context, and consequently, users are expected to read and follow these links for complete understanding. As a result, these questions are more challenging for LLMs to address, as the models must rely solely on the information explicitly provided in the prompt without access to any external supplementary content.

Another structural aspect observed is related to the level of detail provided in the questions. Although both groups provide code as supportive material, we observe that LLMs report higher similarity for questions supported not only by code but also by more supplementary data, like output, expected outcome, and error messages. In this context, for questions with low similarity, we observe ChatGPT and LLaMA dealing with only code for 79% and 74% of the questions, respectively, while for questions with high similarity, the rate was around 60% and 62%, respectively (information collected as previously reported in Table 2). These additional data provide more details about the problem, allowing the LLMs to have a better understanding of the target context.

[18] https://stackoverflow.com/questions/75727923/.

[19] Currently, the provided link is unavailable; however, for each code snippet reported in the documentation, SuperTokens reports an associated explanation.

We observe the same behavior regarding questions addressing programming language tasks when being supported exclusively by code. However, the difference was smaller than the previously discussed domain (for low similarity 78% and 85%, and for high similarity 72% and 76%, for ChatGPT and LLaMA). We believe that programming language questions are more direct as they involve only the aspects of the language itself. For frameworks and libraries, further aspects might be considered, like environment setup and versioning. However, further studies are required to better understand how these aspects could be better handled when prompting LLMs for related questions.

Textual similarity correlation

From all performed analyses, the results show only a statistically significant correlation between the number of tags adopted for a question and the similarity for the answers generated post ChatGPT release (*p-value* < 0.05). The correlation coefficient shows a negative relationship with $\rho = -0.11$, meaning that while one variable increases, the other might decrease. This result indicates that when ChatGPT is prompted to address problems related to one specific tag, it reports more reliable answers (high similarity). While one might expect that providing more information (i.e., multiple tags) would improve reliability, we believe that it can have the opposite effect. For example, the two questions discussed in Section 4.1 only adopt one single tag (`java-8` and `media-queries`, respectively). We believe that by restricting the question to one tag, the LLM can focus its reasoning on a well-defined topic, reducing the risk of generating off-topic responses. However, this negative relationship reveals that ChatGPT might struggle when different contexts are present in the same question, as it has to arrange different topics to generate an answer. When a question includes multiple tags, especially with unrelated concepts, the LLM may struggle to establish a clear reasoning path, potentially affecting the reliability of its answers. For instance, when a user asks for support regarding an unresolved reference, they use three tags, which are all related to each other (`android`, `android-livedata`, and `android-lifecycle`).[20] For this question, ChatGPT generated an answer opposite from the one provided by the Stack Overflow user, achieving a low cosine similarity ($\theta = 0.37$).

RQ2 answer: Although there are structural differences between ChatGPT and LLaMA-2, the LLMs partially share the same challenges. Questions about general conventions or unpopular features of *Framework and Libraries* pose a challenge for the LLMs evaluated here, while *Programming Language* questions are expected to be addressed appropriately. Additional material linked to the questions might play a role in the similarity reached by the LLMs, like code snippets with additional related information. The number of tags associated with a question negatively correlates with textual similarity provided by ChatGPT.

4.3. RQ3: What is the impact of ChatGPT's public release on Stack Overflow posting activity?

Our third research question investigates the impact of ChatGPT's release on the content posted to Stack Overflow. First, we discuss the overall impact of posting and related activities, and then we discuss the impact on specific domains.

Impact on posting activities in Stack Overflow

To address this research question, we first collected the relevant data, further comparing their frequency regarding the number of posted questions, answers, and comments before and after the introduction of ChatGPT (see Table 1). Notably, the number of submitted answers exhibits the most significant decline: 543 533 answers before and 425 391 answers after ChatGPT (i.e., −22%). After the one-year milestone, the decline is even more significant, with a drop of 52% (from 425 391 to 202 326 answers). This decline has a subsequent impact on the number of questions with accepted answers, which dropped from 212 775 before ChatGPT to 157 187 after ChatGPT, and consequently to 78 262 after one year of the release (i.e., −26% and −50%, respectively). Comments and questions follow closely, showing a slightly more significant decline after one year of the release (46% and 48%, respectively).

To gain initial insights into potential trends regarding the decline, Figs. 6 and 7 illustrate the content and temporal evolution of Stack Overflow. Fig. 6 reveals that before ChatGPT's release, there was a higher density of posted content in the upper region of the violin plots. Just after the release, this concentration appears more evenly distributed across the plots, including the middle and lower regions. Finally, after the one-year milestone, we observe a drastic decrease in the concentration of contents. Overall, the median daily count of submitted content is lower after ChatGPT's introduction. Fig. 7 shows the decline in posted content over time. Importantly, oscillating patterns present in the data before ChatGPT's release continue to be observed (they correspond to weekends).

Finally, to statistically assess whether there is a significant difference between the number of posted questions, answers, and comments before, just after, and the one-year milestone of ChatGPT's release, we ran the *Wilcoxon* rank sum test. Our results show we can reject the null hypothesis for all the evaluated contents (*p-value* < 0.01). Regarding the drop of contents after ChatGPT's release, we observe a moderate effect across all evaluated scenarios,[21] reinforcing the findings of previous related studies (del Rio-Chanona et al., 2023; Burtch et al., 2023). Regarding the decline of posted content after the one-year milestone, we observe a large magnitudinal effect, showing a substantial impact on the posting activities,[22] confirming our initial observations reported in Figs. 6 and 7, and bringing evidence about the constant impact in Stack Overflow. While this reflects a significant decrease in posted content, innovations like OverflowAI may influence the shared data volume, as users are driven to consult the tool before publishing new content. This way, our findings cannot ensure a decline in Stack Overflow views, as users might still be accessing Stack Overflow but exploring previously posted content. Further details are discussed in Section 5.

RQ3 answer: After the introduction of ChatGPT, we observe a statistically significant decline in users' activity on Stack Overflow. This is evidenced by the reduced frequency of posted questions, answers, and comments usage, with the most pronounced decline observed after the one-year milestone following ChatGPT's release. Hence, ChatGPT's introduction has impacted the platform's dynamics, likely due to its ability to address many of the queries previously posted on Stack Overflow.

4.3.1. RQ3.1: How are different domains from Stack Overflow impacted by the release of ChatGPT?

To better understand the impact of ChatGPT's release in different domains, based on the domains that represent a challenge for LLMs (RQ2), we aim to statistically compare the number of questions posted in each domain. This way, we explore questions associated with *Programming Languages* and *Frameworks and Libraries*. Fig. 8 presents the trend analysis for the Top-1 tags of each domain evaluated here. Analyzing the charts, we observe that the number of mentions for tags is generally decreasing as time progresses, supporting our previous findings when we observed a general drop in posted content. However, we still aim to investigate each domain individually, as they might suffer different impacts. First, we discuss our findings for programming languages, followed by frameworks and libraries.

[20] https://stackoverflow.com/questions/75449324/.

[21] $r = 0.396, 0.491, 0.463$, for posting questions, answers, and comments, respectively.

[22] $r = 0.745, 0.801, 0.698$, for posting questions, answers, and comments, respectively.

