Who Answers It Better? An In-Depth Analysis of ChatGPT and Stack Overflow Answers to Software Engineering Questions

Samia Kabir
Purdue University
West Lafayette, USA
kabirs@purdue.edu

David N. Udo-Imeh
Purdue University
West Lafayette, USA
dudoimeh@purdue.edu

Bonan Kou
Purdue University
West Lafayette, USA
koub@purdue.edu

Tianyi Zhang
Purdue University
West Lafayette, USA
tianyi@purdue.edu

ABSTRACT

Over the last decade, Q&A platforms have played a crucial role in how programmers seek help online. The emergence of ChatGPT, however, is causing a shift in this pattern. Despite ChatGPT's popularity, there hasn't been a thorough investigation into the quality and usability of its responses to software engineering queries. To address this gap, we undertook a comprehensive analysis of ChatGPT's replies to 517 questions from Stack Overflow (SO). We assessed the correctness, consistency, comprehensiveness, and conciseness of these responses. Additionally, we conducted an extensive linguistic analysis and a user study to gain insights into the linguistic and human aspects of ChatGPT's answers. Our examination revealed that 52% of ChatGPT's answers contain inaccuracies and 77% are verbose. Nevertheless, users still prefer ChatGPT's responses 39.34% of the time due to their comprehensiveness and articulate language style. These findings underscore the need for meticulous error correction in ChatGPT while also raising awareness among users about the potential risks associated with seemingly accurate answers.

CCS CONCEPTS

• **Software and its engineering**; • **General and reference** → **Empirical studies**;

KEYWORDS

stack overflow, q&a, large language model, chatgpt

ACM Reference Format:
Samia Kabir, David N. Udo-Imeh, Bonan Kou, and Tianyi Zhang. 2018. Who Answers It Better? An In-Depth Analysis of ChatGPT and Stack Overflow Answers to Software Engineering Questions . In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Software developers often resort to online resources for a variety of software engineering tasks, e.g., API learning, bug fixing, comprehension of code or concepts, etc. [53, 57, 63]. A vast majority of these help-seeking activities include frequent engagement with community Q&A platforms such as Stack Overflow[1](SO) [52, 53, 62, 63] to seek help, solutions, or suggestions from other developers.

The emergence of *Large Language Models (LLMs)* has demonstrated the potential to transform the web help-seeking patterns of software developers. Recent studies show that programmers utilize AI tools such as GitHub Copilot [27] for faster exploration and queries of problems at hand and turn to web searches or SO only when they need to verify a solution or access the documentation [6, 55, 61]. The ability to engage in interactive conversations and provide apt solutions using natural language has propelled *LLM*s into becoming a popular option among programmers.

In continuation of *LLM* progress, in November 2022, ChatGPT [44] was introduced as an open-access Chatbot, which surpassed the popularity of other models in its category. ChatGPT's capacity to engage in human-like conversations, promptly learn from continuous human feedback, and accessibility to the general public, have all contributed to its popularity. Consequently, ChatGPT's popularity has ignited numerous debates among academics, researchers, and industry professionals on Twitter and other social media platforms [17, 51]. These discussions revolve around the scenarios wherein ChatGPT could potentially replace prominent search engines (e.g., Google) or widely used Q&A platforms (e.g., Stack Overflow).

Despite the increasing popularity of ChatGPT, concerns surrounding its nature as a generative model and the associated risks remain prevalent. Previous studies [8, 24, 26, 28, 41] show that *LLM*s can acquire and propagate factually incorrect knowledge, which can persist in their generated or summarized texts. Additionally, *LLM*s often generate fabricated texts that mimic truthful information, thereby introducing risk for non-expert end-users who lack the means to verify factual inconsistencies [11, 16, 21]. Recent studies show that ChatGPT is also plagued with these issues [12, 30, 36, 43]. The prevalence of misinformation, which can easily mislead users, has prompted Stack Overflow to impose a ban on posting answers generated by ChatGPT [46].

[1]https://stackoverflow.com/

A few recent studies have compared ChatGPT to human experts in legal, medical, and financial domains [30], or analyzed the quality of texts summarized by ChatGPT [25]. To the best of our knowledge, no comprehensive investigation has been conducted to compare ChatGPT with human answers in the field of Software Engineering (SE). Moreover, there is no investigation into the factors contributing to ChatGPT's answer quality and characteristics. In this work, we aim to address this research gap by adopting a mixed-methods research design [35]– a combination of comprehensive manual analysis, linguistic analysis, and a user study to compare human answers and ChatGPT's answers to Stack Overflow questions.

Specifically, we performed stratified sampling to collect ChatGPT's answers to 517 SO questions with different characteristics (e.g., views, question types, etc.). The sample size is statistically significant with a 95% confidence level and 5% margin of error. We manually analyzed ChatGPT's answers and compared them with the accepted SO answers written by human programmers. For a comprehensive evaluation, we extended our analysis beyond assessing the correctness of the answers and employed additional labels to gauge their consistency, comprehensiveness, and conciseness. Our results show that 52% of ChatGPT-generated answers are incorrect and 62% of the answers are more verbose than human answers. Furthermore, about 78% of the answers suffer from different degrees of inconsistency to human answers.

Furthermore, to examine how the linguistic features of ChatGPT-generated answers differ from human answers, we conducted an in-depth linguistic analysis by performing Linguistic Inquiry and Word Count (LIWC) [49] and sentiment analysis on ChatGPT answers and human answers to 2000 randomly sampled SO questions. Our results show that ChatGPT uses more formal and analytical language, and portrays less negative sentiment.

Finally, to capture how different characteristics of the answers influence programmers' preferences between ChatGPT and SO, we conducted a user study with 12 programmers. The study results show that participants' overall preferences, correctness ratings, and quality ratings are more leaned toward SO. However, participants still preferred ChatGPT-generated answers 39% of the time. When asked why they preferred ChatGPT answers even when they were incorrect, participants suggested the comprehensiveness and articulated language structures of the answers to be some reason for their preference.

Our manual analysis, linguistic analysis, and user study collectively demonstrated that while ChatGPT performs remarkably well in many cases, it frequently makes errors and unnecessarily prolongs its responses. However, ChatGPT-generated answers have richer linguistic features which causes users to exhibit a preference for ChatGPT-generated answers and overlook the underlying incorrectness and inconsistencies. Our in-depth analysis points towards a few reasons behind the popularity of ChatGPT and also highlights several research opportunities in the future.

To conclude, this paper makes the following contributions:

- We conducted an in-depth analysis of the correctness and quality of ChatGPT's answers across 4 distinct categories for various types of SO question posts.
- We performed a large-scale analysis of the linguistic characteristics of ChatGPT's answers and compiled an extensive list of distinguishing features that are prominent in ChatGPT's answers.
- We investigated how real programmers consider answer correctness, quality, and linguistic features when choosing between ChatGPT and Stack Overflow (SO) through a within-subjects user study. [2]

The rest of the paper is organized as follows. Section 2 describes the research questions that will help to investigate the characteristics of ChatGPT answers. Section 3 describes the data collection process followed by the methodology of our mixed methods study in three consecutive subsections. Section 4, 5, and 6 describes the analyses results of our manual analysis, linguistic analysis, and user study respectively. Section 7 discusses the implications of our findings and future research directions. Section 8 describes the internal and external threats to validity. Section 9 describes the related work. Section 10 concludes this work.

2 RESEARCH QUESTIONS

The main objective of this work is to study different characteristics of ChatGPT answers in comparison with human-written answers. We investigated the following research questions and provided the motivation for each question below.

