How Close Is ChatGPT to Developer Judgment? A Study on Stack Overflow Java Questions

Felipe Augusto Guimarães Reis[1], **Marcelo A. Maia**[2], **Carlos Eduardo C. Dantas**[1]

[1]Instituto Federal do Triângulo Mineiro (IFTM) – Uberlândia, MG – Brazil

[2]Universidade Federal de Uberlândia (UFU) – Uberlândia, MG – Brazil

felipeaggs@gmail.com,marcelo.maia@ufu.br,carloseduardodantas@iftm.edu.br

Abstract. *Software developers often seek assistance on platforms such as Stack Overflow. However, with the emergence of Large Language Models (LLMs) such as ChatGPT, the way developers seek help online is gradually changing. This shift does not necessarily guarantee the accuracy of the information provided, as LLMs can have a limitation to accurately understand complex domain-specific content leading to incorrect responses. This study aims to evaluate how closely ChatGPT's choices align with those of the Stack Overflow users in accurately addressing technical questions. For this purpose, 776 Java-related questions were selected from Stack Overflow. ChatGPT was asked to analyze five provided answers from Stack Overflow users for each question and identify the one it considered most accurate. The results show that ChatGPT identifies the accepted answer by the Stack Overflow users in 56% of the cases. In the 44% of cases where ChatGPT diverged from the accepted answer, manual analysis revealed that its selected answer was still technically accurate in many instances, although it was not marked as accepted on Stack Overflow. In particular, 31% of these divergent choices were posted after Stack Overflow users had already chosen the accepted one. This suggests that some questions on Stack Overflow may have multiple valid answers, including more recent ones that are as accurate as the accepted answer displayed at the top of the page.*

1. Introduction

In the past decade, technical Q&A platforms such as Stack Overflow (SO) [Stack Overflow] have become increasingly popular among developers [May et al. 2019]. By June 2025, the platform had accumulated approximately 24 million questions, 36 million answers, and 96 million comments [StackExchange 2025]. Developers often seek assistance with the usage and behavior of programming *APIs*, understanding technical concepts, identifying common practices, or making informed technical decisions [Beyer et al. 2020].

In recent years, with the emergence of Large Language Models (LLM) based on transformer architecture [Vaswani et al. 2017] such as ChatGPT [CHATGPT 2025], developed by OpenAI [Radford et al. 2018], developers have increasingly turned to these tools for assistance [Tufano et al. 2024]. Recent studies show that ChatGPT has proven particularly effective for tasks such as code refactoring [Zhang et al. 2024], bug fixing [Sobania et al. 2023], suggesting code changes, and other programming activities on GitHub [Tufano et al. 2024]. Furthermore, it offers an alternative approach by modifying versions of the source code provided by developers [Ebert and Louridas 2023]. ChatGPT could also help with code understanding [Nam et al. 2024] and produce high-quality educational code snippets that are often easier to read than those written by humans on SO [Dantas et al. 2023].

However, recent studies have highlighted the limitations of LLM in accurately understanding complex technical content and integrating the context of domain-specific terminology and documentation [Tamanna et al. 2025]. For example, in a recent work, even when tuned with benchmark issue reports, GPT provides correct answers to only 36.4% of the questions [Tamanna et al. 2025]. Another challenge is the phenomenon of hallucinations [Ji et al. 2023], in which the model generates content that is non-existent or not recognizable to human observers [Zuccon et al. 2023], including code or references that appear plausible but lack reliable provenance [Bifolco et al. 2025].

Rather than evaluating LLMs based on the answers they generate, this study takes a different approach: we investigate whether ChatGPT can accurately identify the best answer to a programming question, as judged by the SO users. This allows us to assess the model's ability to align with real-world developer judgments in information-seeking scenarios.

This study investigates 776 SO questions tagged with the Java programming language, assessing whether ChatGPT can identify the same answer selected by SO users (via scores) and accepted by the question author. Furthermore, the study proposes a Likert scale evaluation to assess the precision of divergent answers selected by ChatGPT. This study provides the following contributions.