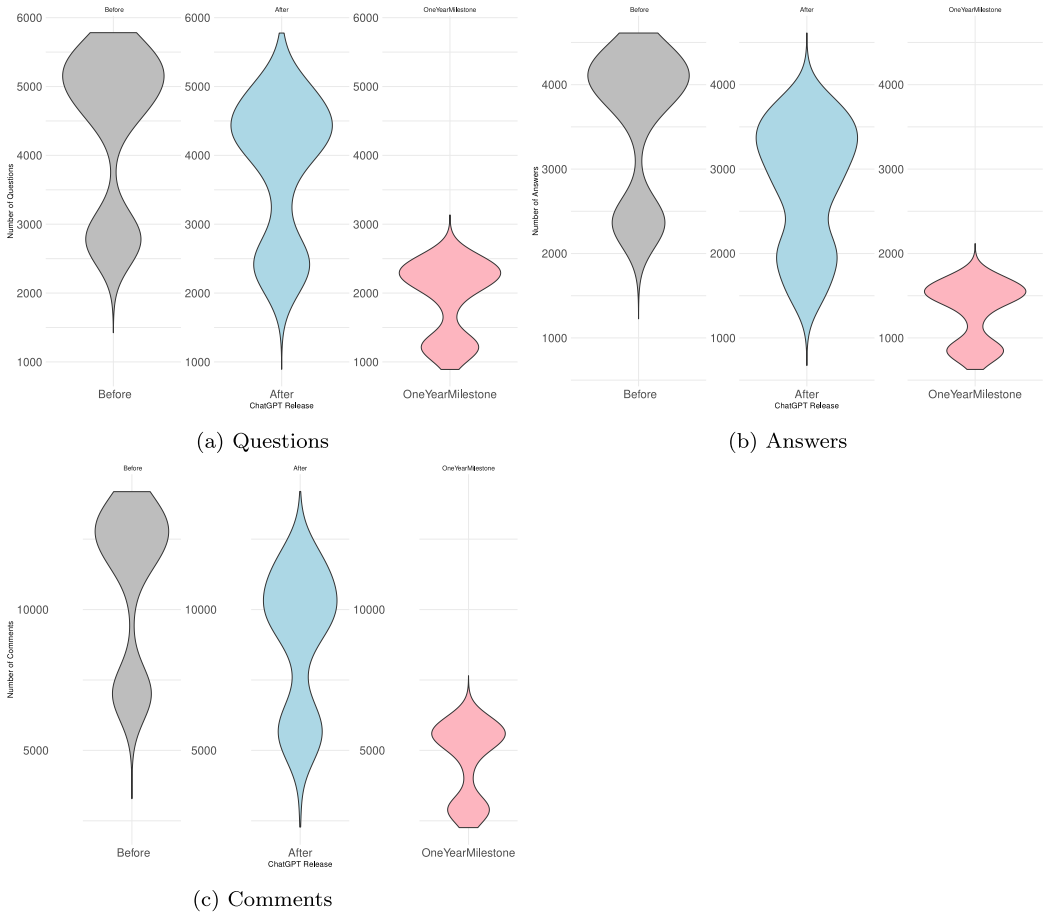


Fig. 6. Violin plots comparing the distribution of the data across three time intervals: Before (July to November 2022), After (November 2023 to May 2023), and One-Year Milestone (November 2023 to May 2024).

Consistent questions' drop associated with the top 10 most cited programming languages

Fig. 9 shows the distribution of questions associated with the top 10 most cited programming languages. Overall, we can observe that `Python` is the most common programming language required by users, followed by `JavaScript`. Such results might be motivated by the continuous adoption of these languages for *Machine Learning* and *Web Development*, respectively. The remaining programming languages present similar frequencies, even oscillating positions with each other, but with a large difference from the top two languages (placed at the bottom of the chart).

Analyzing the chart, we can observe a clear decline in posted questions after the release of ChatGPT for `Python` and `JavaScript` (top 2), the same observation made for general posted content analyzed in RQ3. However, for the remaining programming languages, there is no clear drop as they present a similar frequency of posted questions before and after the release. Statistically assessing whether there is a significant difference between the number of posted questions through the *Wilcoxon* rank sum test, we can reject the null hypothesis for all the evaluated programming languages ($p\text{-}value < 0.01$).

Regarding the effect size, following the previous results reported in this RQ, we observe a moderate effect for most evaluated programming languages (seven out of 10) after the release of ChatGPT. However, when checking the effect size after the one-year milestone, we observe a large effect for all evaluated programming languages, revealing a substantial impact on question tagging these programming languages.

Different impact on most cited frameworks and libraries

Figs. 10 and 11 show the distribution of questions associated with the top ten most cited frameworks and libraries on Stack Overflow, respectively. Different from the chart for programming languages (Fig. 9), we can observe different patterns regarding the distribution of questions here. Regarding the analysis of *frameworks*, we observe that `ReactJS` is the most cited framework, largely isolated from the others. Furthermore, that is the only framework in which we can clearly see a drop in the distribution of questions. Next, we have `NodeJS` and `Android` placed in the second and third positions, respectively. Although they oscillate in their positions, `Android` is reported in the second position for most of the evaluated time. Finally, the remaining frameworks are placed at the bottom of the chart, with some oscillations over time. Different from the programming languages domain, all the frameworks alternate positions except for the first one, showing the ranking of frameworks is more dynamic, with frequent shifts in their relative positions over time. Overall, it is also important to highlight the massive presence of frameworks based on `JavaScript`.

A similar distribution is also observed for the analysis of *libraries*. As highlighted in Fig. 11, `pandas` is the most cited library, placed at the top of the chart with a large distance from the others. As previously discussed for the programming language *Python*, the adoption of `pandas` might be related to their adoption in *Machine Learning*, as it is commonly used for different tasks, like analyzing, cleaning, exploring, and manipulating data. Next, we observe a group of four libraries placed in the middle of the chart, while the remaining ones are placed at the bottom, oscillating positions over time. Finally, similar to the observations made for the *framework* analysis, here we observe a consistent number of libraries supporting different tasks on *Machine Learning* topics, showing the activity of users on these topics.

Aiming to statistically evaluate the impact on the number of posted questions, we also ran the *Wilcoxon* rank sum test. As a result, we

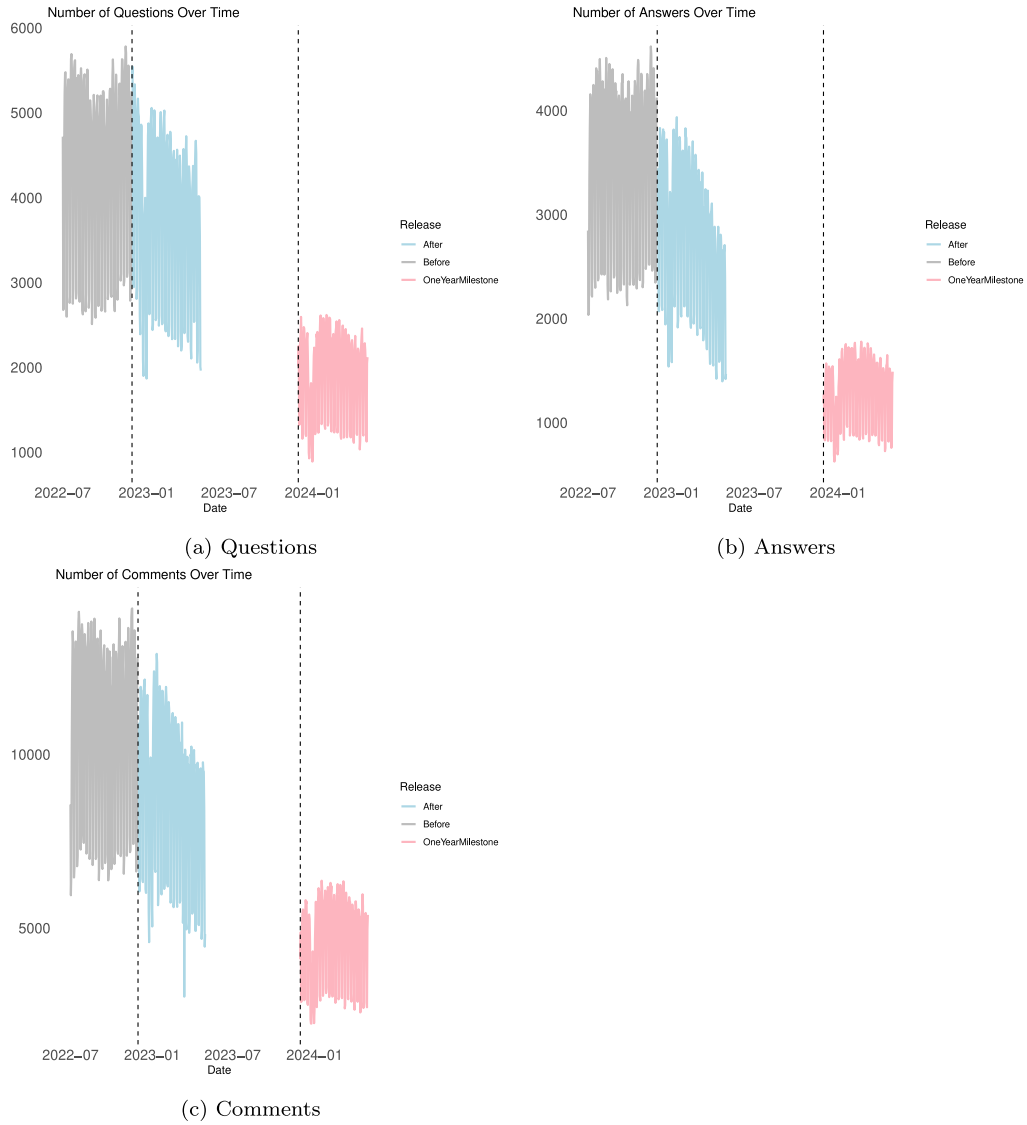
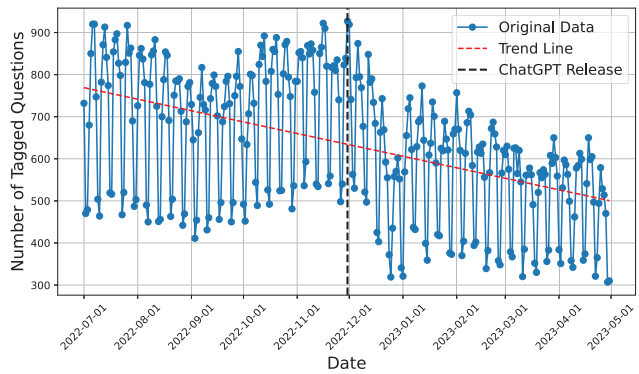


Fig. 7. Distribution of posted content on Stack Overflow (questions, answers, and comments). The gap between the *After* and *One-Year Milestone* intervals represents the period from May to November 2023, during which no data was collected.