- *RQ1. How do ChatGPT answers differ from SO answers in terms of correctness and quality?* Previous work [16, 32, 36] have shown that *LLM*s such as ChatGPT are prone to hallucination and suffer from low quality. Therefore, we want to assess the correctness and other quality issues of ChatGPT's answers (e.g., consistency, conciseness, comprehensiveness) in the context of SE. We expect that with manual analysis, we can empirically study the correctness and quality factors of ChatGPT answers.
- *RQ2. What kinds of quality issues do ChatGPT answer have?* While RQ1 investigates the overall correctness and quality of ChatGPT answers, RQ2 aims to obtain a deep understanding about the types of issues and develop a taxonomy. For example, some parts of an answer can be factually incorrect, and some parts can be irrelevant or redundant. To answer this RQ, we conduct an in-depth analysis of ChatGPT answers to identify these fine-grained quality issues in ChatGPT answers.
- *RQ3. Do the types of SO questions affect the quality of ChatGPT answers?* Previous studies [2, 37] show that linguistic forms of SO answers vary based on the types of SO questions. For example, *How-to* questions have step-by-step answers, *conceptual* questions contain descriptions and definitions, etc. We seek to understand if the types of SO questions influence the characteristics of ChatGPT answers in a similar manner.
- *RQ4. Do the language structure and attributes of ChatGPT answers differ from SO answers?* Previous studies [65] show that human and machine-generated misinformation has distinctively different linguistic features that can aid in identifying misinformation. Prior work [7] has also shown a relationship between the success of Stack Overflow answers and linguistic

[2] We have made our Dataset and Codebooks publicly available at https://github.com/SamiaKabir/ChatGPT-Answers-to-SO-questions to foster future research in this direction.

characteristics. We aim to find the distinct linguistic characteristics of ChatGPT answers and how they compare to accepted SO answers written by human developers.

- *RQ5. Do the underlying sentiment of ChatGPT answers differ from SO answers?* Previous studies [42, 50] discuss the harmful effect of toxicity or negative tone in open source discussions. Previous work [15] also shows the role of underlying sentiment in the success of SO answers. In this work, we seek to find how the underlying positive and negative sentiment of ChatGPT's answers to SO questions compares to accepted SO answers.
- *RQ6. Can users differentiate ChatGPT answers from human answers?* We are curious about where users can discern machine-generated answers from human-written answers and what kinds of heuristics they employ to make the decision. Investigating these heuristics is important, since it helps identify good practices that can be adopted by the developer community and also helps inform the design of automated mechanisms. We also aim to determine if these findings align with findings from manual and linguistic analysis.
- *RQ7. Can users identify misinformation in ChatGPT answers?* Understanding how users identify misinformation is important as it can provide developers with a quick way to determine misinformation in machine-generated answers. If the users can or can not identify the misinformation properly, we expect to find their major tactics or impediments.
- *RQ8. Do users prefer ChatGPT over SO based on correctness, quality, or linguistic characteristics?* Finally, we want to understand the user preference between ChatGPT and human-generated answers based on the correctness, quality, and linguistic characteristics of the answer.

3 METHODOLOGY

To answer the research questions in Section 2, we conducted a mixed-methods study. To answer RQ1 to RQ3, we conducted an in-depth manual analysis through open coding (Section 3.2). To answer RQ4 and RQ5, we conducted a linguistic analysis and a sentiment analysis (Section 3.3). And finally, to address RQ6 to RQ8, we designed and conducted a user study and semi-structured interview (Section 3.4). The following sections provide a detailed description of each method.

3.1 Data Collection

3.1.1 ***SO Question Collection.*** To capture the different characteristics of SO questions, we consider three aspects of the questions—the question popularity, the time when the question was first posted, and the question type. To ensure that the human-written answers have good quality, we only consider SO questions that have an accepted answer post. We adopted a stratified sampling strategy to collect a balanced set of SO questions that fall into different categories w.r.t. their popularity, posting time, and question type.

First, we collected all questions in the SO data dump (March 2023) and ranked them by their view counts. We used view counts as the measurement for the popularity of SO questions. We selected three categories of questions—the top 10% of questions in the view count ranking (*Highly Popular*), the questions in the middle (*Average Popular*, and the bottom 10% in the ranking (*Unpopular*).

Characteristics	Category	Criteria	# of Q.
Type	Conceptual	_	175
	How-to	_	170
	Debugging	_	172
Popularity	Popular	Highest 10% View Count (Avg. 28750.5)	179
	Average Popular	Average View Count (Avg. 905.3)	165
	Unpopular	Lowest 10% View Count (Avg. 42.1)	173
Recency	Old	Before November 30, 2022	266
	New	After November 30, 2022	251

Table 1: Characteristics, Categories, and Criteria of SO Questions posts

Second, from the three categories of questions above, we moved on to categorize them by their recency. We split questions in each popularity category into two recency categories—questions posted before the release of ChatGPT (November 30, 2022) as *Old*, and questions posted after that time as *New* question posts. We selected the release date of ChatGPT to evaluate how the answer characteristics of ChatGPT reflect the presence or absence of specific knowledge in ChatGPT's training data.

Third, for question types, based on the literature [2, 19, 38, 60], we focused on three common questions types—*Conceptual, How-to*, and *Debugging*. We followed prior work [34, 37] and trained a Support Vector Machine (SVM) classifier to predict the type of a SO question based on the question title. The classifier achieves an accuracy of 78%, which is comparable to prior work. Then, we used this classifier to predict the question type of SO questions in each category of questions obtained from the two previous steps.

In the end, we randomly sampled the same number of questions from each category along the three aspects. Given the question type classifier may not be accurate, we manually validated the question type of each sample and discarded those with wrong types. We ended up with 517 sampled questions. Table 1 shows the distribution of these 517 questions along the three aspects.

Additionally, we randomly selected another set of 2000 questions from the SO data dump for linguistic analysis. Since all collected questions are in HTML format, we removed HTML tags and stored them as natural language descriptions along with their metadata (e.g., tags, view count, types, etc.) in CSV files.

3.1.2 ***ChatGPT Answer Collection.*** For each of the 517 SO questions, the first two authors manually used the SO question's title, body, and tags to form one question prompt[3] and fed that to the Chat Interface [45] of ChatGPT. The generated answers by ChatGPT are then stored in CSV files. For the additional 2000 SO questions, ChatGPT 3.5 Turbo API is used. The question title, body, and tags are extracted and concatenated to query the ChatGPT, and the answers are stored in CSV files.

3.2 Manual Analysis

In this section, we discuss the methodology for investigating the correctness and other qualities of ChatGPT answers to SO questions through Open Coding [31].

3.2.1 ***Open Coding Procedure.*** To assess the quality and correctness of ChatGPT's answers to SO questions (RQ1), we used a standard NLP data labeling process [54, 64] to label the ChatGPT

[3]Example prompts are included in the Supplementary Material

answers at the sentence level. Over the course of 5 weeks, three authors met 6 times to generate, refine, and finalize the codebooks to annotate the ChatGPT answers. First, the first two authors familiarized themselves with the data. Each author independently labeled 5 ChatGPT answers at the sentence level and took notes about their observations. The two authors met to review their labeling notes and implemented deductive thematic analysis [13, 29] to come up with 4 broader themes that they want to assess—*Correctness*, *Consistency*, *Comprehensiveness*, *Conciseness*.

In the next step, they revisited the previous 5 ChatGPT answers to generate the initial codebook spanning the 4 assessment themes. In the 2nd meeting, the first two authors met another co-author to resolve the disagreements and refine the codebook. After this step, the refined codebook contained 24 codes in the 4 categories.

With the codebook established, the first two authors then labeled 20 new ChatGPT answers independently based on the codebook. Since one text span in an answer may suffer from multiple quality issues, the labeling is multi-label multi-class classification, where labels are not mutually exclusive. Therefore, we cannot use Cohen's Kappa to measure the agreement level between labelers. Instead, we used Fleiss's Kappa [23] score. The initial score was 0.45, which was not high enough to proceed to label more answers. Thus, the authors met again to discuss the labeling. They carefully reviewed each label in the answers and resolved the conflicts. They further refined the codebook by merging or deleting redundant codes, improving the definitions of ambiguous codes, and introducing new codes. At the end of this step, the number of labels became 21 in 5 categories.

With the new codebook, the first two authors re-labeled 10 of the previous 20 answers and confirmed the agreement. Except for disagreement about the definition and usage of 2 codes in *correctness* category, no new disagreement was discovered. At this point, Fleiss's Kappa score was 0.79. Next, the first two authors met the co-author to review and refine the current codebook and labelings. At the end of this meeting, the codebook was refined to 19 labels.