1. An assessment of the effectiveness of ChatGPT in selecting SO answers as perceived by SO users.
2. A qualitative analysis of instances in which ChatGPT's answers diverged from those chosen by the SO users.
3. Insights on querying ChatGPT API, parameters and prompts.
4. The replication package includes the scripts, parameters, data, and responses returned by ChatGPT API to replicate the experiment. [Reis et al. 2025].

1.1. Research Context

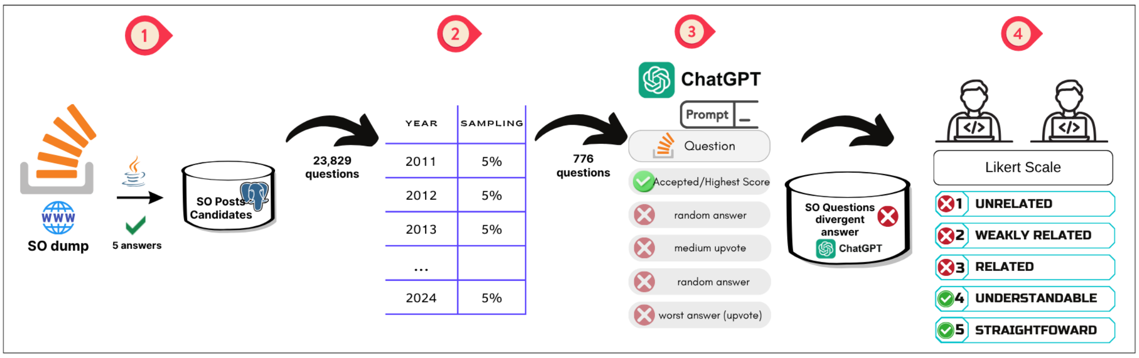
This study was developed as part of an undergraduate final project, not associated with a larger research initiative. The investigation was conducted individually under academic supervision.

2. Methodology

This study is based on the following research question.

- **RQ$_1$: What is the agreement between ChatGPT's selected answers and those preferred by human developers on Stack Overflow (SO)?** Although recent studies have explored the capabilities of LLMs such as ChatGPT in programming tasks, there is limited understanding of how closely their judgments align with those of developers in real-world Q&A platforms such as SO. This research question analyzes the extent to which ChatGPT selects the same answers chosen by SO users.

Figure 1 illustrates the methodology proposed to answer the research question. The following subsections detail the methodology adopted to answer it.



YEAR	SAMPLING
2011	5%
2012	5%
2013	5%
...	
2024	5%

Figure 1. Overview of the proposed methodology

2.1. Extracting Candidate Stack Overflow (SO) Posts

The first step consists of extracting SO posts, including questions and their respective answers. The dump used in this study is publicly available [Stack Overflow Dump], and corresponds to the April 2024 release. This process is illustrated in step 1 of Figure 1.

As SO contains posts on a wide range of topics, this study selected a subset of questions based on the following criteria:

1. Questions with at least five answers, as each ChatGPT prompt contains the question and five candidate answers;
2. Questions in which an accepted answer was selected by the original author, indicating their recognition of their accuracy;
3. Questions where the accepted answer has also the highest score by the SO users, reflecting agreement between the author and the remaining users;
4. Questions tagged with `Java`;
5. Questions that do not contain images (i.e., no `<img>` tags);
6. Questions that include at least one code snippet (i.e., contain the `<code>` tag), in order to evaluate ChatGPT's ability to choose answers for questions involving source code and explanations.

The application of these filters resulted in the selection of 23,829 SO question candidates, which constitute the total sample of this study.

2.2. Selecting the Analyzable Sample

Due to the token-based cost of using ChatGPT, this work applies a sampling strategy to keep the study feasible in terms of time and resources, as shown in Step 2 of Figure 1.