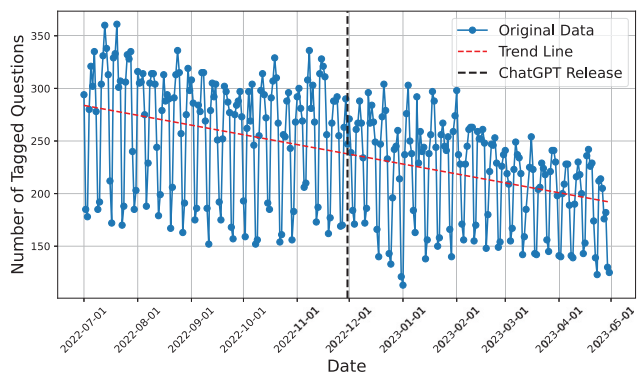
observe different results when compared to our analysis of *programming languages*. Regarding the *frameworks*, we reject the null hypothesis for nine frameworks except for `Spring Boot` (*p-value* = 0.06). A similar behavior was observed for the *libraries* under analysis, as we reject the null hypothesis for all of them except `ggplot2` (*p-value* = 0.34). Although a small fraction of subjects did not present a statistical drop in posted questions, when we analyze the posted content after the one-year milestone, we reject the null hypothesis for all evaluated frameworks and libraries (*p-value* < 0.01). Regarding the effect size, similar to the previous results for programming languages, we observe a moderate effect for most evaluated libraries and frameworks (six out of 10) after the release of ChatGPT. However, when checking the effect size after the one-year milestone, we observe a large effect for all evaluated libraries and frameworks.

On the other hand, we also aim to have an initial insight into domains that exhibited increasing activity over time. For that, we first filtered the tags that appeared in all evaluated time windows of this study (before, after, and one year after ChatGPT's release). As a result, we identified 24 564 commonly shared tags. Next, we examined the number of mentions associated with each tag, selecting those with a consistent increase over time, which resulted in 656 tags (2.68%). These tags span various domains, including new versions of programming languages and frameworks, tools, cloud services, and AI/ML-related technologies. Among these increasing tags, only four exhibited substantial overall activity, each surpassing 1000 mentions: *openai-api*, *vercel*, *powerbi-desktop*, and *next-auth*. These technologies have gained significant attention in recent years, with OpenAI API benefiting from the rise of LLMs, and Vercel and NextAuth growing in popularity alongside Next.js. However, it is important to highlight that the majority of the tags decreased over time, supporting our findings reported here.

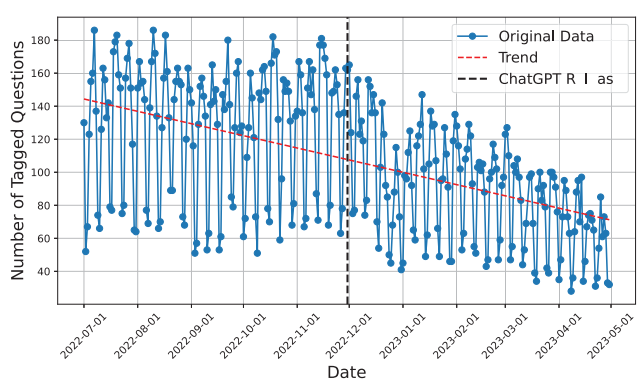
RQ3.1 answer: Despite the observed general drop in posted content in Stack Overflow, initially, after the introduction of ChatGPT, we observe different impacts on the domains analyzed (*programming languages*, *frameworks* and *libraries*). Even analyzing the most cited topics from the previous domains, we observe that, in some cases, there is no statistical decline regarding the volume of posted content. However, after the one-year milestone following ChatGPT's release, we observe a statistically significant drop in posted content regarding all domains analyzed.



(a) Python (Programming Language)



(b) ReactJS (Framework)



(c) Pandas (Library)

Fig. 8. Trend comparison of the number of questions associated with specific tags on Stack Overflow over time. Here, we just present the results for three tags; in our online appendix, we present the trend analysis for all evaluated tags.

4.4. How has the release of ChatGPT impacted the activity patterns of Stack Overflow users?

In this research question, we further investigate the impact of the release of ChatGPT on users' activity on Stack Overflow. First, we discuss the overall impact on the user's activity, and then, we delve into the challenges faced by users when recurring to LLMs to solve their questions.

Overall impact on user's activity patterns in Stack Overflow

Table 3 presents the general observed frequency of users on Stack Overflow. Similar to our previous findings, we observe a general drop in active users (2nd and 3rd rows, respectively, in Table 3). Regardless of the type of user (questioner, and/or respondent, and commentator), all rates present a consistent drop. Such observation corroborates the drop in content posted on Stack Overflow (see Section 4.3). When we compare the number of users active after the release of ChatGPT, based on the different roles analyzed here, we observe that most users were not active before ChatGPT's release(4th row in Table 3). Some may argue that these new active users could be associated with new accounts created after the release of ChatGPT. However, when checking the creation date of these accounts, most of them were created before ChatGPT release (see Table 4). Since different factors might influence these new active users, like sporadic users, it is important to highlight the commitment users have to their community. Such motivation might occur due to new technologies that are released and, consequently, further discussions about how to use and advocate for them.

Although we previously reported a drop in the number of posted questions on Stack Overflow, these active users posting their new questions contribute to new data being generated and, consequently, new versions of LLMs being trained on them. However, keeping users motivated and recruiting new ones to continue the knowledge propagation is a challenge that QA forums must deal with from now on. In Section 5, we further discuss the impact and the need for generating data and LLMs.

New challenges faced by users in Stack Overflow

Once we observe that current active users are new and reminiscent ones, we aim to investigate how these users deal with Stack Overflow combining with LLMs. For that, we investigate whether users replicate questions on Stack Overflow, which were previously asked on ChatGPT, checking whether the users explicitly mentioned that statement in the question content. First, we had to filter the questions asking for clarification about using ChatGPT (13%), like its API (18% and 7% for reminiscent and new users, respectively). Further exploring the remaining questions, though users mention they have consulted ChatGPT for clarifications, 60% of the questions do not present the answers generated by ChatGPT (49% and 70% for reminiscent and new users, respectively). The main reason pointed out by the users is that the answers provided do not fix their problems, as they have tried different proposed suggestions. For example, when a user asks about *constructing an imbalanced dataset based on a pre-defined gini-coefficient*, they report that the ChatGPT could not even generate a working function[23]:

> *I have been working on this problem for a few days now, but my programming and math skills are leaking behind. Also, ChatGPT was not able to construct a working function. If someone has any example code or formula I might use, this would be very helpful. Help is really appreciated!!*

For the users that consider the suggestions provided by ChatGPT somehow useful, they leveraged them as a starting point for further discussions. For example, in one question, a user asks for support when *using the Overpass API*.[24] Based on the initial solution provided, the respondent analyzed it and provided their suggestion, which, based on the respondent, was adequately addressing the initial issue, while the solution provided by ChatGPT was syntactically invalid:

[23] https://stackoverflow.com/questions/76008808/.
[24] https://stackoverflow.com/questions/75558305/.