Finally, the first two authors labeled 20 new ChatGPT answers with the new cookbook and found very few disagreements that are more subjective in nature (e.g., comprehensiveness). At this point, Fleiss's Kappa score was 0.83. With these codebooks, the first two authors split the remaining ChatGPT answers and labeled them separately. The whole labeling process took about 216 man-hours.

3.2.2 ***Definitions and Discussion of Codebook.*** To understand the fine-grained issues with ChatGPT answers (RQ2), the codebooks contain a wide range of labels covering all 4 themes mentioned in the previous subsection 3.2.1. We give a quick overview of these labels below. Section 4 provides more details.

For ***Correctness***, we compared ChatGPT answers with the accepted SO answers and also resorted to other online resources such as blog posts, tutorials, and official documentation. Our codebook includes four types of correctness issues— *Factual, Conceptual, Code*, and *Terminological* incorrectness. Specifically, for incorrect code examples embedded in ChatGPT answers, we identified four types of code-level incorrectness—*Syntax* errors and errors due to *Wrong Logic, Wrong API/Library/Function Usage*, and *Incomplete Code.*

For ***Consistency***, we measured the consistency between ChatGPT answers and the accepted SO answers. Note that inconsistency does not imply incorrectness. A ChatGPT answer can be different from an accepted SO answer, but can still be correct. Five types of inconsistencies emerged from the manual analysis—*Factual, Conceptual, Terminological, Coding*, and *Number of Solutions* (e.g., ChatGPT provides four solutions where SO gives only one).

For ***Conciseness***, three types of conciseness issues were identified and included in the codebook—*Redundant*, *Irrelevant*, and *Excess* information. *Redundant* sentences reiterate everything stated in the question or in other parts of the answer. *Irrelevant* sentences talk about concepts that are out of the scope of the question being asked. And lastly, *Excess* sentences provide information that is not required to understand the answer.

For ***Comprehensiveness***, we consider two labels—*Comprehensive*, *Not Comprehensive.* To consider an answer to be comprehensive, it needs to fulfill two requirements– 1) All parts of the question are addressed in the answer, and 2) All parts of the solution are addressed in the answer.

3.3 Linguistic Analysis

Previous studies show that user preference and acceptance of SE problems can depend on underlying emotion, tone, and sentiment showcased by the answer [7, 15, 56]. In this section, we describe the methods utilized to determine linguistic features and sentiments of ChatGPT's answers.

3.3.1 ***Linguistic Characteristics.*** We employed the widely used commercial tool Linguistic Inquiry and Word Count (LIWC) [49], as our preferred tool for analyzing the linguistic features of ChatGPT and SO answers. LIWC is a psycholinguistic database that provides a dictionary of validated psycholinguistic lexicon in predetermined categories [49] that are psychologically meaningful. LIWC counts word occurrence frequencies in each category which holds important information about the emotional, cognitive, and structural components associated with text or speech. LIWC has been used to study AI-generated misinformation [65], emotional expressions in social media posts [39], success of SO answers [7], etc. For our work, we considered the following categories:

- **Linguistic Styles:** We considered four attributes for this category – *Analytical Thinking* (complex thinking, abstract thinking), *Clout* (power, confidence, or influential expression), *Authentic* (spontaneity of language), and *Emotional Tone.*
- **Affective Attributes:** Affective attributes capture expressions and features related to emotional status. They include—*Affect* (overall emotional expressions, e.g., "happy, cried"), *Positive Emotion* (e.g., "happy, nice"), *Negative Emotion* (e.g., "hurt, cried").
- **Cognitive Processes:** Cognitive processes represents features that are related to cognitive thinking and processing, e.g., causation, knowledge, insight, etc. For this category, we considered *Insight* (e.g., "think, know"), *Causation* (e.g., "because"), *Discrepancy* (e.g., "should, would"), *Tentative* (e.g., "perhaps"), *Certainty* (e.g., "always"), and *Differentiation* (e.g., "but, else").
- **Drives Attributes:** Drives capture expressions that show the need, desire, and effort to achieve something. For *Drives* category, we considered *Drives, Affiliation, Achievement, Power, Reward,* and *Risk* attributes.

- **Perceptual Attributes:** This category captures the attributes that are related to *Perceive, See, Feel*, or *Hear* something.
- **Informal Attributes:** This category captures the causality in everyday conversations. The attributes in this category are – *Informal Language, Swear Words, Netspeak* (e.g., "btw, lol"), *Assent* (e.g., "OK, Yeah"), *Nonfluencies* (e.g., "er, hmm"), and *Fillers* (e.g., "Imean, youKnow").

We computed word frequency in each of the categories for 2000 ChatGPT and SO answers with LIWC. For ease of understanding, we computed the relative differences (*RD*) in linguistic features between 2000 pairs of ChatGPT and SO answers from the computed average word frequencies in each category.

$$RD = \frac{ChatGPT avg.frequency - SOavg.frequency}{SOavg.frequency}$$

3.3.2 ***Sentiment Analysis.*** Lexicon-based LIWC evaluates emotions and tones based on psycholinguistic features and captures the sentiment of texts only based on overall polarity. Hence, LIWC is insufficient when it comes to capturing the intensity of the polarity [10]. Moreover, LIWC can not capture sarcasm, irony, misspelling, or negation which is necessary to analyze sentiment in human written texts on Q&A platforms. Therefore, we adopted an automated approach and employed a machine learning algorithm to evaluate and compare sentiments in ChatGPT and SO answers. To evaluate the underlying sentiment portrayed by ChatGPT and SO answers, we conducted sentiment analysis on 2000 pairs of ChatGPT and SO answers. We used a RoBERTa-base model from Hugging Face [22] for sentiment analysis on software engineering texts[4]. This model is re-finetuned from the "twitter-roberta-base-sentiment" model[5] with the 4423 annotated SO posts dataset built by Calefato et al. [14]. This well-balanced dataset has 35% posts with positive emotions, 27% of posts with negative emotions, and 38% of neutral posts that convey no emotions.

3.4 User Study

To understand user behavior while choosing between answers generated by ChatGPT and SO, we conducted a within-subjects users study with 12 participants with various levels of programming expertise. Our goal is to observe how users put importance on different characteristics of the answers and how those preferences align with our findings from manual and linguistic analysis.

3.4.1 ***Participants.*** For the user study, we recruited 12 participants (3 female, 9 male). 7 participants were graduate students, and 4 participants were undergraduate students in STEM or CS at R1 universities, and 1 participant was a software engineer from the industry. The participants were recruited by word of mouth. Participants self-reported their expertise by answering multiple-choice questions with 5 options— Novice, Beginner, Competent, Proficient, and Expert. Regarding their programming expertise, 8 participants were proficient, 3 were competent, and 1 was beginner. We also asked participants about their ChatGPT expertise—3 participants were proficient, 6 participants were competent, 1 participant was a beginner, and 2 were novices. Additionally, we asked participants about their familiarity with SO and ChatGPT by asking how often they use them. They self-reported this by answering multiple-choice questions with 5 options— Never, Seldom, Some of the Time, Very Often, and All the Time. For SO, 4 participants answered *all the time*, 5 answered *very often*, 2 answered *some of the time*, and 1 answered *seldom*. For ChatGPT, 3 answered *very often*, 3 answered *some of the time*, 2 answered *seldom*, and 4 answered *never*.

3.4.2 ***SO Question Selection.*** From our labeled dataset, we randomly selected 8 questions where ChatGPT gave incorrect answers for 5 questions and correct answers for 3 questions. Among the selected 8 questions, there was 1 C++, 1 PHP, 2 HTML/CSS, 3 JavaScript, and 1 Python question.