From the initial set of 23,829 SO questions, we applied stratified random sampling by year, selecting 5% of the questions from each year. This approach ensures temporal representativeness while reducing the dataset to a manageable size [Cochran 1977].

As a result, the final analyzable sample consists of 776 questions.

2.3. Prompting ChatGPT

In this subsection, shown in step 3 of Figure 1, ChatGPT was used to determine, among the 776 questions in the sample, which of the answers provided was the most accurate. The prompts were constructed using the *Langchain* library and executed with the GPT-4.1

model, configured with a *temperature* value of 0.0 to minimize randomness in responses [Langchain].

The context was defined using a *system* prompt, as shown in the textbox below. In this prompt, `[QUESTION_TITLE]` refers to the title of the question. The field `[QUESTION_BODY]` contains the body of the question, `[ANSWER_ID_TEXT]` which includes five possible answers, and `[QUESTION_TAGS]` contains provided as an additional context.

Template Used to Prompt ChatGPT

System: "You are a software engineer. Your task is to carefully read a Java programming-related question and five possible answers and select the one that most accurately addresses the question. Respond with only the ID of the most accurate answer."
User: "You will receive the title, the question tags, and body of a Java programming question, followed by five possible answers. Your task is to identify which answer most accurately addresses the question. Return only the ID of that answer. No explanation, no formatting, and no additional text.
Question Title: `QUESTION_TITLE`
Question Tags: `QUESTION_TAGS`
Question: `QUESTION_BODY`
Possible Answer ID = 1: `ANSWER_1_TEXT`
Possible Answer ID = 2: `ANSWER_2_TEXT`
Possible Answer ID = 3: `ANSWER_3_TEXT`
Possible Answer ID = 4: `ANSWER_4_TEXT`
Possible Answer ID = 5: `ANSWER_5_TEXT`"

This work limited the input to five possible answers from each SO post, due to the token constraints of the language model's input context. To select the answers to be evaluated, the following were chosen: (i) the answer marked as accepted by the question author and highest score by the SO users; (ii) the answer lowest score by the SO users; (iii) one answer with a medium (intermediate) score; and (iv) two randomly selected answers. To avoid positional bias, the order of the five answers was randomized in each prompt. This ensured that the choice of ChatGPT was based only on content, not on the answer position.

To ensure clarity and consistency, we tried prompts with various combinations to identify the optimal combination. For example, in one of the early prompt versions, each answer option was labeled with its actual SO post ID rather than a simple sequential number from 1 to 5. With this approach, we observed that in 96 of the 776 questions (12.37%), ChatGPT returned an answer ID that was not among the five options explicitly listed on the prompt. Interestingly, these IDs corresponded to other valid answers to the same SO question, indicating that the model recognized the source of the question and independently retrieved an alternative answer from the same thread. This unexpected behavior revealed the need to strictly limit the focus of the model to the provided options, leading us to revise the prompt and use simple numerical labels (1 to 5) instead of actual post IDs.

To address concerns about the stability of the response, the prompts were executed five times, always using the same configuration. In all runs, the number of accepted answers identified by ChatGPT remained highly consistent, with only negligible variation. This behavior reinforces the deterministic nature of the model in this configuration.

2.4. Assessing ChatGPT Divergent Answers

As mentioned in the previous section, each SO question submitted to ChatGPT included five possible answers, with one being both accepted by the question's author and highest score by the SO users. However, the other four answers are not necessarily incorrect, as a single SO question can often have multiple valid responses.

To evaluate the SO questions in which ChatGPT selected one of the four lower score answers, we performed a manual assessment of its responses. Two evaluators (the first author with more than five years of Java experience and his advisor with more than 15 years) rated how well each answer aligned with the question using a 5-point Likert scale based on the following criteria.