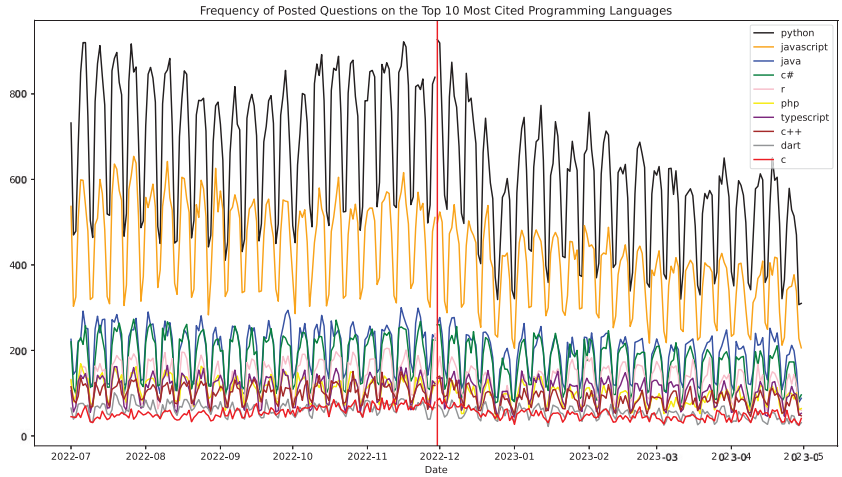


Fig. 9. Distribution of questions associated with the top ten most cited programming languages on Stack Overflow questions.

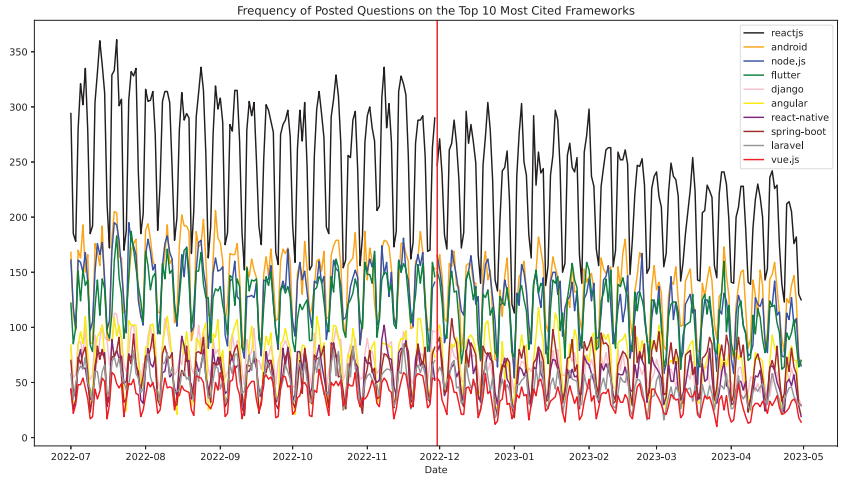


Fig. 10. Distribution of questions associated with the top ten most cited frameworks on Stack Overflow questions.

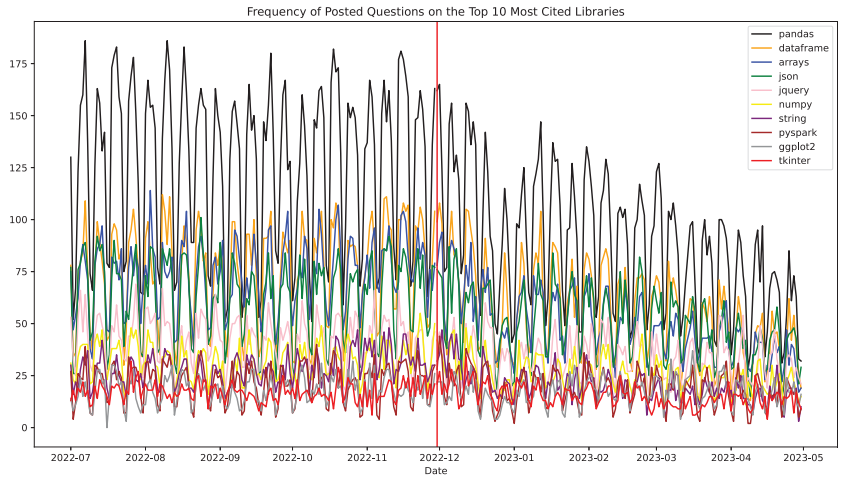


Fig. 11. Distribution of questions associated with the top ten most cited libraries on Stack Overflow questions.

Table 3
User activeness on Stack Overflow. The three first *Active Users* rows are related to users with active history on pre-, post, and one-year milestone of the ChatGPT's release, respectively. The rows for *Inactive*, *Exclusive Active*, and *Reminiscent Users* are calculated by analyzing a specific window along with its preceding windows. For instance, post-release *Inactive Users* are identified as those who were reported as active during the pre-release window.

		Questioners	Respondents	Questioners/Respondents	Commentators
Active users	Pre-Release	314,179	62,007	67,240	239,045
	Post-release	293,811	56,517	57,445	206,530
	One-year milestone	174,097	31,901	30,980	116,830
Inactive users	Post-release	259,981	49,457	56,671	179,300
	One-year milestone	259,207	48,112	50,823	171,771
Excl. active users	Post-release	239,613	43,967	46,876	146,785
	One-year milestone	149,503	25,284	26,489	88,257
Rem. users	Post-Release	54,198	12,550	10,569	59,745
	One-year milestone	24,594	6617	4491	28,573

Table 4
Distribution of users based on account creation date related to ChatGPT release date. We only consider users with active accounts and not in anonymous mode, as we could check their account creation date.

	Questioners	Respondents	Questioners and respondents	Commentators
Accounts Pre-ChatGPT	146,115 (61.1%)	34,943 (79.59%)	38,115 (81.78%)	105,043 (71.72%)
Accounts Post-ChatGPT	93,051 (38.9%)	8961 (20.41%)	8495 (18.22%)	41,420 (28.28%)
	239,166	43,904	46,610	146,463

> *I suggest the following as a starting point, which isn't exactly what you've asked for but way better than what ChatGPT offers as a solution (which is, in fact, invalid Overpass QL syntax).*

In another question, when a user asks about converting `datetime.datetime` objects in timezone-aware `datetime` ones, ChatGPT provides its answer using the `pytz` library.[25] The respondent answers the question by providing code and using another library (`pandas`). Furthermore, the respondent also analyzes the answer provided by ChatGPT and warns the questioner about the deprecation of the `pytz` library. These scenarios highlight how the discussions started on ChatGPT are further extended in Stack Overflow; while questioners use generated answers to give context to their issues, respondents go further by exploring these answers and providing further feedback.

Regarding the topics that represent challenges for Stack Overflow users, we observe that reminiscent and new users face the same challenges when trying to get answers from ChatGPT, and later appeal to Stack Overflow. Similar to RQ2 (see Section 4.2), we observe that questions regarding *Frameworks and Libraries* represent the most recurrent challenge faced by reminiscent and new users (41% and 38%, respectively), followed by question targeting *Programming Languages* (22% and 25%, for reminiscent and new users, respectively). Supporting our claims that these topics currently challenge LLMs, users know these limitations as they experience and recognize them in practice. For example, when a user asks for support regarding starting a new project using multiple frameworks like `Next.JS`, they consider the provided support of ChatGPT was not very helpful. They further argue that the failed ChatGPT's attempt is due to their limited access to information dated until late 2021[26]:

> *To be honest, the first thing I tried when I ran into walls was ChatGPT 4. It helped a little, but as it says in their documentation, it only knows up to late 2021. All the current libraries have changed [...].*

When conducting our analysis, we observe that more than 20% of the questions were removed from Stack Overflow for moderation reasons. We believe these questions were removed as they violated the politics of Stack Overflow regarding the adoption of ChatGPT to address questions. Based on our local mined dump, these removed questions have a reference for the adoption of ChatGPT by their questioners; however, even when they shared the suggestions, they were used to improve the context of the target question. Some may argue that the answers to these questions might have resulted in their exclusion; however, most of these questions even presented associated answers. In the same way, there was no case of questions properly asking users to validate suggestions generated by ChatGPT. Such observations highlight the complexity of removing questions of Stack Overflow only based on reference for ChatGPT usage. We further discuss their impact in Section 5.

RQ4 answer: After the introduction of ChatGPT, we observe a drop in the number of active users in Stack Overflow, even under different roles (questioners, respondents, and commentators). Most active users' post-release were inactive before ChatGPT, showing that they are reminiscent users, as their accounts were created before the release. Finally, reminiscent and new users share the same challenges when addressing their questions on ChatGPT and later appealing to Stack Overflow.

5. Discussion

In this section, we discuss the impact of LLMs on web forums, addressing different challenges from different fields.

LLMs: Impact and future direction

del Rio-Chanona et al. (2023) discuss the ongoing impact of adopting ChatGPT regarding the evolving challenges associated with acquiring the required data for training novel models. Our results corroborate this discussion, as we also observe an overall decrease in the posting activities on Stack Overflow (questions, answers, and comments). However, for some specific domains, we did not observe a statistical drop in posting activities, showing that some communities are active and engaged with their mates. Although different factors might be related to these observations, it is important to understand the motivation behind them. For example, there is limited support for LLMs for addressing more recent versions of frameworks and libraries.