3.4.3 ***Protocol.*** Each user study started with consent collection and an introduction to the study procedure. Then the participants started the task for the user study that was designed as a Qualtrics survey.[6] Each SO question had its individual page in the survey. The sequence of questions on each page for each of the 8 SO questions is in the following order—**(1)** participant's expertise in the topic (e.g., C++, PHP, etc.) of the SO question (5-point Likert Scale), **(2)** the SO question, **(3)** an answer generated by ChatGPT or an accepted SO answer written by human[7], **(4)** 5-point scale questions for assessing the correctness, comprehensiveness, conciseness, and usefulness of this answer, **(5)** the other answer **(6)** 5-point scale questions for assessing the correctness, comprehensiveness, conciseness, and usefulness of the second answer, **(7)** Which answer do you prefer more? (1, 2), **(8)** Guess which answer is machine-generated (1, 2), **(9)** Identify which answer is incorrect (1, 2, none), **(10)** Confidence rating (5-point Likert Scale).

The SO questions, SO answers, and ChatGPT answers were presented with similar text formats (e.g., font type, font size, code format, etc.) to maintain visual consistency among the answers and questions. All participants observed the questions in the same order. However, participants were allowed to skip to the next question if they were not familiar with the topic of a certain question. The order of ChatGPT and SO answers was selected randomly (i.e., not always Answer 1 was ChatGPT answer). Additionally, participants were encouraged to use external verification methods such as Google search, tutorials, and documentation. We closely monitored the study so that participants can not use SO or ChatGPT. Each participant received 20 minutes to examine and rate answers to SO questions. Participants were made aware that finishing all 8 questions were not required and were requested to take as much time as needed for each question. All participants used up the given 20 minutes of time in the study. On average, participants assessed the quality of the answers to 5 questions.

3.4.4 ***Semi-Structured Interview.*** The survey was followed by a light-weight semi-structured interview. Each interview took about 10 minutes on average. During the interview, we reviewed the participant's response to the survey together with the participant and asked them why they preferred one answer over the other.

[4]Cloudy1225/stackoverflow-roberta-base-sentiment

[5]cardiffnlp/twitter-roberta-base-sentiment

[6]The survey is included in Supplementary Material

[7]The order of the two answer are randomized. If the first answer is generated by ChatGPT, the second answer is written by human and vice versa.

Then, we asked the participants about their heuristics to identify the machine-generated answer before revealing the correct answer to them. If the participants were correct, we asked a follow-up question asking about the characteristics of the machine-generated answers that influenced their decision.

Lastly, we asked how they determined the incorrect information in an answer. We also asked follow-up questions such as why they failed to identify some misinformation, what the main challenges were in verifying the correctness, what additional tool support they wish to have, etc.[8]

3.4.5 ***Qualitative Analysis of the Interview Transcripts.*** The first author reviewed all 12 interview transcripts and labeled the transcripts with the open coding methodology [31]. The author marked all insightful responses that mentioned factors related to participants' preferences, the heuristics used by the participants, the obstacles they faced, and the tool support they wished to have. After this step, the author did a bottom-up thematic analysis [13, 29] to group the low-level codes into high-level patterns and themes. The final codebook for thematic analysis contains 5 themes and 21 patterns.[9] The overall process took about 6 man-hours.

4 MANUAL ANALYSIS RESULTS

This section presents the results and findings for RQ1-RQ3.

4.1 RQ1: Overall Correctness and Quality

To evaluate the overall correctness, we computed the number of correct and incorrect labels for 517 answers. Our results show that, among the 517 answers we labeled, 248 (48%) answers are ***Correct***, and 259 (52%) answers are ***Incorrect***. To evaluate the quality of the answers in the other three categories, we computed the number of each label in each category in a similar manner. Our results show that 22% of the answers are ***Consistent*** with human-answer, 65% of the answers are ***Comprehensive***, and only 23% of the answers are ***Concise***. Moreover, on average, ChatGPT and SO answers contain 266.43 tokens (σ = 87.99) and 213.80 tokens (σ = 246.04) respectively. The mean difference of 52.63 tokens is statistically significant (Paired t-Test, p-value <0.001). Table 2 shows a complete list of our manual analysis results.

Finding 1

ChatGPT's answers are more incorrect, significantly lengthy, and not consistent with human answers half of the time. However, ChatGPT's answers are very comprehensive and successfully cover all aspects of the questions and the answers.

4.2 RQ2: Fine-Grained Issues

Our thematic analysis reveals three types of ***incorrectness*** in ChatGPT answers—*Conceptual* (54%), *Factual* (36%), *Code* (28%) and *Terminology* (12%) errors. Note that these errors are not mutually exclusive. Some answers have more than one of these errors. *Factual* errors occur when ChatGPT states some fabricated or untruthful information about existing knowledge, e.g., claiming a certain API solves a problem when it does not, fabricating non-existent links, untruthful explanations, etc. On the other hand, *Conceptual* errors occur if ChatGPT fails to understand the question. For example, the user asked how to use public and private access modifiers, and ChatGPT answered the benefits of encapsulation in C++. *Code* error occurs when the code example in the answer does not work, or can not provide desired output. And lastly, *Terminology* errors are related to wrong usages of correct terminology or any use of incorrect terminology, e.g., *perl* as a header of *Python* code.

[8]A complete list of interview questions have been uploaded to the Supplementary Material.
[9]We have submitted the codebook as Supplementary Material.

		Correct		Consistent		Comprehensive		Concise	
		Yes	No	Yes	No	Yes	No	Yes	No
Popularity	Popular	0.55	0.45	0.21	0.79	0.64	0.36	0.16	0.84
	Avg. Popular	0.46	0.54	0.22	0.78	0.64	0.36	0.26	0.74
	Not Popular	0.42	0.58	0.25	0.75	0.66	0.34	0.28	0.72
Type	Debugging	0.45	0.55	0.17	0.83	0.63	0.37	0.40	0.60
	How-to	0.47	0.53	0.21	0.79	0.67	0.33	0.13	0.87
	Conceptual	0.48	0.52	0.28	0.72	0.64	0.36	0.16	0.84
Recency	Old	0.53	0.47	0.22	0.78	0.68	0.32	0.17	0.83
	New	0.42	0.58	0.22	0.78	0.61	0.39	0.29	0.71
Overall	_	0.48	0.52	0.22	0.78	0.65	0.35	0.23	0.77

Table 2: Percentage of ChatGPT answers in 4 categories across multiple question post Popularity, Type, and Time. The statistically significant (Pearson's Chi-square Test p-value<0.05) relations are highlighted in blue.

Specifically, for code errors, our analysis reveals 4 types of code errors—wrong logic (48%), wrong API/library/function usage (39%), incomplete code (11%), and wrong syntax (2%). Some generated code has more than one of these errors. Logical errors are made by ChatGPT when it can not understand the problem, fails to pinpoint the exact part of the problem, or provides a solution that does not solve the problem. For example, in many debugging instances, we found that ChatGPT tries to resolve one part of the given code, whereas the problem lies in another part of the code. We also observed that ChatGPT often fabricates APIs or claims certain functionalities that are wrong.

Finding 2

Many answers are incorrect due to ChatGPT's incapability to understand the underlying context of the question being asked. Whereas, ChatGPT makes less amount of factual errors compared to conceptual errors.

Finding 3

ChatGPT rarely makes syntax errors for code answers. The majority of the code errors are due to applying wrong logic or implementing non-existing or wrong API, Library, or Functions.

Among the answers that are ***Not Concise***, 46% of them have *Redundant* information, 33% have *Excess* information, and 22% have *Irrelevant* information. For *Redundant* information, during our labeling process we observed that many of the ChatGPT answers repeat the same information that is either stated in the question or stated in other parts of the answers. For *Excess* information, we observed a handful of cases where ChatGPT unnecessarily gives background information such as long definitions, or writes something at the end of the answer that does not add any necessary information to understand the solution. Lastly, many answers contain *Irrelevant* information that is out of context or scope of the question. In answers with conceptual errors, we observed this behavior more often. There are answers that have a combination of more than one of these conciseness issues.

