Human Evaluation of GPT-4’s Answers to Stack Overflow Questions

June 6, 2023

David Andrews
dha@xoba.com

Abstract

The rapid development of large language models (LLMs), such as GPT-4, has raised questions about their potential to replace traditional online resources for content generation in various domains. In the context of software development, platforms like Stack Overflow have served as the primary source for answering technical questions. This study aims to compare the technical correctness and quality of GPT-4′s answers with human-authored responses to Stack Overflow questions. Using a survey-based research design, we collected data from 85 software developers on their preferences between human and GPT-4 generated answers. Subsequent application of a mixed-effects logistic regression model revealed that participants had a 58.7% chance of preferring GPT-4′s answers to human answers. The results suggest that emerging technologies such as GPT-4, which can answer technical questions more rapidly while maintaining quality, may reduce the market share of traditional online resources like Google and Stack Overflow in the future. This study contributes to the growing body of literature on LLMs and provides insights for future research on improving alignment techniques and verifying model outputs.

1. Introduction

1.1. ChatGPT

ChatGPT, a model released as a public research demo in November 2022, has skyrocketed in popularity in the last few months, garnering one million users in the first 5 days after its launch, solidifying its position as the fastest growing app in history [1]. ChatGPT belongs to a class of models called large language models (LLMs), which are based on transformer models, a type of machine learning model. Essentially, LLMs are trained on large corpuses of data from the Internet and are optimized for predicting the next word in a sequence of words. This process of predicting the next word is repeated indefinitely until the model reaches the end of a sequence. At each step of the iteration, the model produces a probability distribution of all words, which is then randomly sampled. As such, words with high probabilities of coming next in a sequence will be sampled with a higher probability than lower probability words [2, 3].

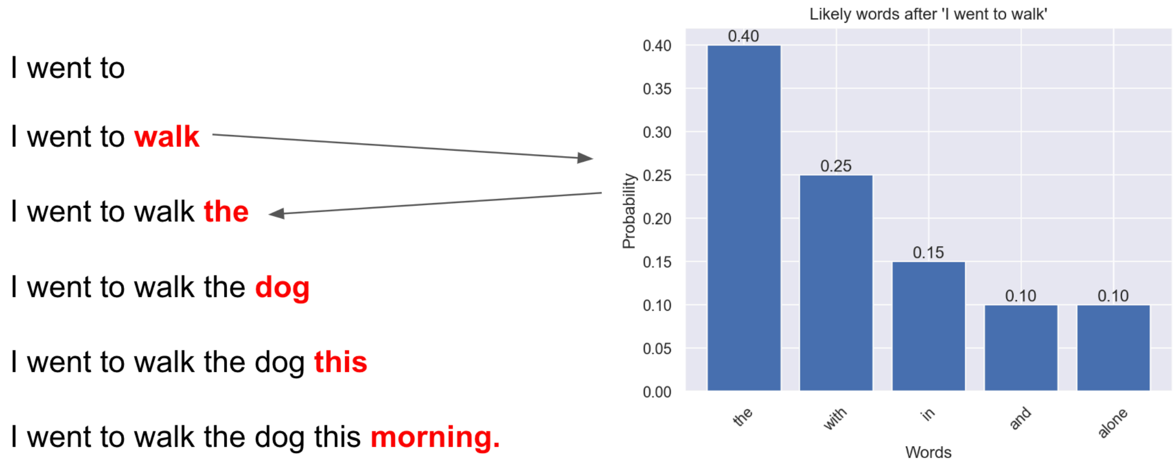


Figure 1: An intuition behind how transformers generate sequences token-by-token

In this way, models such as ChatGPT can learn to produce coherent text, mimicking text from its training dataset. In addition to this process, ChatGPT was further trained using OpenAI's InstructGPT process, allowing humans to grade the model's outputs to improve their quality. This process, called reinforcement learning with human feedback (RLHF), aligns the model's answers to human preferences. The effect of this process is that the model's outputs become more detailed, more useful, and the model can hold conversations better in a chat interface [4].

1.2. Other LLMs

Before ChatGPT, many years of research into the space of LLMs was conducted. ChatGPT is part of a lineage of other GPT (Generative Pre-trained Transformer) models, such as GPT-2 and GPT-3. Through scaling up the sizes of models over time and increasing the training data size, researchers were able to show that this could improve performance across a diverse range of tasks, which included reading comprehension, translation, question-answering, news article writing, etc. [5, 6]. The typical ways that these models are evaluated and compared to each other are based on standardized benchmarks. Along with accuracy on the training dataset and human evaluation, some of these benchmarks include HellaSwag, a common sense reasoning benchmark [7], StoryCloze, a task involving choosing the correct ending to a short story [8], and testing for theory of mind [9].

Recently, with the release of GPT-4 in March 2023, researchers have begun to use human standardized tests along with previous methods to evaluate the model's performance. Overall, it has been seen from preliminary research that GPT-4 outperforms all previous GPT models and scores highly in AP tests as well as the Uniform Bar Exam, scoring in the top 90th percentile while ChatGPT could only perform in the bottom 10th percentile [10, 11]. It scores at a human level in common sense reasoning tasks such as HellaSwag and has a theory of mind comparable to a seven year old, quantifying its ability to converse effectively with humans [9].