Specifically, regarding the impact on the domains analyzed, as reported in Section 4.3.1, we observed an initial different impact for

[25] https://stackoverflow.com/questions/74879121/.
[26] https://stackoverflow.com/questions/75861694/.

some libraries and frameworks, later resulting in an overall large impact for all domains analyzed. Different factors might contribute to this particular result. No statistical difference initially observed for specific libraries and frameworks might be caused by the limitations of LLMs (the data on which they were trained). LLMs do not properly support tasks that involve information generated after their training without further treatment, although they should be able to generalize knowledge. Another possible reason is related to the fact that different communities associated with these topics might desire to remain active in supporting their members. Even in such cases, though data generation does not represent a challenge, keeping these communities active and supportive might be challenging. However, the advance of new versions of LLMs, with expanded parameter exploration, demonstrates the potential of LLMs to perform tasks better and, as a result, deliver improved performance.

Nevertheless, despite the declining trend in user-generated content, it is plausible that data generation processes may persist, even through different mechanisms, like the usage of data generated by and through LLMs themselves. These alterations would impose specific user requirements to utilize ChatGPT for content creation. Consequently, data generation may still transpire, although, in altered formats, the critical distinction lies in the potential restriction of its accessibility to the broader public.

Replacing Stack Overflow with LLMs

Our study shows that LLMs perform well when addressing general questions (Kashefi and Mukerji, 2023). However, for some specific domains, some challenges still need to be improved for the full and broad adoption of LLMs over Stack Overflow (Ray, 2023). Such challenges are recurrent aspects of computer science, based on the constant development of technologies and their replacements over time. Trying to overcome this situation, we observe users sharing their issues on Stack Overflow, after trying to get answers for them on ChatGPT. In these cases, LLMs could be used as a starting point for users, supporting them in properly reporting their issues (context) and providing initial answers. Later, users could share these initial discussions and even the initial proposed solutions. Finally, together with their community, they could come to a final solution. Such behavior was observed for old and new users in Stack Overflow, showing that this kind of problem is recurrent and requires attention.

For users that opt to rely exclusively on LLMs for addressing their questions, someone may recommend leveraging LLMs to accomplish better support for the required tasks. This way, different techniques can be used, like fine-tuning and merging LLMs. For fine-tuning, the main idea behind this technique is to fine-tune pre-trained models on smaller datasets specific to a particular domain (Kasneci et al., 2023). However, fine-tuning models represents an expensive task as it requires time and the same resources used during the pre-training (Kaddour et al., 2023). This way, fine-tuning models for each new technology or their newer versions is not a feasible approach. Stack Overflow recently announced OverflowAI, a roadmap of practices for the integration of generative AI in the platform.[27] They aim to improve how users interact with its platform and access knowledge. For that, they leverage LLMs and generative AI to support users when searching for content, assist them in knowledge creation, and enhance productivity for developers and technical professionals. Although the main idea is to support users in their daily tasks, and some features are publicly available, to leverage the use of OverflowAI, users might deal with additional costs.

[27] https://stackoverflow.blog/2023/07/27/announcing-overflowai/.

Prompting LLMs for tech questions

When users ask questions on Stack Overflow, they are requested to provide the context of their problems (Galappaththi et al., 2022). For example, while questions regarding programming languages are more prominent to have associated code with and even documentation, questions exploring GUI and setup of tools/environment might adopt non-textual information (Nasehi et al., 2012). Although these non-textual sources provide details that could be hard and time-consuming for users to provide with textual elements, they reflect a limitation for current LLMs (no-textual element support). To overcome this limitation, users must adapt how they prompt LLMs (detailed textual information) (White et al., 2023), while LLMs could also further explore new ways of supporting non-textual information sources (Li et al., 2023b). In the same way, we observe that when LLMs were prompted and given code snippets associated with supplementary data, like error messages and outcomes, they reached higher similarity when compared to the accepted answers. For coding tasks like this, we encourage users to adopt such an approach, as that supplementary data adds important context information that could guide the LLM when addressing the questions.

Our findings show that LLMs perform well when addressing Stack Overflow questions, highlighting their capabilities for programming language tasks. As previously mentioned, supplementary data could be used as input with code snippets. For questions from other domains, there is a gap regarding the supplementary data that could support LLMs. Researchers could conduct further studies to discover and provide these new guidelines. Based on these guidelines, users could better structure their questions and find better support from LLMs. In the same way, we believe further investigation could be performed to explore LLMs addressing unanswered questions in Stack Overflow. For that, the previously mentioned guidelines could be used to also structure the questions given to LLMs.

Stack Overflow users also provide external sources of information when asking or answering questions. For the answers with high textual similarity, 33% of them have associated external information, while 70% provide links for supporting official documentation (Baltes et al., 2020), which is beneficial to the user to double-check the solution or learn more about the problem. LLMs did not report any external link or material when generating their answers, which could be understood as documentation or reference for users. Providing such information might represent a valuable option for the users to reflect on their trust in the provided solution, or even educate them to further investigate the problem by themselves (Uddin et al., 2019; Robillard and DeLine, 2011).

LLMs impact on education

Our results show that LLMs perform well when addressing questions related to generic problems and programming languages. Although users can benefit from using LLMs during programming tasks, they are expected to be able to critically analyze the generated code and, if necessary, make adjustments to improve it (Marsicano et al., 2017). For example, when discussing an answer provided by LLaMA in Section 4.1, we observe the proposed solution does not fix the problem. Although the code could be easily fixed, that is a decision that requires the user to understand the code, locate the bug, and then, apply the required changes (Johnson et al., 2019; Oliveira et al., 2020). Another scenario takes place when LLMs suggest code using deprecated libraries, that could potentially introduce vulnerabilities in the code (Decan et al., 2018) (see Section 4.4). This case requires more attention, as the user must be mature enough to evaluate the impact of handling these vulnerabilities. These issues could be addressed by exploring new features on LLMs, like exploring conventional programming tools to detect general errors or vulnerabilities. Recently, OpenAI released a new feature, Code Interpreter, which allows the ChatGPT to execute Python code (OpenAI, 2023).

The same discussion also occurs when students benefit from these models, by prompting LLMs for different purposes, specifically on programming tasks (Surameery and Shakor, 2023). Given the power of these models, students should use them carefully, not to limit them in their learning process. For example, when answering RQ4, we observe some users posting answers generated by ChatGPT in their questions and explaining the reasons why the generated answers did not resolve their issues. Although this critical thinking is an expected skill for students, they might develop such skills with time through proper learning. Besides the common skills regarding programming, students must also develop new skills to improve their experiences when prompting the LLMs, like prompt engineering and related fields (Strobelt et al., 2022; White et al., 2023).

5.1. Implications

Our findings offer valuable insights into the practical impact of using LLMs as assistant tools in software development. Additionally, they highlight the impact of replacing QA forums with LLMs and the broader implications for future generations of models. Next, we discuss how our results affect different stakeholders.

Implications for Developers: Overall, we believe that Stack Overflow users will continue using LLMs as assistant tools. While these models can generate useful responses, their limitations in certain domains, as reported here, may lead to incorrect conclusions, emphasizing the need for careful validation. While Stack Overflow users are accustomed to reading entire threads and multiple answers (Zhang et al., 2019), LLM users typically receive a direct response without the need for extensive browsing. However, they can engage in a back-and-forth discussion with the model, simulating a one-on-one interaction and challenging its responses.

Implications for Communities in Stack Overflow: With the decline of posting activity in Stack Overflow, communities should foster discussions, encourage peer-reviewed responses, keep users motivated, and actively contribute to knowledge sharing. Knowing that the role of these communities is not restricted to Q&A forums but also extends to maintaining high-quality technical discussions, curating reliable information, and mentoring new contributors, it is crucial to reinforce engagement and sustain a collaborative environment. By doing so, these communities can complement LLM-generated responses, ensuring that software developers continue to have access to accurate and well-reviewed knowledge.

Implications for Model Maintainers/Researchers: As previously discussed, the decline in high-quality data generation will impact the development of new models. Knowing that domain-specific data is essential for improving future models and ensuring equitable access to reliable information, researchers and model maintainers should invest in strategies to sustain and expand high-quality datasets. We believe different strategies can be adopted, like fostering collaborations with developer communities to curate and validate domain-specific knowledge, and encouraging open data-sharing initiatives. Additionally, hybrid approaches that integrate human expertise with LLM-generated content can help maintain knowledge quality and relevance over time.