And lastly, for inconsistency with human answers, we found five types of ***Inconsistencies***—*Conceptual* (67%), *Factual* (44%), *Code* (55%), *Terminology* (6%), and *Number of Solutions* (42%). The first four types of inconsistencies occur for the same reason as incorrectness. The only difference is that inconsistency does not always mean incorrectness as explained in Section 3.2.2. Similar to incorrectness, conceptual inconsistencies are higher than factual inconsistencies. Our observation also reveals that ChatGPT-generated code is very different in format, semantics, syntax, and logic than human-written code. This contributes to the higher number of *Code* inconsistencies. The *Number of solutions* inconsistency is very prominent as ChatGPT often provides many additional solutions to solve a problem.

4.3 RQ3: Effects of Question Type

To evaluate the relationship between question types and ChatGPT's answer quality, we calculated the percentage of each label across all categories for each question type. As our data is entirely categorical, we evaluated the statistical significance of the relationship between each question type and each of the 4 label categories with Pearson's Chi-Squared test. Table 2 highlights all relationships that are statistically significant (p-value < 0.05). Our results show that ***Question Popularity*** and ***Recency*** have a statistically significant impact on the ***Correctness*** of answers. Specifically, answers to popular questions and questions posted before November 2022 have fewer incorrect answers than answers to questions with other levels of popularity or time range. This implies that ChatGPT generates more correct answers when it has more information about the question topic in its training data. Although answers to ***Debugging*** questions have higher incorrectness, there was not any statistically significant relation between ***Question Type*** and ***Correctness***.

Additionally, we found a statistically significant relationship between ***Question Type*** and ***Inconsistency***. Since there are often multiple ways to debug and fix a problem, the inconsistencies between human and ChatGPT-generated answers for ***Debugging*** questions are higher, with 83% of ***inconsistent*** answers. Our observation aligns with this result too. While labeling the answers, we found that almost half of the correct ***Debugging*** answers use different logic, API, or library to solve a problem that produces the same output as human answers.

Our results also show that ChatGPT answers are consistently ***Comprehensive*** for all categories of SO questions and do not vary with different ***Question Type, Recency,*** or ***Popularity***.

Moreover, our analysis shows that all three of the SO question categories have a statistically significant impact on the conciseness of the answer. Answers to all questions, irrespective of the ***Type, Recency,*** and ***Popularity***, are consistently verbose. Specifically, answers to ***Popular*** questions are ***Not Concise*** 84% of the time, while answers for ***Average*** and ***Not Popular*** questions are ***Not Concise*** 74% and 72% of the time. This suggests that for questions targeting popular topics, ChatGPT has more information on them and adds lengthy details. We found the same pattern for ***Old*** questions. Answers to ***Old*** questions (83%) are more verbose than ***New*** questions (71%). Finally, for ***Question Type***, ***Debugging*** answers are more ***Concise*** (40%) compared to ***Conceptual*** (16%) and ***How-to*** (13%) answers, which are extremely verbose. This is because of ChatGPT's tendency to elaborate definitions for ***Conceptual*** questions and to generate step-by-step descriptions for ***How-to*** questions.

Linguistic Features	Rel. Diff.(%)	Linguistic Features	Rel. Diff.(%)
		Drive Attributes	
Language Styles		Drives	9.53***
Analytic	20.65***	Affiliation	16.05**
Clout	13.01***	Achievement	10.85***
Authentic	-38.50***	Power	22.86***
Tone	14.95***	Reward	2.23
		Risk	-7.08
Affective Attributes		**Perception Attributes**	
Affect	-6.53**	Perception	-26.28***
Positive Emotion	2.09	See	-34.98***
Negative Emotion	-34.45***	Hear	-16.50*
		Feel	7.55
Cognitive Attributes		**Informal Attributes**	
Insight	-8.86**	Informal Language	-53.97***
Causation	23.94***	Swear words	-71.52**
Discrepancy	-35.89***	Netspeak	-60.03***
Tentative	-10.23***	Assent	-11.86
Certainty	-4.23	Nonfluencies	-55.34***
Differentiation	-13.29***	Fillers	-82.85***

Table 3: Relative Linguistic Differences (%) between 2000 pairs of ChatGPT and SO answers. Positive numbers indicate higher occurrence frequencies of linguistic features in ChatGPT answers compared to SO, and negative numbers indicate lower occurrence frequencies. Numbers marked with (*) indicate differences that are statistically significant (Paired t-Test, *** p-value <0.001, ** p-value<0.01, * p-value<0.05)

Finding 4

The Popularity, Type, and Recency of the SO questions affect the correctness and quality of ChatGPT answers. Answers to more *Popular* and *Older* posts are less incorrect and more verbose. *Debugging* answers are more inconsistent, but less verbose, and *Conceptual* and *How-to* answers are the most verbose.

5 LINGUISTIC ANALYSIS RESULTS

5.1 RQ4: Linguistic Characteristics

Table 3 presents the relative differences in the linguistic features between ChatGPT and SO. As stated in Section 3.3, relative differences capture the normalized difference in word frequencies for each linguistic feature between ChatGPT and SO. The positive relative differences indicate the features that are prominent to ChatGPT, and the negative relative differences indicate the features that are prominent to SO. Our result shows a statistically significant linguistic difference between ChatGPT and SO answers.

ChatGPT answers differ from SO answers in terms of ***language styles***. ChatGPT answers are found to contain more words related to *analytical thinking and clout expressions*. This indicates that ChatGPT answers communicate a more abstract and cognitive understanding of the answer topic, and the language style is more influential and more confident. On the other hand, ChatGPT answers include fewer words related to *authenticity*, indicating that SO answers are more spontaneous and non-regulated.

For ***affective*** attributes that capture emotional status, we found SO answers contain more keywords related to emotional status. Though not statistically significant, ChatGPT answers portray more positive emotions, whereas SO answers portray significantly stronger negative emotions than ChatGPT.

Moreover, ChatGPT answers contain significantly more ***drives*** attributes compared to SO answers. ChatGPT conveys stronger *drives, affiliation, achievement*, and *power* in its answers. On many occasions we observed ChatGPT inserting words and phrases such as "of course I can help you", "this will certainly fix it", etc. This observation aligns with the higher *drives* attributes in ChatGPT-generated answers. However, ChatGPT does not convey risks as much as SO does. This indicates human answers warn readers of the consequential effects of solutions more than ChatGPT does.

For ***informal*** attributes, SO answers are highly informal and casual. On the contrary, ChatGPT answers are very formal and do not make use of swear words, netspeak, nonfluencies, or fillers. In our observation, we rarely saw ChatGPT using a casual conversation style. On the other hand, SO answers often had words such as "btw", "I guess", etc. SO answers also contain higher ***perceptual*** and ***cognitive*** keywords than ChatGPT answers. This means SO answers portrays more insights and understanding of the problem.

Finding 5

Compared to SO answers, ChatGPT answers are more formal, express more analytic thinking, showcase more efforts towards achieving goals, and exhibit less negative emotion.

5.2 RQ5: Sentiment Analysis

Our results show that, for ChatGPT, among 2000 answers, 1707 (85.35%) answers portray positive sentiment, 291 answers (14.55%) portray neutral sentiment, and only 2 answers (0.1%) portray negative sentiment. On the other hand, for SO, 1466 of the 2000 answers (73.30%) portray positive sentiment, 513 answers (25.65%) portray neutral, and 21 answers (1.05%) portray negative sentiment. To assess the sentiment difference between ChatGPT and SO answers, we performed a McNemar-Bowker test on the sentiments. Since we have paired-nominal data, we opted for the McNemar-Bowker test for testing the goodness of fit when comparing the distribution of counts of each label. The results are statistically significant ($X^2 = 186.84, df = 3, p < 0.001$). Our results show that for 13.90% questions, ChatGPT's answers portrayed positive sentiment while the SO's answers portrayed neutral or negative sentiments. On the other hand, only 2 ChatGPT answers portrayed negative sentiment when the SO answer was positive or neutral. Our result indicates that ChatGPT shows significantly more positive and less negative sentiment compared to SO.