1.3. Issues

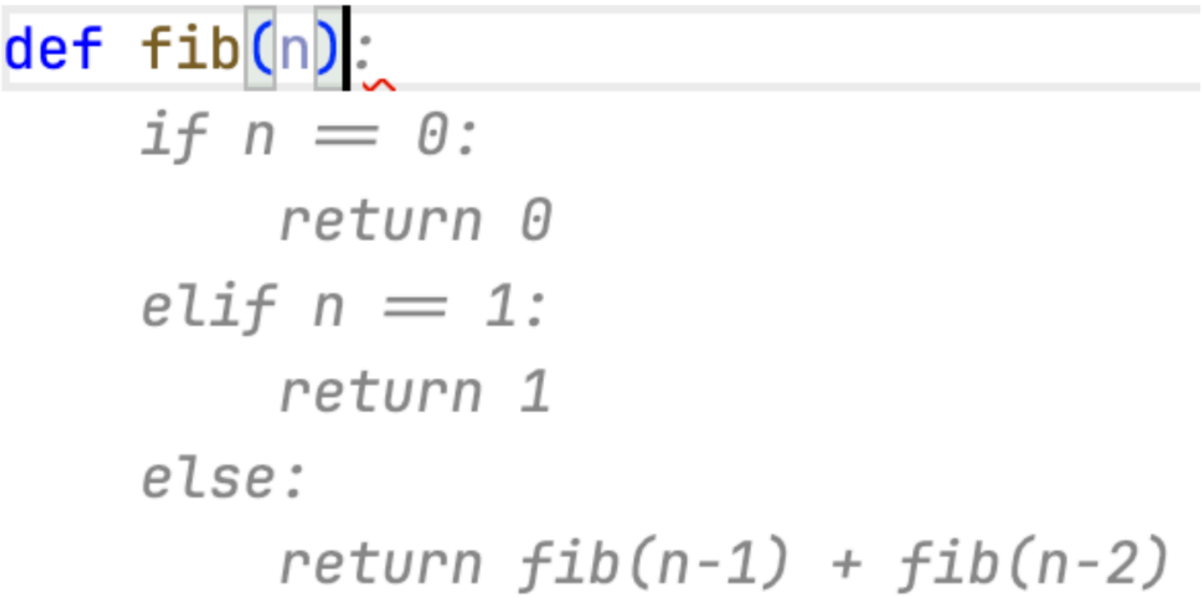
Despite the enormous progress in the capabilities of these models, the way that these models are trained and how they function present several limitations to their use. Regarding the process of predicting the next word in a sequence, it is fundamentally a probabilistic process, meaning that the model may produce disinformation, or hallucinate, if an incorrect, low probability word is selected at some point in writing the response. This hallucination is often exaggerated when the question is technical and requires complex reasoning and analysis [6]. Moreover, the process that newer LLMs are being trained with, RLHF, creates unexpected outcomes. As RLHF encourages the model to maximize its expected rating by humans, it may create factually incorrect answers that seem correct on the surface, as the model is purely maximizing the probability of receiving a high score from humans rather than maximizing factually accurate answers. In this way, models based on RLHF are fundamentally limited as they are based on the opinions of humans.

1.4. Banning of ChatGPT

With this potential for models to hallucinate inaccurate answers to technically complex problems and make its answers appear correct through RLHF, deceiving humans in the process, online technical software development forums such as Stack Overflow were swift to ban the use of ChatGPT to answer user questions on the forum, something that became commonplace in the first few days after ChatGPT's release [12]. With this new ruling, however, a major question arose: can LLMs effectively help humans during the software development process?

1.5. Code Generation Models

One method for aiding software development is through the use of code generation models. These models primarily help people by acting as an advanced autocomplete engine for code, being able to cut down on the amount of code that software engineers have to write manually.



```
def fib(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)
```

Figure 2: An example of how transformers trained on source code, such as Github Copilot, can be used to in code editors for advanced code completion.

Many solutions have been developed, such as models that automatically generate comments to code [13], ones that integrate into modern code editors [14], and ones that summarize code [15]. While these solutions provide developers with much assistance during the development of software, they are still fundamentally limited by the fact that they are not based on natural language and cannot answer questions about programming in the same way that is done on Stack Overflow.

1.6. Retrieval Models

Technical question-answering models, which focus on answering natural language questions related to software development, are more flexible than code-generation models due to their ability to work with natural language. Two main classes of solutions exist for this type of model, one being retrieval-based models. One such model, “doc2vec” [16] aligns questions and answers as embedding vectors in the same vector space, and a classifier model is trained to predict the likelihood of an answer being preferred for a question. Several answers are then ranked by this preference probability and presented to the user. Another similar model, “AnswerBot” [17], expands on this previous work, by combining several highly probable answers from its database using a summarization model before presenting the result to the user.