1. **Unrelated**: ChatGPT selected an answer that is not related to the question.
2. **Weakly Related**: ChatGPT selected an answer that does not objectively address the question.
3. **Related**: ChatGPT selected an answer that is generally related to the question, but requires substantial modifications to be considered a proper solution.
4. **Understandable**: ChatGPT selected an answer that partially addresses the question, requiring only a few adjustments to fully resolve the question.
5. **Straightforward**: ChatGPT selected an answer that directly and completely addresses the question.

Afterward, the average rating was calculated for each answer. If the two scores differed by more than one Likert point and at least one of them was greater than 3, the answer was marked for reevaluation. If disagreement persisted after discussion, the final score was determined by averaging the two evaluations. Each final rating was classified into one of two groups: low to moderate (Likert 1, 2, or 3) and high (Likert 4 or 5).

3. Results

The results are presented to address the research question.

Table 1. Distribution of Answer Types Selected by ChatGPT

Answer Type	Count	Percentage
Accepted and Highest Score Answer	433	55.80%
Intermediate Answer ID	113	14.56%
First Random Answer ID	108	13.92%
Second Random Answer ID	75	9.66%
Lowest Score Answer ID	47	6.06%
Total	776	100%

3.1. ChatGPT Prompts

Table 1 presents the distribution of the answer types selected by ChatGPT for the 776 SO questions. In 433 questions (55.80%), ChatGPT selected the accepted answer with the highest score from SO users. In the remaining 343 questions (44.21%), the model selected a different answer. Among these, the lowest score given by SO users was also the least selected by ChatGPT, accounting for only 6.06% of the total selections. A *chi-square* test confirmed that ChatGPT selected accepted answers significantly more often than other types (p-value < 0.01)

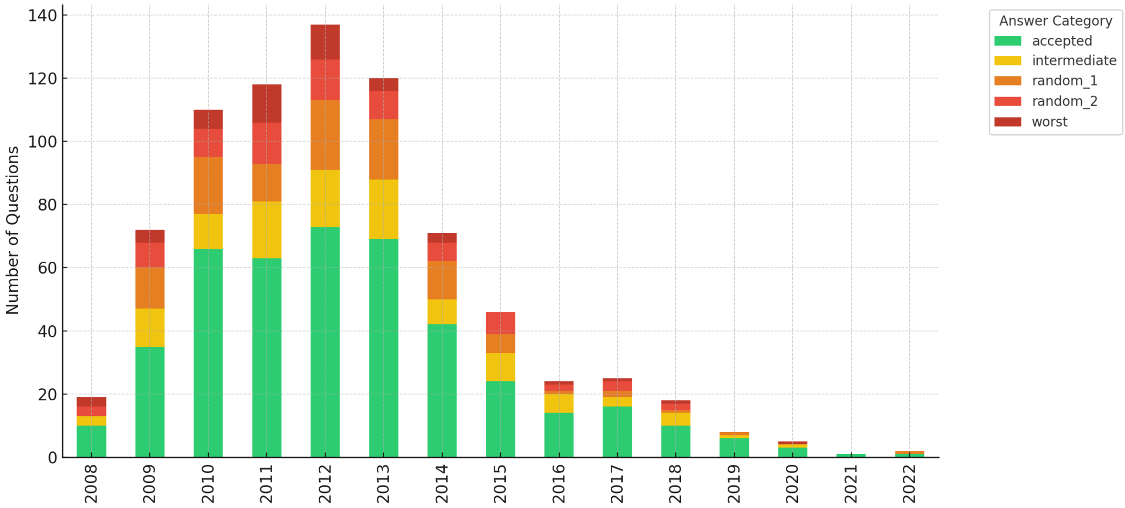
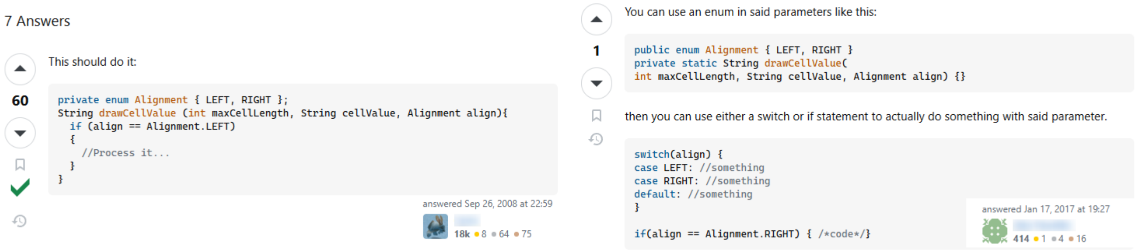