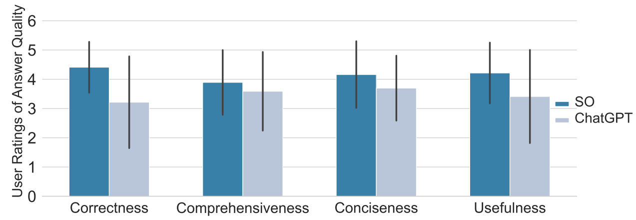

Figure 1: Quality of answers as rated by participants. Difference in *Correctness, Conciseness*, and *Usefulness* are statistically significant (Paired t-Test, p-value < 0.05)

Finding 6

ChatGPT answers portray significantly more positive sentiments compared to SO answers.

6 USER STUDY RESULTS

We retrieved 56 pairs of ratings of SO and ChatGPT answers as rated by 12 participants. Figure 1 presents the average ratings of the SO and ChatGPT answers in all 4 aspects. Overall, users found SO answers to be more correct (mean rating SO: 4.41, ChatGPT: 3.21, Welch's t-test: p-value< 0.001), more concise (SO: 4.16, ChatGPT: 3.69, Welch's t-test: p-value < 0.05), and more useful (SO: 4.21, ChatGPT: 3.42, Welch's t-test: p-value < 0.01). For comprehensiveness, the average ratings are 3.89 and 3.98 for SO and ChatGPT, but this result is not statistically significant.

Additionally, our thematic analysis revealed five themes— *Process of differentiating ChatGPT answers from SO answers*, *Heuristics of verifying correctness*, *Reasons for incorrect determination*, *Desired support*, and *Factors that influence user preference*. Findings from our quantitative and thematic analysis for each of the research questions are described in the following subsections.

6.1 RQ6: Differentiating ChatGPT answers from SO answers

Our study results show that participants successfully identified which one is the machine-generated answer 80.75% of the time and failed only 14.25% of the time (Welch's t-Test, p-value < 0.001).

From thematic analysis, we identified the factors that participants found helpful to discern ChatGPT answers from human answers. 6 out of 12 participants reported the writing style of answers to be helpful in identifying the ChatGPT answer. Participant P5 mentioned, *"good grammar"*, and P8 mentioned, *"header, body, summary format"* to be contributing factors for identification. Two other factors are language style (e.g., casual or formal language, format) (10 out of 12) and length (7 out of 12). Additionally, 5 out of 12 participants found unexpected or impossible errors as a helpful factor to identify the machine-generated answers. Apart from these, tricks and insights that only experienced people can provide (5 out of 12), and high-entropy-generation (1 out of 12) were two other reported factors. Our result suggests that most participants use language and writing styles, length, and the presence of abnormal errors to determine the source of an answer.

Finding 7

Participants can correctly discern ChatGPT answers from SO answers 80.75% of the time. They look for factors such as formal language, structured writing, length of answers, or unusual errors to decide whether an answer is generated by ChatGPT.

6.2 RQ7: Assessing Correctness of ChatGPT Answers

Our study result shows that users could successfully identify the incorrect answers only 60.66% of the time and failed 39.34% of the time (Welch's t-test, p-value < 0.05).

When we asked users how they identified incorrect information in an answer, we received three types of responses. 10 out of 12 participants mentioned they read through the answer, tried to find any logical flaws, and tried to assess if the reasoning make sense. 7 participants mentioned they identified the terminology and concepts they were not familiar with and did a Google search, and read documentation to verify the solutions. And lastly, 4 out of 12 users mentioned that they compared the two answers and tried to understand which one made more sense to them.

When a participant failed to correctly identify the incorrect answer, we asked them what could be the contributing factors. 7 out of 12 participants mentioned the logical and insightful explanations, and comprehensive and easy-to-read solutions generated by ChatGPT made them believe it to be correct. 6 participants mentioned lack of expertise to be the reason. However, we ran Pearson's Chi-Square test to evaluate the relationship between the wrong identification of incorrect answers and topic expertise and found no significant relation between these two. P7 and P10 stated ChatGPT's ability to mimic human answers made them trust the incorrect answers.

Finding 8

Users overlook incorrect information in ChatGPT answers (39.34% of the time) due to the comprehensive, well-articulated, and humanoid insights in ChatGPT answers.

Additionally, participants expressed their desire for tools and support that can help them verify the correctness. 10 out of 12 participants emphasized the necessity of verifying answers generated by ChatGPT before using it. Participants also suggested adding links to official documentation and supporting in-situ execution of generated code to ease the validation process.

6.3 RQ8: Factors for User Preference

Participants preferred SO answers 65.18% of the time. But participants still preferred ChatGPT answers 34.82% of the time (Welch's t-test, p-value < 0.01). Among the ChatGPT preferences, 77.27% of the answers were incorrect.

For *factors that influence user preference*, 10 out of 12 participants mentioned correctness to be the main contributing factor for preference. 8 participants mentioned answer quality (e.g., conciseness, comprehensiveness) as contributing factors. 6 participants mentioned they put emphasis on how insightful and informative the answer is while preferring. 6 participants stated language style to be one of the factors, 2 of these 6 participants preferred the casual, spontaneous language style of SO answer, while the other 4 preferred the well-structured and polite language of ChatGPT. P2 mentioned, *"It feels like it's trying to teach me something"*. Finally, 5 participants mentioned the format, look and feel (e.g., highlighting, color scheme) as contributing factors toward preference.

Finding 9

Participants preferred SO answers more than ChatGPT answers (65.18% of the time). Participants found SO answers to be more concise, and useful. A few reasons for SO preferences are – correctness, conciseness, casual and spontaneous language, etc.

7 DISCUSSION AND FUTURE WORK

In this section, we discuss the implications of our analysis results and future research directions.

Conceptual errors are as pressing as factual errors. It is evident from our results ChatGPT produces incorrect answers more than half of the time. 54% of the time the errors are made due to ChatGPT not understanding the concept of the questions. Even when it can understand the question, it fails to show an understanding of how to solve the problem. This contributes to the high number of *Conceptual* errors. Since questions asked in SO are human-written large questions, ChatGPT often focuses on the wrong part of the question or gives high-level solutions without fully understanding the minute details of a problem. From our observation, another reason behind conceptual errors is ChatGPT's limitation to reason. In many cases, we saw ChatGPT gives a solution, code, or formula without foresight or thinking about the outcome. On the other hand, the rate of *Factual* error is lower than *Conceptual* error but still composes a large portion of incorrectness. Although existing work focus on removing hallucinations from *LLM* [20, 48], those are only applicable to fixing *Factual* errors. Since the root of *Conceptual* error is not hallucinations, but rather a lack of understanding and reasoning, the existing fixes for hallucination are not applicable to reduce conceptual errors. Prompt engineering and Human-in-the-loop fine-tuning can be helpful in probing ChatGPT to understand a problem to some extent [59, 66], but they are still insufficient when it comes to injecting reasoning into *LLM*. Hence it is essential to understand the factors of conceptual errors as well as fix the errors originating from the limitation of reasoning.

Code errors are analogous to Non-Code errors. For errors in code examples provided by ChatGPT, we found wrong logic and wrong API/library/function usage to be two of the main reasons. From our observation, wrong logic is analogous to conceptual error. Most of the logical errors are due to the fact that ChatGPT can not understand the problem as a whole, fails to pinpoint the exact part of the problem, etc. For example, in many debugging instances, we found that ChatGPT tries to resolve one part of the given code, whereas the problem lies in another part of the code. Also, in many cases, ChatGPT makes discernible logical errors that no human expert can make. For example, setting a loop ending condition to be equal to something that is never true or never false (e.g. $while(i < 0 \ and \ i > 10)$). Referring to the previous discussion, we believe for fixing logical errors, it is imperative to focus on teaching ChatGPT to reason. On the other hand, wrong

API/library/function usage is analogous to factual error and further research should be conducted to reduce hallucination from codes.

Low-quality is linked to incorrectness. Since our study results suggest that a large percentage of the ChatGPT answers suffer from poor quality, we argue that it is inevitable to reduce the errors in order to make ChatGPT more useful in SE tasks. Moreover, steps should be taken to reduce the verbosity of ChatGPT answers. Much of the redundant, excessive, and irrelevant information emerges from the same reason as conceptual errors and the inability to pinpoint a problem. Addressing these sources of problems and devising more holistic summarizing while text generation can be some future research direction to improve answer quality.