The key concept behind retrieval-based models is the usage of large databases of human-generated question-answer pairs to retrieve an effective answer for a given question. The main advantage of systems like this is that the answers have all been closely inspected by humans, as they were written by humans, providing some assurance of quality and accuracy. However, a major caveat is their inability to adapt to new, previously unasked questions. If prompted with an unfamiliar question that lacks a matching answer in the database, the system will return an irrelevant response.

1.7. Generative Models

To achieve greater adaptability in question-answering, generative components can be incorporated into models. As demonstrated by "AnswerBot", using a summarization model to combine answers into a more relevant and concise format, one that did not previously exist in the database, yielded more diverse and useful results than sites like Google and Stack Overflow, showing significant potential for generative models to improve question-answering [17]. One such hybrid model, ReTrans, combines a typical retrieval model for obtaining relevant answers with a generative model to produce an answer to a question. A discriminator model then selects the best answer of the two models, leveraging both the accuracy of retrieval models and the flexibility of generative models [18].

Furthermore, purely generative models exist, models which do not rely on a database for answers, but instead generate new responses to questions. This research area is relatively new, with fairly recent studies showing that transformer models such as GPT-2 trained on Stack Overflow questions could produce grammatically and syntactically correct answers but with limited technical and semantic accuracy. According to the researchers, the small model size of GPT-2 constrained its ability to effectively answer technical questions [19]. However, current LLMs, like ChatGPT and GPT-4, are much larger and more general, enabling them to answer questions with a higher degree of accuracy than their predecessors, opening up the possibility for them to be able to answer technical questions effectively.

1.8. Gap

Overall, there has been limited research on how purely generative models compare to human-generated responses for technical software development questions. Given that the current state-of-the-art LLM, GPT-4, was released in March 2023, there is a significant knowledge gap regarding its ability to accurately answer technical questions. As a result, the following research question can be asked: Can GPT-4 provide useful and meaningful answers to Stack Overflow questions?

2. Method

2.1. Design

In my study, I employed a descriptive, survey-based research design in which participants were tasked with comparing GPT-4 generated answers with human answers to Stack Overflow questions to assess the perceived quality of GPT-4's answers to technical questions. In order to mitigate bias, the source and authorship of answers was not revealed to the participants. The primary goal of the study was to determine the probability of developers preferring GPT-4 answers over human answers, drawing on a similar design to previous studies that evaluated humans' ability to differentiate between LLM and human-written texts [6]. Although my approach slightly deviates by asking participants to choose the answer which they prefer rather than asking them to correctly label the authors of the answers, it effectively gathers results to see how LLM-generated responses truly compare to those created by humans in terms of quality.

2.2. Procedure

2.2.1. Initial Data Processing

To collect the Stack Overflow questions and answers that would be used for the survey, a collection of 58,329,357 Stack Overflow posts were retrieved from the Stack Overflow data dump, an online archive of all Stack Overflow posts up until March 6th, 2023. The data was then processed before use in the survey using a Python program which did the following:

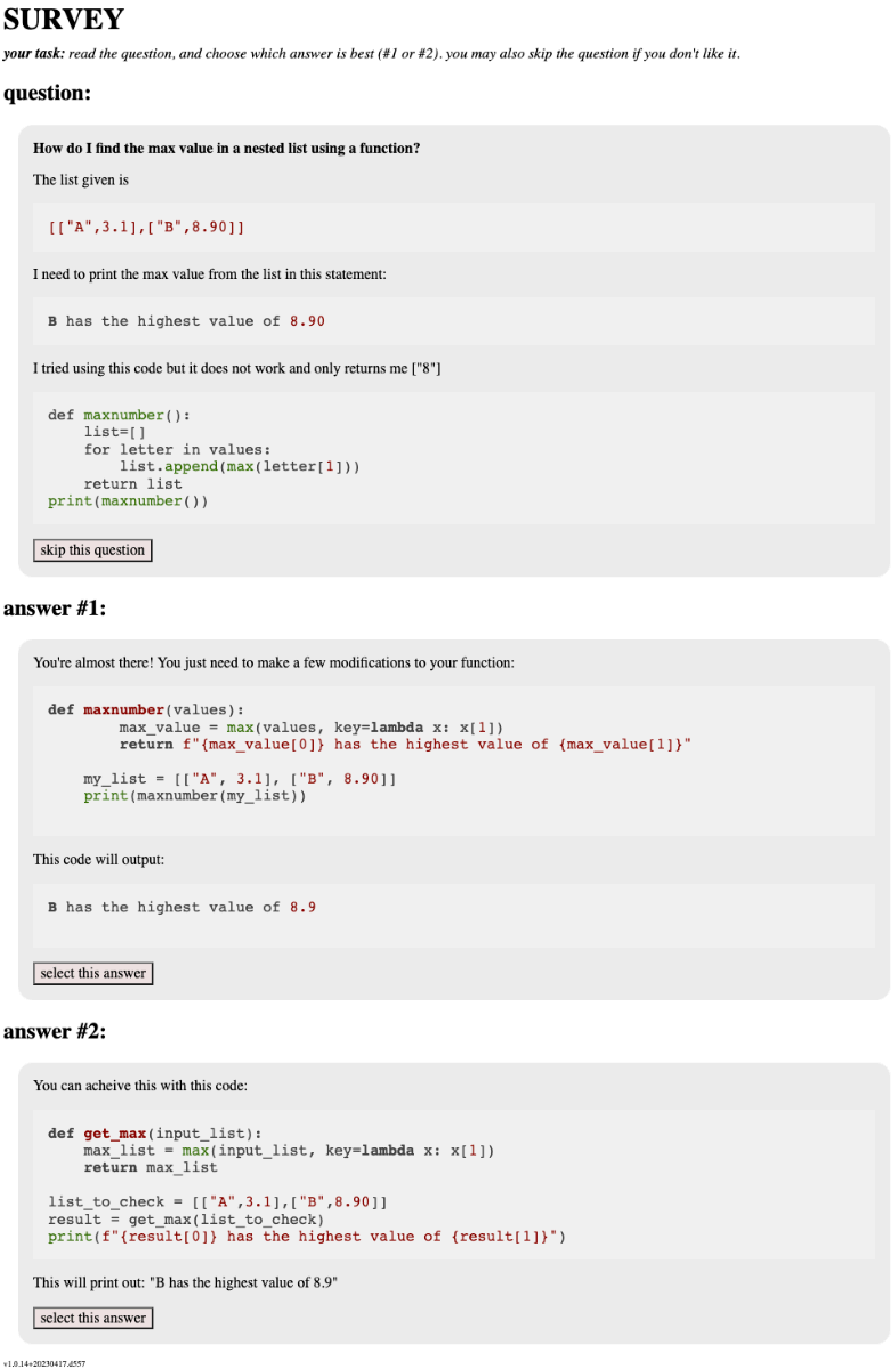
1. All posts from before September 30, 2021, GPT-4's training data cut off, were removed from the dataset in order to make sure that GPT-4 had not seen the questions during training, avoiding memorization bias and making sure that GPT-4 was answering novel questions that it had not seen before.
2. All questions with no answers were removed from the dataset as at least one human answer was needed to compare to GPT-4's answer.
3. All answers other than the chronologically first in each thread were removed to make sure that the Stack Overflow user answering the questions was not advantaged in any way over GPT-4, being able to see previous users' submissions and potentially building from them or exploring new ideas. By selecting the first answer to a question, I made sure that the human's answer is similar to GPT-4's in that the human did not have access to any other answers.
4. Question-answer pairs that had links or images were removed as GPT-4 does not have access to the Internet to access links present in questions, nor can it view images as it was not enabled in the API during the time the research was conducted.
5. Question-answer pairs without the Python tag were removed as the domain of the survey was narrowed in order to make sure that most participants were familiar with the topics discussed in the questions and answers. Without this narrowing of scope, allowing for all content on Stack Overflow, the probability of participants knowing the content on any particular question and answer would be slim to none. With Python being one of the world's most popular languages currently, it was an easy choice to attract a large number of participants and obtain higher quality results [20].

After this initial data processing, the final Stack Overflow dataset consisted of 136,201 question-answer pairs, a 99.77% reduction from the original data dump size.

2.2.2. Survey

A survey platform was created which allowed me to collect preferences of users. The survey platform was developed in the Go programming language and was hosted on the Internet. The participants were first shown a consent form (Appendix A) which informed them about the study, purposely omitting reference to GPT-4 or Stack Overflow to make sure that users were not biased in their assessment of answers. The consent form also emphasized the fact that participants should focus on the quality of their selections rather than the quantity of preferences they submitted in order to ensure that participants submitted more technically correct responses. For instance, if a participant did not know which answer to select to a question, rather than selecting a random answer, users were allowed to skip the page.

After the consent form was accepted, participants were shown a randomly sampled Stack Overflow question from the filtered dataset at the top of the page, the first answer below the question, and the second answer below the first answer. Importantly, one of the answers was generated by GPT-4 and the other answer was written by a human. Whether the first answer or the second answer was generated by GPT-4 or a human was randomized on each page to make sure that answer ordering bias was eliminated. The participant, when they deemed one answer to be better than the other, could select the answer and continue to the next survey page, where they were shown another question and two answers.



SURVEY

your task: *read the question, and choose which answer is best (#1 or #2). you may also skip the question if you don't like it.*

question:

How do I find the max value in a nested list using a function?

The list given is

```
[["A",3.1],["B",8.90]]
```

I need to print the max value from the list in this statement:

```
B has the highest value of 8.90
```

I tried using this code but it does not work and only returns me ["8"]

```
def maxnumber():
    list=[]
    for letter in values:
        list.append(max(letter[1]))
    return list
print(maxnumber())
```

answer #1:

You're almost there! You just need to make a few modifications to your function:

```
def maxnumber(values):
        max_value = max(values, key=lambda x: x[1])
        return f"{max_value[0]} has the highest value of {max_value[1]}"

    my_list = [["A", 3.1], ["B", 8.90]]
    print(maxnumber(my_list))
```

This code will output:

```
B has the highest value of 8.9
```

answer #2:

You can acheive this with this code:

```
def get_max(input_list):
    max_list = max(input_list, key=lambda x: x[1])
    return max_list

list_to_check = [["A",3.1],["B",8.90]]
result = get_max(list_to_check)
print(f"{result[0]} has the highest value of {result[1]}")
```

This will print out: "B has the highest value of 8.9"

v1.0.14+20230417.d557

Figure 3: A sample survey page that the participants were given.