Figure 2. Distribution of ChatGPT answer types by year

Figure 2 shows the distribution of the answer categories selected by ChatGPT over the years. Although accepted answers were consistently the most selected, all categories appeared throughout the years, indicating that such choices are not limited to older or newer questions.



7 Answers

60

This should do it:

```
private enum Alignment { LEFT, RIGHT };
String drawCellValue (int maxCellLength, String cellValue, Alignment align){
  if (align == Alignment.LEFT)
  {
    //Process it...
  }
}
```

answered Sep 26, 2008 at 22:59

[redacted]

18k ● 8 ● 64 ● 75

(A) Accepted and Highest Score

1

You can use an enum in said parameters like this:

```
public enum Alignment { LEFT, RIGHT }
private static String drawCellValue(
int maxCellLength, String cellValue, Alignment align) {}
```

then you can use either a switch or if statement to actually do something with said parameter.

```
switch(align) {
case LEFT: //something
case RIGHT: //something
default: //something
}

if(align == Alignment.RIGHT) { /*code*/}
```

answered Jan 17, 2017 at 19:27

[redacted]

414 ● 1 ● 4 ● 16

(B) Selected by ChatGPT

Figure 3. The answer selected by the Stack Overflow (SO) users, and the answer selected by ChatGPT.

However, after manually reviewing the 343 questions in which ChatGPT diverged from SO users, we identified cases in which ChatGPT selected answers that were more recent than accepted ones. Figure 3 illustrates an example in which the question was published in SO in September 2008.[1] The answer that was later accepted by the author was published the same day. In contrast, the selected answer by ChatGPT was published in January 2017, nearly nine years later.

[1] https://stackoverflow.com/questions/142420/java-enum-parameter-in-method

To investigate the incidence of such cases, we queried the *votes.xml* data from [Stack Overflow Dump] to extract the date on which each answer was accepted by the author of the question. As illustrated in Table 2, ChatGPT recommended answers that were more recent than the author's acceptance date in 107 questions, older in 94 questions, and with the same date in 142 questions. In other words, in 31.2% (107 out of 343) of the questions where ChatGPT diverged from the SO users, it selected an answer that did not yet exist when the author of the question chose the accepted answer.

We then compared the score between the 107 questions in which ChatGPT recommended answers that were more recent than the acceptance date. We extracted the number of upvotes and downvotes for both the answer selected by ChatGPT and the accepted answer by SO users, considering only cases where both answers already existed at the time of comparison.

Table 2. Comparison between the date of the answer selected by ChatGPT, the accepted answer date, and the author acceptance date

Comparison	Accepted Answer Date		Acceptance Date	
	Questions	%	Questions	%
ChatGPT selected is more recent	121	35.3%	107	31.2%
ChatGPT selected is older	19	5.5%	94	27.4%
Same date	203	59.2%	142	41.4%
Total	343	100%	343	100%

A Wilcoxon signed rank test was performed between the two samples (p-value < 0.01), revealing a statistically significant difference in scores. For example, in the case illustrated in Figure 3, after January 2017, the accepted answer received 41 upvotes and 4 downvotes (score of 37), while the ChatGPT-selected answer received only 1 upvote. In other words, even when comparing within the same period, there is still a significant difference in scores. However, since the accepted answer appears at the top of the SO page and receives greater visibility, having fewer scores could not necessarily indicate that the ChatGPT-selected answer has a lower quality.