Users get tricked by appearance. Our user study results show that users prefer ChatGPT answers 34.82% of the time. However, 77.27% of these preferences are incorrect answers. We believe this observation is worth investigating. During our study, we observed that only when the error in the ChatGPT answer is obvious, users can identify the error. However, when the error is not readily verifiable or requires external IDE or documentation, users often fail to identify the incorrectness or underestimate the degree of error in the answer. Surprisingly, even when the answer has an obvious error, 2 out of 12 participants still marked them as correct and preferred that answer. From semi-structured interviews, it is apparent that polite language, articulated and text-book style answers, comprehensiveness, and affiliation in answers make completely wrong answers seem correct. We argue that these seemingly correct-looking answers are the most fatal. They can easily trick users into thinking that they are correct, especially when they lack the expertise or means to readily verify the correctness. It is even more dangerous when a human is not involved in the generation process and generated results are automatically used elsewhere by another AI. The chain of errors will propagate and have devastating effects in these situations. With the large percentage of incorrect answers ChatGPT generates, this situation is alarming. Hence it is crucial to communicate the level of correctness to users.

Communication of incorrectness is the key. Although ChatGPT's chat interface has a one-line warning- "*ChatGPT may produce inaccurate information about people, places, or facts*", we believe such a generic warning is insufficient. Each answer should be accompanied by a level of incorrectness and uncertainty in the answer. Previous studies show that *LLM* knows when it is lying [4], but does *LLM* know when it is speculating? And how can we communicate the level of speculation? Therefore, it is imperative to investigate how to communicate the level of incorrectness of the answers. Moreover, human factor research on communicating the level of incorrectness in codes or programming answers in a way that users understand is another important direction worth exploring.

Caution and awareness are necessary. Finally, awareness should be created among users regarding the ignorance that is induced by seemingly correct answers. Verifying the answers no matter how correct or trustworthy they look is imperative. We would like to conclude this discussion with the statement that, AI is most effective when supervised by humans. Therefore, we call for the responsible use of ChatGPT to increase human-AI productivity.

8 THREATS TO VALIDITY

Internal Validity. Our manual analysis raises a few threats to internal validity. Due to the subjective nature of the manual analysis, it is limited by human judgment and bias. We tried to reduce this threat by multiple levels of reviews and agreements and only labeled the data when Fleiss's Kappa was large enough.

Besides, our user study raises several threats such as sample size, participants' bias, and thematic analysis. Although 12 is a small sample size, each participant provided us with 5 data points on average (56 in total). This reduces the threat due to the small number of participants. To reduce participants' own bias against human or machine-generated answers, we hid the source of answers during the study and made the answers visually similar (e.g., same font size, type, code style, etc.). And for thematic analysis, we followed several iterations to make a comprehensive list of codes (21) and 5 themes to capture minute details from user feedback and reduce subjective human bias.

External Validity. The external validity concerns the generalizability of the findings of our analyses. To reduce this threat, we collected SO answers across diverse categories and topics. Furthermore, we recruited participants with different levels of expertise in programming and in individual topics.

9 RELATED WORK

The proliferation of social media and Q&A platforms for software engineering tasks immensely affected the help-seeking behavior among software engineers [58, 60]. These platforms changed the way software engineers communicate, ask for help, and write software [1, 62] as they incorporate crowdsourced knowledge in different steps. Existing literature establish the role and benefits of Q&A platforms [40, 47, 60]. Treude et al. [60] show that Q&A platforms such as SO are highly effective in code reviews and answering conceptual questions. Through a mixed-methods study, Mamykina et al. [40] show that the chance of quickly getting help from the SO community is very high, with a median time of 11 minutes. However, despite the popularity and effectiveness of SO, concerns about the fact that SO can potentially interrupt developers' workflow and impair their performance persists [58, 62]. Since Q&A platforms are not integrated with IDEs, this causes developers to switch between IDEs and external resources which in turn causes delay and interruptions [5].

The emergence of GitHub Copilot [27] addressed this gap by integrating assistive AI tools in IDEs in the form of code autocompletion. Moreover, Copilot can provide further assistance such as code translation, explanations, and documentation generation [3, 55]. Studies also show that AI pair-programming has shifted developers' behavior from code writing to code understanding, rendering better efficiency [9], and productivity [33]. The online help-seeking behavior of developers also changed along with other behavior shifts. Developers use Copilot for prompt and concise solutions, and only turn to web searches or SO to access the documentation or verify solutions [6, 55, 61]. Regardless of the popularity and acceptance of Copilot among developers, studies show that Copilot produces buggy code that can become a liability for novice developers [18]. Moreover, Copilot generates codes that are inferior in quality and requires significant deletion and modifications [33]. Vaithilingam et al. [61] show that despite the limitations, developers still prefer using Copilot as a convenient starting point.

Although Copilot remains popular and useful among developers, ChatGPT has gained popularity among programmers of all levels and expertise since its release in November 2022. The usability and effectiveness of ChatGPT compared to human experts are examined in different fields such as legal, medical, financial, etc. [30]. More recent studies show that ChatGPT often fabricates facts, and generates low quality or misleading information [12, 30, 36, 43]. To the best of our knowledge, no other work conducted an in-depth and comprehensive analysis of the characteristics and quality of ChatGPT answers to SO questions in the field of SE.

10 CONCLUSION

In this paper, we empirically studied the characteristics of ChatGPT answers to SO questions through mixed-methods research consisting of manual analysis, linguistic analysis, and user study. Our manual analysis shows that ChatGPT produces incorrect answers more than 50% of the time. Moreover, ChatGPT suffers from other quality issues such as verbosity, inconsistency, etc. Results of the in-depth manual analysis also point towards a large number of conceptual and logical errors in ChatGPT answers. Additionally, our linguistic analysis results show that ChatGPT answers are very formal, and rarely portray negative sentiments. Although our user study shows higher user preference and quality rating for SO, users make occasional mistakes by preferring incorrect ChatGPT answers based on ChatGPT's articulated language styles, as well as seemingly correct logic that is presented with positive assertions. Since ChatGPT produces a large number of incorrect answers, our results emphasize the necessity of caution and awareness regarding the usage of ChatGPT answers in SE tasks. This work also seeks to encourage further research in identifying and mitigating different types of conceptual and factual errors. Finally, we expect this work will foster more research on transparency and communication of incorrectness in machine-generated answers, especially in the context of SE.

REFERENCES

[1] Rabe Abdalkareem, Emad Shihab, and Juergen Rilling. 2017. What do developers use the crowd for? a study using stack overflow. *IEEE Software* 34, 2 (2017), 53–60.

[2] Miltiadis Allamanis and Charles Sutton. 2013. Why, when, and what: analyzing stack overflow questions by topic, type, and code. In *2013 10th Working conference on mining software repositories (MSR)*. IEEE, 53–56.

[3] Irene Alvarado, Idan Gazit, and Amelia Wattenberger. 2022. GitHub Next | GitHub Copilot Labs. https://githubnext.com/projects/copilot-labs/.

[4] Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734* (2023).

[5] Alberto Bacchelli, Luca Ponzanelli, and Michele Lanza. 2012. Harnessing stack overflow for the ide. In *2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE)*. IEEE, 26–30.

[6] Shraddha Barke, Michael B James, and Nadia Polikarpova. 2023. Grounded copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages* 7, OOPSLA1 (2023), 85–111.

[7] Blerina Bazelli, Abram Hindle, and Eleni Stroulia. 2013. On the personality traits of stackoverflow users. In *2013 IEEE international conference on software maintenance*. IEEE, 460–463.

[8] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 610–623.

[9] Christian Bird, Denae Ford, Thomas Zimmermann, Nicole Forsgren, Eirini Kalliamvakou, Travis Lowdermilk, and Idan Gazit. 2022. Taking Flight with Copilot: Early insights and opportunities of AI-powered pair-programming tools. *Queue* 20, 6 (2022), 35–57.