2.2.3. GPT-4 Prompt

The prompt given to GPT-4 for each survey page was the following:

```
Make believe you are a typical user on https://stackoverflow.com, answering a question
there. Your goal is to give the best answer you can in the style of typical answers.
Do not include URL's in your answer.

Here is the question (as an HTML document) for you to answer:

[question]

The following will be your answer, also in HTML (no markdown), making use of customary
elements when appropriate, including <p> for paragraphs, <li> for list items, wrapping
code in <pre> and <code>, etcetera. Your answer should be approximately [number of
characters in human answer] characters, not including HTML tags.
```

In the prompt, GPT-4 was instructed to behave like a typical human on Stack Overflow, answering the question in correct HTML format for the survey along with making sure that its response was approximately the same number of characters as the human answer in order to remove bias due to the answer length.

2.2.4. Distribution

The survey was distributed partially through Fiverr and mostly through Amazon Mechanical Turk. Importantly, I ensured that participants from Fiverr advertised themselves as expert Python code reviewers and participants from Amazon Mechanical Turk were qualified IT specialists in software development and engineering. This was done to make sure that the preferences they made had a degree of accuracy based on their qualifications. The survey was active for two weeks on both platforms.

3. Results

In total, I received two responses from Fiverr and 83 responses from Amazon Mechanical Turk, with a count of 377 answer selections, or preferences, ranging from 1 to 92 selections per participant. The distribution of the number of preference submitted by users is shown below:

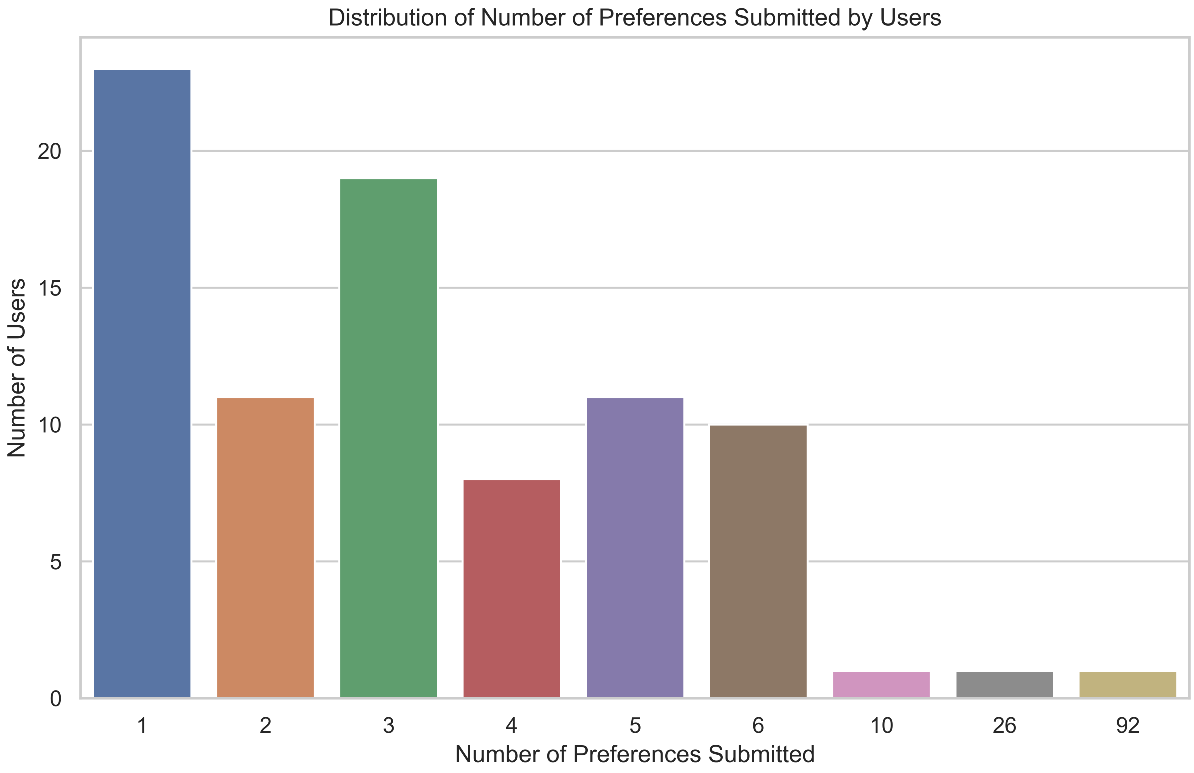


Figure 4: Distribution of hte number of preferences submitted by participants.