Understanding these challenges is crucial for both practitioners and researchers seeking to enhance AI-assisted programming. Additionally, our study highlights the role of developer communities in complementing LLM-generated knowledge, reinforcing the importance of peer-driven support. Finally, we discuss the significance of data availability in improving future models and ensuring equitable access to high-quality training data

6. Threats to validity

Our study design introduces particular validity concerns, which we discussed as follows.

Construct to Validity: While we assess the potential of Language Model Models (LLMs) to address questions posted on Stack Overflow, our analysis is limited to determining the similarity between the answers generated by LLMs and the answers labeled as `accepted` by Stack Overflow users. We decided to compare with the accepted answers as they truly align with the specific inquiries posed by the original asker, ensuring a closer correspondence to the requester's intent. It is plausible that the answers generated by LLMs could effectively address the underlying questions, but deviate from the answers marked as accepted, leading to potential misclassification. Even when comparing the generated answers with the entirety of answers available for a given question, there remains a possibility that the generated responses differ from these answers but still offer valuable solutions.

When randomly selecting the group of questions used to answer RQ1 and RQ2, we did not filter out questions that referred ChatGPT or LLaMA. Knowing that this could introduce bias by including questions that directly reference these models, we verified its impact by checking occurrences of these terms and re-running our analysis. Since only two occurrences were found (out of 384 questions) and the results remained unchanged, we believe this issue does not compromise the validity of our findings.

While our data collection includes information from Stack Overflow that postdates the inception of the training data used for ChatGPT, it remains uncertain whether the LLaMA model was also exposed to this more recent data during its training. Consequently, there exists a potential for the outcomes presented herein by LLaMA to be influenced by memorization effects (Carlini et al., 2022), which may introduce a degree of bias into our results. To address this concern, we could apply the same approach used for ChatGPT by filtering the data based on LLaMA's knowledge cutoff. Specifically, LLaMA's initial cutoff in September 2022 suggests that potential bias may have been introduced when collecting data between July and November 2022 (ChatGPT's release). Furthermore, subsequent tuning updates, including those up to July 2023, may have influenced LLaMA's responses, potentially altering its behavior. However, despite this potential threat, our findings show that both models exhibit similar trends, with ChatGPT outperforming LLaMA. This consistency suggests that any memorization effects are unlikely to meaningfully impact our overall conclusions.

Internal Validity: Since we compute the textual similarity using the cosine metric, the potential for misclassification of answers exists due to structural and stylistic disparities relative to the accepted answers. As previously discussed, that metric might lead to false negatives in our analysis. To mitigate this concern, we have incorporated an additional dimension to our analysis by assessing similarity through semantic evaluation by prompting LLMs. We know this method also introduces certain potential limitations, which we endeavor to mitigate by employing a specified similarity scale to standardize and refine semantic similarity assessment. Concerning sentiment analysis, it is noteworthy that the tone and sentiment inherent to the input provided in the prompt can influence the tone assumed by the LLM; however, we prompt each LLM with the same input, trying to eliminate possible associated bias.

When investigating the challenges faced by LLMs to address the questions, we rely on performing manual analysis. To mitigate potential biases, an initial exploration was performed in order to establish the information to be extracted, resulting in a spreadsheet adopted as a guide to the researcher during this step. That spreadsheet encompasses a comprehensive list of labels and their potential values, serving as a guiding reference for the researcher throughout the analytical process, ensuring systematic and rigorous scrutiny. Someone may argue that we could have missed some types of information; however, by verifying the type of information that can be shared when asking a question on Stack

Overflow, we can ensure our analysis covers all the types currently supported.[28]

When evaluating the impact of activity patterns on users in Stack Overflow, our goal was to capture the behavioral impact directly associated with ChatGPT's release. Including data from a more distant pre-ChatGPT period could introduce noise from unrelated historical factors (e.g., long-term platform trends, policy changes, or seasonal patterns), making it harder to isolate ChatGPT's effect. By restricting the analysis to a short pre-release window, we aim to reduce the risk of confounding effects and maintain focus on the most relevant comparison. However, we acknowledge that further studies are required to understand and measure possible co-founding factors associated with ChatGPT's release.

Reliability Validity: For performing our semantic similarity analysis, two different LLMs were used; both of which led to similar conclusions, showing an overall disagreement with our manual analysis. However, when running the same analysis multiple times with ChatGPT, we also observed a low inter-agreement (Krippendorff alpha = 0.34), indicating some level of inconsistency in its responses. This suggests that while another human annotator might introduce further precision, the fundamental conclusions are unlikely to change, given the weak agreement between LLM runs.

External Validity: Our results are limited in the context of technical questions posted on Stack Overflow and the LLMs evaluated here. Although we cannot generalize our findings to other web forum communities, the concerns we have elucidated may have broader applicability to web forums in a more general context. Regarding the LLMs evaluated here, we aimed to explore how different LLMs perform based on our study setup. However, future versions of the same LLMs we evaluate, and other new proposed ones, might unveil nuances and aspects beyond the scope of our current investigation.

7. Related work

LLMs have been explored for different purposes, especially after the release of ChatGPT. In this section, we discuss some related studies, highlighting the differences reported by our study. First, our closest related studies usually investigate correctness, while we investigate reliability, since technically correctness may still be misleading or harmful to users in some contexts. Second, overall, most related studies focused on exploring Stack Overflow under the exclusive perspective of ChatGPT. In this work, we take a different turn by also evaluating LLaMA and further comparing its results with ChatGPT. In the same way, while related work mostly relies on prompting LLMs for specific domains of questions, we adopt a systematic approach to broadly select random questions from different domains and time spaces. This way, besides evaluating the reliability of generated questions, we also explore potential bias resulting from previous training data and the domains in which LLMs struggle to provide support.

Widjojo and Treude (2023) investigate how LLMs can support developers when fixing code errors. To that end, the authors examine 100 code snippets with compiler errors from previous work and new ones generated by them. Next, they explore different combinations of configurations to prompt Stack Overflow and ChatGPT (versions 3.5 and 4) to address each error. The authors report that when prompting ChatGPT with the offending code snippet, the LLM reports contents more instrumental and helpful. Furthermore, they observe ChatGPT 4.0 outperforms its previous version (3.5), as it considers the context associated with the error, while ChatGPT 3.5 focuses on straightforward solutions.

Similar to our goal, Kabir et al. (2023) focus on comparing the answers provided by humans and ChatGPT to Stack Overflow questions. They select a sample of 517 questions and prompt ChatGPT to answer them. The authors investigate the correctness of ChatGPT answers by adopting a manual analysis. They report that 52% of generated answers are incorrect, while 22% are consistent with human answers. Despite the different processes to define their sample and the analysis performed, and based on the computed cosine similarity, our results show better performance of ChatGPT as around 85% of the generated answers presented a good similarity (higher than 0.5).

In the same way, Delile et al. (2023) investigate the usage of ChatGPT-3.5 to address privacy-related questions from Stack Overflow. The authors adopt a different approach by analyzing 82 out of 932 mined questions from a broader time window (2016 to 2023), which could impact their results due to using related training data. Next, they manually compare the generated answers by the LLM with the accepted ones. The authors report that the LLM presented itself as an alternative solution for addressing the Stack Overflow questions. Overall, although the results conform with ours, they do not explore the topics that challenge LLMs. Furthermore, the study has some threats we try to address here, like the memorization (Carlini et al., 2022) and diversity of evaluated LLMs.

Similarly, Oishwee et al. (2024) investigate ChatGPT's ability to answer Stack Overflow questions related to Android permissions. Like us, the authors mine Stack Overflow questions, prompt them to the LLM, and manually evaluate how closely the generated responses match the accepted answers. Since their dataset overlaps with ChatGPT-3.5's training cutoff, potential biases may have influenced the results. The authors report that 53.26% of the generated answers align with accepted Stack Overflow answers, while an additional 27% align with positively voted responses when they do not match the accepted ones, showing the potential of ChatGPT to assist developers in their daily tasks.

Pinto et al. (2023) investigate the usage of ChatGPT to provide feedback to developers. They selected six open-ended questions from two subjects (① caching, and ② stress and performance testing), that were asked for one expert in each subject and a group of 40 developers (questionnaire). Unlike our work, which prompts ChatGPT to answer open-ended questions, the authors prompt the LLM to grade the answers provided by the participants. Additionally, when grading the experts' answers, the authors prompt ChatGPT to correct their answers, resulting in a general consensus among experts concerning the quality and accuracy of the explanations provided.