[10] Kristof Boghe. 2020. We Need to Talk About Sentiment Analysis. https://medium.com/p/9d1f20f2ebfb.

[11] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[12] Ali Borji. 2023. A categorical archive of chatgpt failures. *arXiv preprint arXiv:2302.03494* (2023).

[13] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[14] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. 2018. Sentiment polarity detection for software development. In *Proceedings of the 40th International Conference on Software Engineering*. 128–128.

[15] Fabio Calefato, Filippo Lanubile, Maria Concetta Marasciulo, and Nicole Novielli. 2015. Mining successful answers in stack overflow. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. IEEE, 430–433.

[16] Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. *arXiv preprint arXiv:2106.09231* (2021).

[17] Alex Castillo. Dec, 2022. [Twitter Post] ChatGPT will replace StackOverflow. https://twitter.com/castillo__io/status/1599255771736604673?s=20.

[18] Arghavan Moradi Dakhel, Vahid Majdinasab, Amin Nikanjam, Foutse Khomh, Michel C Desmarais, and Zhen Ming Jack Jiang. 2023. Github copilot ai pair programmer: Asset or liability? *Journal of Systems and Software* 203 (2023), 111734.

[19] Lucas BL De Souza, Eduardo C Campos, and Marcelo de A Maia. 2014. Ranking crowd knowledge to assist software development. In *Proceedings of the 22nd International Conference on Program Comprehension*. 72–82.

[20] Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329* (2022).

[21] Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics* 9 (2021), 1012–1031.

[22] Hugging Face. 2022. Hugging Face – The AI community building the future. https://huggingface.co/.

[23] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76, 5 (1971), 378.

[24] Dilrukshi Gamage, Piyush Ghasiya, Vamshi Bonagiri, Mark E Whiting, and Kazutoshi Sasahara. 2022. Are deepfakes concerning? analyzing conversations of deepfakes on reddit and exploring societal implications. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–19.

[25] Catherine A Gao, Frederick M Howard, Nikolay S Markov, Emma C Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T Pearson. 2022. Comparing scientific abstracts generated by ChatGPT to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. *BioRxiv* (2022), 2022–12.

[26] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462* (2020).

[27] Inc. GitHub. 2023. GitHub Copilot · Your AI pair programmer. https://github.com/features/copilot.

[28] Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 166–175.

[29] Greg Guest, Kathleen M MacQueen, and Emily E Namey. 2011. *Applied thematic analysis*. sage publications.

[30] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597* (2023).

[31] Beverley Hancock, Elizabeth Ockleford, and Kate Windridge. 2001. *An introduction to qualitative research*. Trent focus group London.

[32] Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839* (2021).

[33] Saki Imai. 2022. Is github copilot a substitute for human pair-programming? an empirical study. In *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*. 319–321.

[34] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. Summarizing source code using a neural attention model. In *54th Annual Meeting of the Association for Computational Linguistics 2016*. Association for Computational Linguistics, 2073–2083.

[35] R Burke Johnson and Anthony J Onwuegbuzie. 2004. Mixed methods research: A research paradigm whose time has come. *Educational researcher* 33, 7 (2004), 14–26.

[36] Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. ChatGPT: Jack of all trades, master of none. *Information Fusion* (2023), 101861.

[37] Bonan Kou, Muhao Chen, and Tianyi Zhang. 2023. Automated Summarization of Stack Overflow Posts. In *45th IEEE/ACM International Conference on Software Engineering, ICSE 2023, Melbourne, Australia, May 14-20, 2023*. IEEE, 1853–1865. https://doi.org/10.1109/ICSE48619.2023.00158

[38] Bonan Kou, Yifeng Di, Muhao Chen, and Tianyi Zhang. 2022. SOSum: a dataset of stack overflow post summaries. In *Proceedings of the 19th International Conference on Mining Software Repositories*. 247–251.

[39] Adam DI Kramer. 2012. The spread of emotion via Facebook. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 767–770.

[40] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. 2011. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2857–2866.

[41] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661* (2020).

[42] Courtney Miller, Sophie Cohen, Daniel Klug, Bogdan Vasilescu, and Christian Kastner. 2022. " Did you miss my comment or what?" understanding toxicity in open source discussions. In *Proceedings of the 44th International Conference on Software Engineering*. 710–722.

[43] Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852* (2023).

[44] OpenAI. 2023. ChatGPT: Optimizing Language Models for Dialogue. https://openai.com/blog/chatgpt.

[45] OpneAI. 2022. ChatGPT Chat Interface. https://chat.openai.com/.

[46] Stack Overflow. 2023. Temporary policy: Generative AI (e.g., ChatGPT) is banned. https://meta.stackoverflow.com/questions/421831/temporary-policy-generative-ai-e-g-chatgpt-is-banned.

[47] Chris Parnin, Christoph Treude, Lars Grammel, and Margaret-Anne Storey. 2012. Crowd documentation: Exploring the coverage and the dynamics of API discussions on Stack Overflow. *Georgia Institute of Technology, Tech. Rep* 11 (2012).

[48] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813* (2023).

[49] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.

[50] Huilian Sophie Qiu, Yucen Lily Li, Susmita Padala, Anita Sarma, and Bogdan Vasilescu. 2019. The signals that potential contributors look for when choosing open-source projects. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–29.

[51] Quora. 2023. Will ChatGPT replace Stack Overflow? https://www.quora.com/Will-ChatGPT-replace-Stack-Overflow.

[52] Md Masudur Rahman, Jed Barson, Sydney Paul, Joshua Kayani, Federico Andrés Lois, Sebastián Fernandez Quezada, Christopher Parnin, Kathryn T Stolee, and Baishakhi Ray. 2018. Evaluating how developers use general-purpose web-search for code retrieval. In *Proceedings of the 15th International Conference on Mining Software Repositories*. 465–475.

[53] Nikitha Rao, Chetan Bansal, Thomas Zimmermann, Ahmed Hassan Awadallah, and Nachiappan Nagappan. 2020. Analyzing web search behavior for software engineering tasks. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 768–777.

[54] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Sequence labeling with multiple annotators. *Machine learning* 95 (2014), 165–181.

[55] Steven I Ross, Fernando Martinez, Stephanie Houde, Michael Muller, and Justin D Weisz. 2023. The programmer's assistant: Conversational interaction with a large language model for software development. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 491–514.

[56] Ben Shneiderman. 1980. *Software psychology: Human factors in computer and information systems (Winthrop computer systems series)*. Winthrop Publishers.

[57] James Skripchuk, Neil Bennett, Jeffrey Zhang, Eric Li, and Thomas Price. 2023. Analysis of Novices' Web-Based Help-Seeking Behavior While Programming. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 945–951.

[58] Margaret-Anne Storey, Christoph Treude, Arie Van Deursen, and Li-Te Cheng. 2010. The impact of social media on software engineering practices and tools. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*. 359–364.

[59] Hendrik Strobelt, Albert Webson, Victor Sanh, Benjamin Hoover, Johanna Beyer, Hanspeter Pfister, and Alexander M Rush. 2022. Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. *IEEE transactions on visualization and computer graphics* 29, 1 (2022), 1146–1156.

[60] Christoph Treude, Ohad Barzilay, and Margaret-Anne Storey. 2011. How do programmers ask and answer questions on the web?(nier track). In *Proceedings of the 33rd international conference on software engineering*. 804–807.

[61] Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Chi conference on human factors in computing systems extended abstracts*. 1–7.

[62] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. 2013. Stackoverflow and github: Associations between software development and crowdsourced knowledge. In *2013 International Conference on Social Computing*. IEEE, 188–195.

[63] Xin Xia, Lingfeng Bao, David Lo, Pavneet Singh Kochhar, Ahmed E Hassan, and Zhenchang Xing. 2017. What do developers search for on the web? *Empirical Software Engineering* 22 (2017), 3149–3185.

[64] Jinghang Xu, Wanli Zuo, Shining Liang, and Xianglin Zuo. 2020. A review of dataset and labeling methods for causality extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*. 1519–1531.

[65] Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.

[66] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910* (2022).