As can be seen from this distribution, most users submitted one preference, with a few users submitting 10 or more preferences. Users submitted a mean of 4.435 preferences and a median of three preferences. When looking at the proportion of selections that preferred GPT-4 compared to that of humans, GPT-4's answers were selected 238 times while human answers were selected 139 times.

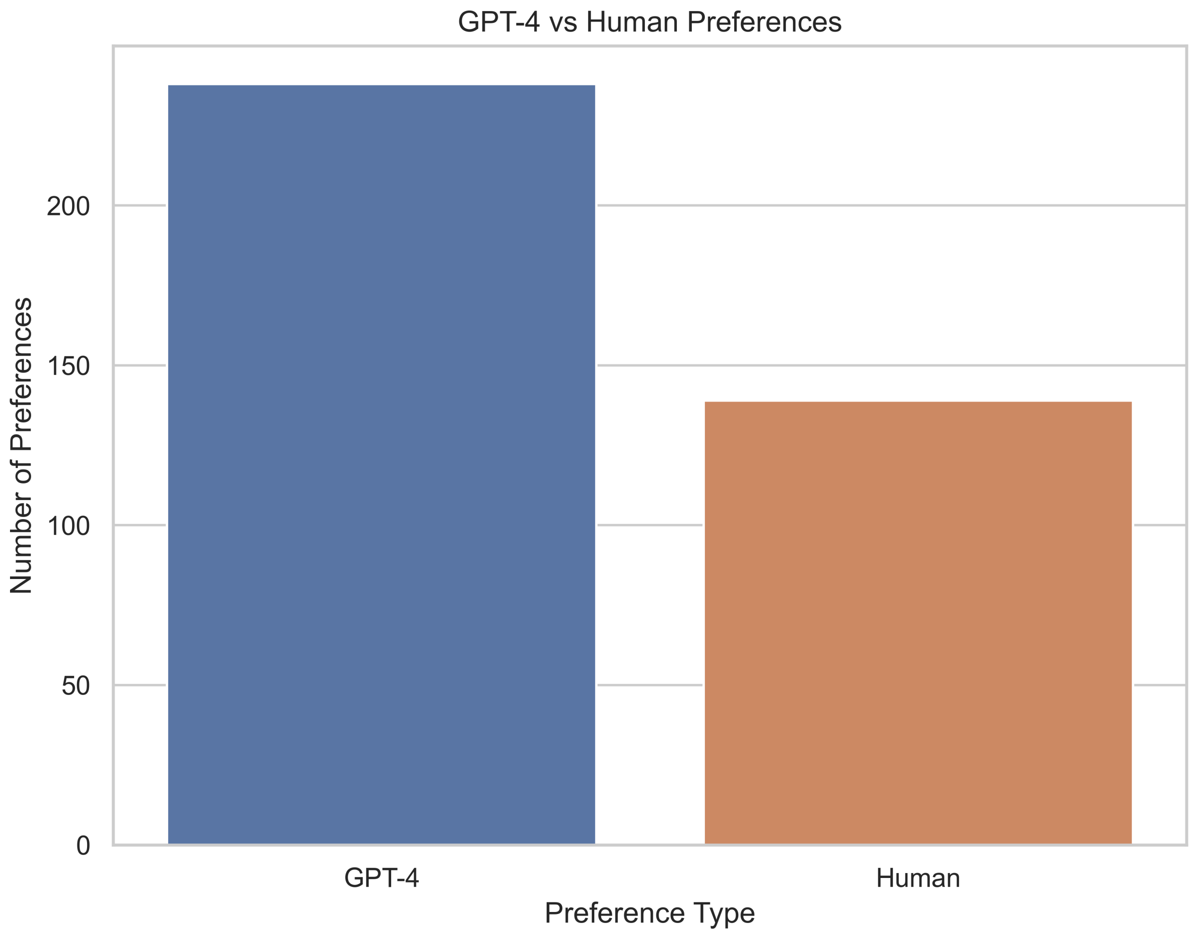


Figure 5: Proportion of all responses that were in favor of either the GPT-4 answer or the human answer.

While doing a simple single proportion z-test may seem enticing, it incorporates too much bias as it overweights the people who responded more than the people who responded fewer times. A potential way of mitigating this is to normalize each participant's results by the number of preferences submitted.