Liu et al. (2023) investigate the potential of adopting ChatGPT as a code assistant tool for developers. For that, they conducted a study with a group of students, exploring questions related to *general algorithms*, *library*, and *debugging*. This way, the participants were asked to address the selected question, consulting ChatGPT and Stack Overflow. The results show that adopting ChatGPT for answering the *algorithm library* questions outperforms Stack Overflow regarding participants providing correct answers, supporting our findings here.

Previous studies also investigate the impact of ChatGPT on posting activities on Q&A communities. While Xue et al. (2023) focus on the impact of posting questions exclusively in Stack Overflow, del Rio-Chanona et al. (2023) adopt a different approach by considering other popular web forum platforms. While the first study focuses on posted questions, the last explores general posting (questions and answers). Ultimately, both studies collect posting information from all evaluated platforms and propose a model to estimate the effect. Similar to our findings, del Rio-Chanona et al. (2023) observe a significant decrease of 15.6% in the posting activities on Stack Overflow when compared to the other platforms, while Xue et al. (2023) report a significantly negative reduction of the posting by 2.64%. Furthermore, they also report a decrease in the readability of posted questions and cognition level (2.55% and 0.4%, respectively). We go one step further by investigating not only the immediate impact after the release of ChatGPT, but also the impact after one year of its release. Furthermore, we also investigate the impact on specific domains in Stack Overflow, reporting that no statistically significant impact was initially observed for some domains.

[28] https://stackoverflow.com/editing-help.

However, after one year of the release, the decline was constant and significant for all domains evaluated. Similarly, Burtch et al. (2023) also investigate that impact, but not grouping related topics as we do here.

Similar to our study, Burtch et al. (2023) and Xue et al. (2023) also investigate the impact of ChatGPT on Stack Overflow users, focusing on the access traffic and the way users deal with Stack Overflow, respectively. Xue et al. (2023) report that new users are negatively impacted by ChatGPT by asking longer and less readable questions. We take a different perspective by investigating the drop in users' participation in their roles in Stack Overflow (questioner, respondent, and commentator), and the new types of users. Furthermore, we explore the challenges current users face regarding the adoption of ChatGPT and Stack Overflow to address their questions.

8. Conclusion

With the release of different LLMs, especially those targeting different tasks in software engineering, practitioners and researchers have investigated means for ensuring their proper usage. Large Language Models such as ChatGPT and LLaMA are known for their potential to support users in their work, commonly treated as assistant tools. In this work, we investigate the potential of using LLMs to address Stack Overflow questions. We have conducted an empirical study assessing the reliability of the answers generated by these LLMs for existing questions in Stack Overflow while identifying possible challenges.

Overall, our results report that answers generated by ChatGPT and LLaMA-2 show a high degree of textual similarity to the ones accepted in Stack Overflow. Although ChatGPT outperforms LLaMA regarding the textual similarity of generated answers, LLaMA performs well, representing a good, strong, and free option for the general audience. While LLMs perform well on questions associated with general and specific problems on *programming languages*, they face some challenges when addressing *frameworks and libraries* questions. As a result, users might appeal to Stack Overflow when they do not get the expected support for their issues on LLMs. Finally, we also observe a significant decline in user activity on Stack Overflow since the release of ChatGPT (questions, answers, and comments). Initially, for some domains, there is no statistically significant difference regarding the frequency of posted questions, showing that their communities keep active on Stack Overflow. However, after one year of the release, we observed a constant and significant decline for all domains evaluated.

These findings reinforce the capability of LLMs to perform different tasks, shedding light on discussions regarding their impact. Replacing tools, like Stack Overflow with LLMs, represents a huge change that is not prudent to take for now. Currently, LLMs still have to further explore new features to provide generalizable, reliable, and valuable knowledge for their audience, while users are required to develop new skills to improve their experiences when adopting these models.

CRediT authorship contribution statement

Leuson Da Silva: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jordan Samhi:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Data curation, Conceptualization. **Foutse Khomh:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank Arghavan Dakhel for her support in setting up the environment for this study. We also thank the anonymous reviewers for their valuable comments on improving an earlier version of this paper. This work is funded by the following organizations and companies: Fonds de Recherche du Quebec (FRQ), Natural Sciences and Engineering Research Council of Canada (NSERC), the Canadian Institute for Advanced Research (CIFAR), and the Canada Research Chairs Program. However, the findings and opinions expressed in this paper are those of the authors and do not necessarily represent or reflect those organizations/companies.

Data availability

To promote open science and facilitate reproducibility, we make all our artifacts available to the community. This includes the datasets used in our experimentations, the source code for the scripts that were used, the results produced, and any other artifacts related to our study: https://github.com/leusonmario/chat-stack, https://doi.org/10.5281/zenodo.15086541.

References

Asaduzzaman, M., Mashiyat, A.S., Roy, C.K., Schneider, K.A., 2013. Answering questions about unanswered questions of stack overflow. In: 2013 10th Working Conference on Mining Software Repositories. MSR, IEEE, pp. 97–100.

Association, I.S., et al., 1990. Standard glossary of software engineering terminology. IEEE Std 610–612.

Bakker, M., Chadwick, M., Sheahan, H., Tessler, M., Campbell-Gillingham, L., Balaguer, J., McAleese, N., Glaese, A., Aslanides, J., Botvinick, M., et al., 2022. Fine-tuning language models to find agreement among humans with diverse preferences. Adv. Neural Inf. Process. Syst. 35, 38176–38189.

Baltes, S., Treude, C., Robillard, M.P., 2020. Contextual documentation referencing on stack overflow. IEEE Trans. Softw. Eng. 48 (1), 135–149.

Barua, A., Thomas, S.W., Hassan, A.E., 2014. What are developers talking about? An analysis of topics and trends in stack overflow. Empir. Softw. Eng. 19, 619–654.

Blanco, G., Pérez-López, R., Fdez-Riverola, F., Lourenço, A.M.G., 2020. Understanding the social evolution of the java community in stack overflow: A 10-year study of developer interactions. Future Gener. Comput. Syst. 105, 446–454.

Burtch, G., Lee, D., Chen, Z., 2023. The consequences of generative ai for ugc and online community engagement. Available at SSRN 4521754.

Calefato, F., Lanubile, F., Marasciulo, M.C., Novielli, N., 2015. Mining successful answers in stack overflow. In: 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories. IEEE, pp. 430–433.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., Zhang, C., 2022. Quantifying memorization across neural language models. arXiv preprint arXiv:2202.07646.

Dakhel, A.M., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M.C., Jiang, Z.M.J., 2023. Github copilot ai pair programmer: Asset or liability? J. Syst. Softw. 203, 111734.

Decan, A., Mens, T., Constantinou, E., 2018. On the impact of security vulnerabilities in the npm package dependency network. In: Proceedings of the 15th International Conference on Mining Software Repositories. pp. 181–191.

del Rio-Chanona, M., Laurentsyeva, N., Wachs, J., 2023. Are large language models a threat to digital public goods? Evidence from activity on stack overflow. arXiv preprint arXiv:2307.07367.

Delile, Z., Radel, S., Godinez, J., Engstrom, G., Brucker, T., Young, K., Ghanavati, S., 2023. Evaluating privacy questions from stack overflow: Can chatgpt compete? arXiv preprint arXiv:2306.11174.

Dias, K., Borba, P., Barreto, M., 2020. Understanding predictive factors for merge conflicts. Inf. Softw. Technol. 121, 106256.

Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., et al., 2020. Codebert: A pre-trained model for programming and natural languages. arXiv preprint arXiv:2002.08155.

Galappaththi, A., Nadi, S., Treude, C., 2022. Does this apply to me? an empirical study of technical context in stack overflow. In: Proceedings of the 19th International Conference on Mining Software Repositories. pp. 23–34.

GitHub, 2024. Github copilot: Your ai pair programmer. URL https://github.com/features/copilot/.

Goodrich, B., Rao, V., Liu, P.J., Saleh, M., 2019. Assessing the factual accuracy of generated text. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 166–175.

Hämäläinen, P., Tavast, M., Kunnari, A., 2023. Evaluating large language models in generating synthetic hci research data: a case study. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. pp. 1–19.

Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., Wang, H., 2023. Large language models for software engineering: A systematic literature review. arXiv preprint arXiv:2308.10620.

Johnson, J., Lubo, S., Yedla, N., Aponte, J., Sharif, B., 2019. An empirical study assessing source code readability in comprehension. In: 2019 IEEE International Conference on Software Maintenance and Evolution. ICSME, IEEE, pp. 513–523.

Kabir, S., Udo-Imeh, D.N., Kou, B., Zhang, T., 2023. Who answers it better? An in-depth analysis of chatgpt and stack overflow answers to software engineering questions. arXiv preprint arXiv:2308.02312.