3.1.1. Classification of the Divergent Answers

To assess the accuracy of ChatGPT's selections in divergence cases, a Likert scale evaluation was applied, as detailed in Section 2.4. To evaluate whether the highest Likert values (4 and 5) appeared significantly more frequently than the lower values (1, 2, and 3), we grouped the responses accordingly and performed a *chi-square* test. The results show that 4 and 5 occurred 235 times (evaluators average rating), while 1, 2, and 3 occurred 108 times. A *chi-square* test revealed that this difference is statistically significant (p-value < 0.01) , indicating that, according to the evaluators, ChatGPT often chooses accurate SO answers.

However, in many cases, particularly among the 68.8% of divergence cases where both the ChatGPT-selected answer and the accepted answer were already available at the time of acceptance, some factors may have influenced the author's choice. These include not only the quality of the explanation itself but also subjective preferences. For example, in a question, the author asked for assistance in comparing two dates.[2] The accepted answer suggested using the *Joda Time* API, while the ChatGPT selected answer provided a solution using the standard Java API. In other cases, the accepted answer contains images to facilitate comprehension, which was not evaluated by ChatGPT.[3]

4. Threats to Validity

Regarding construction validity threats, one concern is the lack of prior evaluation to verify whether the answer accepted by both the question author and the SO users is indeed the most accurate. To mitigate this threat, we only included SO questions in which both the author and the remaining users (highest score) selected the same answer. This strategy aims to reduce the risk of subjective answer selection and to increase confidence in the accuracy of the accepted answer.

Another validity threat of the construct is the limitation of each prompt to only five possible answers. To mitigate each prompt, we combined the accepted answer with a mix of low-score, intermediate, and randomly selected answers. This ensured the diversity of possible answers while keeping the prompt within the token limit of the model.

Regarding external validity threats, one limitation is the restriction of the analysis to questions tagged with the Java programming language. This decision was made primarily for qualitative analysis. Since SO includes questions in multiple programming languages, expanding the scope could have introduced challenges in consistently interpreting and evaluating the divergences between ChatGPT and SO answers, thereby affecting the reliability of the analysis.

5. Related Work

This section discusses related work addressing misalignments between LLMs and human judgment, as well as studies that inspired the methodological approach adopted in this research.

Bifolco et al.[Bifolco et al. 2025] investigated the reliability of the code and references generated by ChatGPT. Their study found that many of the suggested links and references lack proper provenance or do not exist. This raises concerns about the reliability of the content generated by LLMs. This work has a similar motivation, but instead of validating the suggested links, it assesses the alignment between the ChatGPT and the users on SO questions and answers.

Tamanna et al. [Tamanna et al. 2025] investigated ChatGPT performance when answering technical questions derived from bug reports in open-source projects. Even when using retrieval-augmented generation (RAG), ChatGPT correctly answered only 36.4% of the questions, mainly due to difficulties in interpreting stack traces and integrating technical context. They proposed a tool that preprocesses the input and improves accuracy. Although this work does not propose tools to improve the accuracy of ChatGPT responses, it includes a discussion of how different prompt formulations, based on previous experiments, can influence the response of the model.

Dantas et al. [Dantas et al. 2023] assessed the readability of ChatGPT generated code snippets compared to human-written code from SO. To evaluate whether the gen-

[2] https://stackoverflow.com/questions/32372279/generating-all-days-between-2-given-dates-in-java

[3] https://stackoverflow.com/questions/31696439/how-to-build-a-docker-container-for-a-java-application

erated snippets were semantically aligned with the input queries, the authors employed a 5-point Likert scale. This work uses the same scale for a different purpose, i.e., to evaluate ChatGPT judgment in selecting the most accurate answer among five candidate answers retrieved from SO.

6. Conclusion

This study evaluates the alignment of ChatGPT with human judgment based on 776 SO questions. The main findings are presented below.