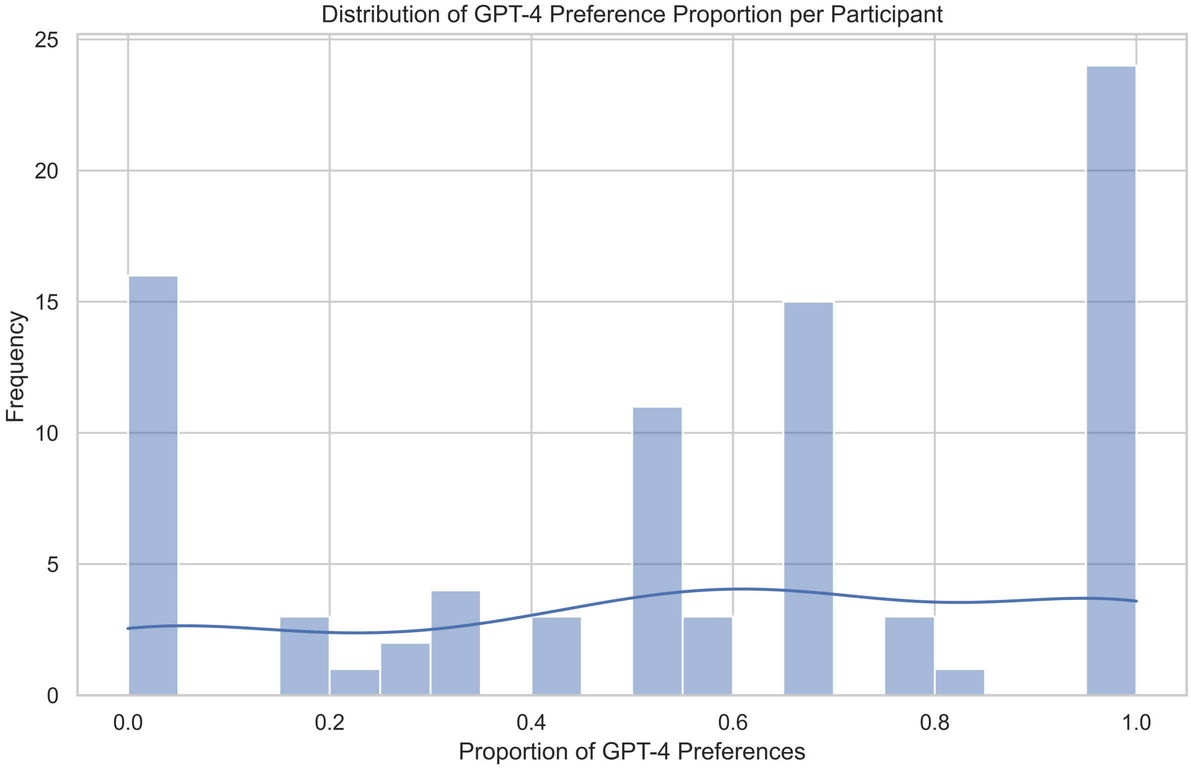


Figure 6: Preferences normalized to the number of responses per participant.

While this model seems to remove bias from participants who submitted large amounts of preferences, it does not account for the fact that participants who submitted only one response will not have their preference accurately estimated. Instead, the only conclusion that can be made about their preference is that they either prefer GPT-4's answers 100% of the time or human answers 100% of the time. This anomaly can be seen by the high bars at 0 and 1 in the above distribution.

One model that solves the issues found in the previous two models is the mixed-effects logistic regression model. This model is a modification of a typical logistic regression model in that it finds the probability of preferring GPT-4's answer over a human answer while also taking into account the individual bias present in the participants. Since the model takes into account these factors, a more accurate average preference can be found from the data.

After performing the regression, it was found that the intercept of the model was 0.352 ± 0.327, or as a probability, 0.587 ± 0.081 (p-value = 0.035). Additionally, the odds ratio found was 1.423 ± 0.398 and the variance and standard deviation for the random effect for users was 0.747 and 0.864 respectively.

Interpreting these results, since the p-value for the intercept is significant at a 5% level, we can conclude that software developers have a 58.7% probability of preferring GPT-4 answers over human answers. Based on the odds ratio, the odds of preferring GPT-4 answers were 1.423 times higher than the odds of preferring human answers. Furthermore, the high variation and standard deviation in the random effect for each user shows that people have varied preferences towards humans or GPT-4 (a visual of user-level probabilities for preferring GPT-4 is included in Appendix B). Importantly, participants who only submitted one answer, such as a participant who chose one human answer, have their preference more accurately reflected with this new model as while they may prefer human answers more on average, the model predicts that they do not prefer human answers 100% of the time and will prefer GPT-4 some of the time.

4. Discussion

4.1. Implications

What these results show is that resources and forums for software developers such as Google and Stack Overflow are at least matched and may be outpaced by technologies such as GPT-4. Considering that on average, GPT-4's answers will be preferred 58.7% of the time over human answers shows that GPT-4 has reached a comparable level in technical accuracy to humans for answering software development questions. Possibly in the future, larger and more powerful models will be able to further surpass humans in technical answering capabilities, overtaking Stack Overflow as the dominant question-answering service in the area. Currently, solutions such as GPT-4 have the potential to steal market share from Stack Overflow and Google purely due to the fact that GPT-4 can produce answers much faster than humans. While GPT-4 took only an average of 19.88 seconds to answer a Stack Overflow question, the first human response to a question on Stack Overflow takes many minutes to hours. With this drastic difference in response time and little to no quality difference between GPT-4's responses and human responses, it is not out of the question that sites such as Stack Overflow may be slowly replaced by this new technology.