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., McHardy, R., 2023. Challenges and applications of large language models. arXiv preprint arXiv:2307.10169.

Kashefi, A., Mukerji, T., 2023. Chatgpt for programming numerical methods. J. Mach. Learn. Model. Comput. 4 (2).

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., et al., 2023. Chatgpt for good? On opportunities and challenges of large language models for education. Learn. Individ. Differ. 103, 102274.

Krippendorff, K., 2011. Computing Krippendorff's alpha-reliability. URL https://repository.upenn.edu/handle/20.500.14332/2089.

Lahitani, A.R., Permanasari, A.E., Setiawan, N.A., 2016. Cosine similarity to determine similarity measure: Study case in online essay assessment. In: 2016 4th International Conference on Cyber and IT Service Management. IEEE, pp. 1–6.

Lee, P., Bubeck, S., Petro, J., 2023. Benefits, limits, and risks of gpt-4 as an ai chatbot for medicine. N. Engl. J. Med. 388 (13), 1233–1239.

Li, R., Allal, L.B., Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., Li, J., Chim, J., et al., 2023a. Starcoder: may the source be with you!. arXiv preprint arXiv:2305.06161.

Li, J., Li, D., Savarese, S., Hoi, S., 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597.

Liang, J.T., Badea, C., Bird, C., DeLine, R., Ford, D., Forsgren, N., Zimmermann, T., 2024. Can gpt-4 replicate empirical software engineering research? In: Proceedings of the ACM on Software Engineering 1. FSE, pp. 1330–1353.

Liu, J., Tang, X., Li, L., Chen, P., Liu, Y., 2023. Which is a better programming assistant? A comparative study between chatgpt and stack overflow. arXiv preprint arXiv:2308.13851.

Lyu, M.R., 2007. Software reliability engineering: A roadmap. In: Future of Software Engineering. FOSE'07, IEEE, pp. 153–170.

Mann, H.B., Whitney, D.R., 1947. On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 18 (1), 50–60. http://dx.doi.org/10.1214/aoms/1177730491.

Marsicano, G., Pereira, D.V., da Silva, F.Q., França, C., 2017. Team maturity in software engineering teams. In: 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM, IEEE, pp. 235–240.

Nasehi, S.M., Sillito, J., Maurer, F., Burns, C., 2012. What makes a good code example?: A study of programming q & a in stackoverflow. In: 2012 28th IEEE International Conference on Software Maintenance. ICSM, IEEE, pp. 25–34.

Oishwee, S.J., Stakhanova, N., Codabux, Z., 2024. Large language model vs. stack overflow in addressing android permission related challenges. In: Proceedings of the 21st International Conference on Mining Software Repositories. pp. 373–383.

Oliveira, D., Bruno, R., Madeiral, F., Castor, F., 2020. Evaluating code readability and legibility: An examination of human-centric studies. In: 2020 IEEE International Conference on Software Maintenance and Evolution. ICSME, IEEE, pp. 348–359.

Online Appendix, 2025. Link. URL https://github.com/leusonmario/chat-stack.

OpenAI, 2023. Code interpreter. URL https://openai.com/blog/chatgpt-plugins#code-interpreter.

Orosz, G., 2025. Stack overflow is dead, almost, the pragmatic engineer blog. URL https://blog.pragmaticengineer.com/stack-overflow-is-almost-dead.

Ozkaya, I., 2023. Application of large language models to software engineering tasks: Opportunities, risks, and implications. IEEE Softw. 40 (3), 4–8.

Pinto, G., Cardoso-Pereira, I., Monteiro, D., Lucena, D., Souza, A., Gama, K., 2023. Large language models for education: Grading open-ended questions using chatgpt. In: Proceedings of the XXXVII Brazilian Symposium on Software Engineering. pp. 293–302.

Ragkhitwetsagul, C., Krinke, J., Paixao, M., Bianco, G., Oliveto, R., 2019. Toxic code snippets on stack overflow. IEEE Trans. Softw. Eng. 47 (3), 560–581.

Ray, P.P., 2023. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet Things Cyber-Phys. Syst..

Robillard, M.P., DeLine, R., 2011. A field study of api learning obstacles. Empir. Softw. Eng. 16, 703–732.

Rubei, R., Di Sipio, C., Nguyen, P.T., Di Rocco, J., Di Ruscio, D., 2020. Postfinder: Mining stack overflow posts to support software developers. Inf. Softw. Technol. 127, 106367.

Sadasivan, V.S., Kumar, A., Balasubramanian, S., Wang, W., Feizi, S., 2023. Can ai-generated text be reliably detected? arXiv preprint arXiv:2303.11156.

Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24 (5), 513–523.

Shapiro, S.S., Wilk, M.B., 1965. An analysis of variance test for normality (complete samples). Biometrika 52 (3/4), 591–611.

Squire, M., 2015. Should we move to stack overflow? Measuring the utility of social media for developer support. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, vol. 2. IEEE, pp. 219–228.

StackOverflow, 2023a. Announcing overflowai. URL https://stackoverflow.blog/2023/07/27/announcing-overflowai/.

StackOverflow, 2023b. Temporary policy: Generative ai (e.g. chatgpt) is banned. URL https://meta.stackoverflow.com/questions/421831/temporary-policy-generative-ai-e-g-chatgpt-is-banned.

Strobelt, H., Webson, A., Sanh, V., Hoover, B., Beyer, J., Pfister, H., Rush, A.M., 2022. Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. IEEE Trans. Vis. Comput. Graphics 29 (1), 1146–1156.

Surameery, N.M.S., Shakor, M.Y., 2023. Use chat gpt to solve programming bugs. Int. J. Inf. Technol. Comput. Eng. (IJITC) (ISSN: 2455-5290) 3 (01), 17–22.

Syam, G., Lal, S., Chen, T., 2023. Empirical study of the evolution of python questions on stack overflow. e-Inform. Softw. Eng. J. 17 (1).

Tamburri, D.A., Kruchten, P., Lago, P., van Vliet, H., 2013. What is social debt in software engineering? In: 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering. CHASE, IEEE, pp. 93–96.

Tang, R., Chuang, Y.-N., Hu, X., 2023. The science of detecting llm-generated texts. arXiv preprint arXiv:2303.07205.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Uddin, G., Baysal, O., Guerrouj, L., Khomh, F., 2019. Understanding how and why developers seek and analyze api-related opinions. IEEE Trans. Softw. Eng. 47 (4), 694–735.

Verdi, M., Sami, A., Akhondali, J., Khomh, F., Uddin, G., Motlagh, A.K., 2020. An empirical study of c++ vulnerabilities in crowd-sourced code examples. IEEE Trans. Softw. Eng. 48 (5), 1497–1514.

Wagner, S., Barón, M.M., Falessi, D., Baltes, S., 2024. Towards evaluation guidelines for empirical studies involving llms. arXiv preprint arXiv:2411.07668.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C., 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382.

Widjojo, P., Treude, C., 2023. Addressing compiler errors: Stack overflow or large language models? arXiv preprint arXiv:2307.10793.

Xia, X., Bao, L., Lo, D., Kochhar, P.S., Hassan, A.E., Xing, Z., 2017. What do developers search for on the web? Empir. Softw. Eng. 22, 3149–3185.

Xu, F.F., Alon, U., Neubig, G., Hellendoorn, V.J., 2022. A systematic evaluation of large language models of code. In: Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming. pp. 1–10.

Xue, J., Wang, L., Zheng, J., Li, Y., Tan, Y., 2023. Can chatgpt kill user-generated q & a platforms? Available at SSRN 4448938.

Yazdaninia, M., Lo, D., Sami, A., 2021. Characterization and prediction of questions without accepted answers on stack overflow. In: 2021 IEEE/ACM 29th International Conference on Program Comprehension. ICPC, IEEE, pp. 59–70.

Yli-Huumo, J., Maglyas, A., Smolander, K., 2016. How do software development teams manage technical debt?–An empirical study. J. Syst. Softw. 120, 195–218.

Zhang, T., Upadhyaya, G., Reinhardt, A., Rajan, H., Kim, M., 2018. Are code examples on an online q & a forum reliable? A study of api misuse on stack overflow. In: Proceedings of the 40th International Conference on Software Engineering. pp. 886–896.

Zhang, H., Wang, S., Chen, T.-H., Hassan, A.E., 2019. Reading answers on stack overflow: Not enough! IEEE Trans. Softw. Eng. 47 (11), 2520–2533.

Zheng, Z., Ning, K., Chen, J., Wang, Y., Chen, W., Guo, L., Wang, W., 2023. Towards an understanding of large language models in software engineering tasks. arXiv preprint arXiv:2308.11396.