1. ChatGPT and SO users agree on 56% of the questions, with divergences occurring in the remaining 44%.
2. In 31% of the divergence cases, the answer chosen by ChatGPT was not available at the time the SO users selected the accepted answer. However, these ChatGPT-selected answers tend to receive significantly lower user scores.
3. A manual analysis of the divergences shows that ChatGPT generally selects accurate answers, which may have been overlooked by SO users due to subjective factors, such as the use of specific Java libraries or additional explanations that go beyond plain text, including figures or external links.

This work presents opportunities for future improvement. For example, the research could be extended to include an open coding analysis to extract potential patterns on the divergences between ChatGPT and SO users. Future studies could also incorporate a wider range of LLMs, such as Gemini and DeepSeek, to compare their performance.

References

Beyer, S., Macho, C., Di Penta, M., and Pinzger, M. (2020). What kind of questions do developers ask on stack overflow? a comparison of automated approaches to classify posts into question categories. *Empirical Softw. Engg.*, 25(3):2258–2301.

Bifolco, D., Cassieri, P., Scanniello, G., Di Penta, M., and Zampetti, F. (2025). Do llms provide links to code similar to what they generate? a study with gemini and bing copilot. In *IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, pages 223–235.

CHATGPT (2025). Chatgpt web page. `https://chatgpt.com/`. Accessed: 2025-02-20.

Cochran, W. G. (1977). *Sampling Techniques*. John Wiley & Sons, New York, NY, 3rd edition.

Dantas, C. E., Rocha, A. M., and Maia, M. A. (2023). Assessing the readability of chatgpt code snippet recommendations: A comparative study. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*, SBES '23.

Ebert, C. and Louridas, P. (2023). Generative AI for software practitioners. *IEEE Software*, 40:30–38.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Langchain. `https://www.langchain.com/`. Accessed: 2025-02-20.

May, A., Wachs, J., and Hannák, A. (2019). Gender differences in participation and reward on Stack Overflow. *Empirical Software Engineering*, 24(4).

Nam, D., Macvean, A., Hellendoorn, V., Vasilescu, B., and Myers, B. (2024). Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA. Association for Computing Machinery.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2018). Language models are unsupervised multitask learners.

Reis, F. A. G., Maia, M. M., and Dantas, C. E. C. (2025). Replication Package. `https://doi.org/10.5281/zenodo.15750158`.

Sobania, D., Briesch, M., Hanna, C., and Petke, J. (2023). An Analysis of the Automatic Bug Fixing Performance of ChatGPT . In *2023 IEEE/ACM International Workshop on Automated Program Repair (APR)*, pages 23–30, Los Alamitos, CA, USA. IEEE Computer Society.

Stack Overflow Dump. `https://archive.org/details/stackexchange`. Accessed: 2025-02-20.

Stack Overflow. `https://stackoverflow.com`. Accessed: 2025-02-20.

StackExchange (2025). Stackexchange web page. `https://data.stackexchange.com/stackoverflow/queries`. Accessed: 2025-02-20.

Tamanna, S. B., Uddin, G., Wang, S., Xia, L., and Zhang, L. (2025). Chatgpt Inaccuracy Mitigation During Technical Report Understanding: Are we There Yet? . In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 2290–2302, Los Alamitos, CA, USA. IEEE Computer Society.

Tufano, R., Mastropaolo, A., Pepe, F., Dabić, O., Penta, M. D., and Bavota, G. (2024). Unveiling ChatGPT's usage in open source projects: A mining-based study.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Zhang, Z., Xing, Z., Zhao, D., Xu, X., Zhu, L., and Lu, Q. (2024). Automated refactoring of non-idiomatic python code with pythonic idioms. *IEEE Transactions on Software Engineering*, PP:1–22.

Zuccon, G., Koopman, B., and Shaik, R. (2023). Chatgpt hallucinates when attributing answers. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP '23, page 46–51, New York, NY, USA. Association for Computing Machinery.