Additionally, many new questions on Stack Overflow are often deleted due to them not following the correct etiquette expected of questions. This results in a hostile environment, especially to beginner programmers, who may not know how to describe the questions that they have perfectly to a standard, and are instead forced to search for people with similar questions to them that have received answers [21]. This situation mirrors the shortcomings of retrieval based models, which are not flexible enough to create answers to questions that have not been asked before. GPT-4 solves this problem by both being as correct as Stack Overflow questions most of the time while producing new answers to questions that have not been asked before. Moreover, GPT-4 can have real-time conversations with users, promoting better, more focused learning for beginners. As such, GPT-4 and future LLMs may become an invaluable resource for beginner programmers in the future.

4.2. Limitations

Despite careful work in order to eliminate bias and to extract accurate information from the collected data, there are several limitations to my study. One limitation is that I had no way of ensuring that participants did not select random answers on the survey. Since there was a monetary incentive to complete the survey, which is true on surveying platforms such as Amazon Mechanical Turk and Fiverr, this may have occurred. Additionally, participants may have selected answers to questions that they did not fully understand instead of picking the "skip" option on the survey. If this was the case, they may have selected the answer which appeared to be better on the surface, which is often the case with GPT-4's answers, as they always have correct grammar and are often presented in a more palatable way compared to the human answers. Furthermore, even if participants understood the question, they may have been persuaded to choose GPT-4's answer over the human answer due to GPT-4's RLHF training, which encourages the model to formulate answers that humans will likely prefer, regardless of factual accuracy. Another important limitation is that the prompt used for GPT-4 likely influenced the specific answers it wrote, making my findings likely not universally applicable for all prompts.

However, these limitations are likely not much of a concern as IT specialists and Python code reviewers were surveyed to specifically mitigate randomness in answer selection, and I was ultimately able to find statistically significant evidence that GPT-4's answers are equally matched or better than human answers most of the time. Even though the study cannot prove that GPT-4 is technically correct in all scenarios, the use of experts helps to give more assurance compared to random participants from the population.

5. Conclusion

The purpose of this study was to compare the technical correctness and quality of GPT-4's answers to that of human answers to Stack Overflow questions. Through the use of a survey method preference data was collected on 377 Stack Overflow questions from 85 software developers from Fiverr and Amazon Mechanical Turk. A mixed-effects logistic regression model was chosen to analyze the data effectively while mitigating bias, revealing that participants had a 58.7% chance of preferring GPT-4's answers to human answers. As a result, it can be predicted that services such as Google and Stack Overflow in the future will have their market share lessened due to emerging technologies such as GPT-4, which can answer technical questions in a fraction of the time while preserving answer quality. With Stack Overflow having a strict question etiquette that effectively restricts the use of the service to to beginners in the field, being unable to phrase a question to a certain standard, technologies such as GPT-4 may see especially high usage due to its fast answering time, potentially higher accuracy with beginner questions that do not have a high degree of technicality, and conversational aspect, allowing a beginner programmer to question the model more deeply than can typically exist on online forums. While the study I present contains several limitations, such as the fact that the selections from the participants may not have been completely accurate and GPT-4's prompt influencing the model's outputs in potentially unreproducible ways, the result determined by the study still holds some significance due to the copious number of countermeasures against bias which were employed. Future research should use this study as a stepping stone, potentially investigating methods of formally verifying GPT-4's answers, developing better alignment techniques which mitigate unexpected results, and testing larger models for accuracy using the technique described in this study as a measure of overall technical accuracy.

Bibliography

[1] K. Hu, "Chatgpt sets record for fastest-growing user base - analyst note," *Reuters*, 2023. [Online]. Available: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/

[2] A. Vaswani, N. Shazeer, et al., *Attention Is All You Need*, I. Guyon, U. V. Luxburg, et al., Eds., vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[3] OpenAI, "Chatgpt: optimizing language models for dialogue," OpenAI, 2022. [Online]. Available: https://openai.com/blog/chatgpt

[4] L. Ouyang, J. Wu, et al., "Training language models to follow instructions with human feedback," 2022.

[5] A. Radford, J. Wu, et al., *Language Models Are Unsupervised Multitask Learners*, 2019.

[6] T. Brown, B. Mann, et al., *Language Models Are Few-Shot Learners*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, 2020, p. 1877. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[7] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: can a machine really finish your sentence?," 2019.

[8] N. Mostafazadeh, N. Chambers, et al., *A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories*, 2016, p. 839, doi: 10.18653/v1/N16-1098. [Online]. Available: https://aclanthology.org/N16-1098

[9] M. Kosinski, "Theory of mind may have spontaneously emerged in large language models," 2023.

[10] OpenAI, “Gpt-4,” openai.com, 2023. [Online]. Available: https://openai.com/research/gpt-4

[11] OpenAI, “Gpt-4 technical report,” *Arxiv*, 2023.

[12] M. , “Temporary policy: chatgpt is banned,” Meta Stack Overflow, 2022. [Online]. Available: https://meta.stackoverflow.com/questions/421831/temporary-policy-chatgpt-is-banned

[13] F. Ortin, W. Wen, et al., “Code2tree: a method for automatically generating code comments,” *Scientific Program.*, vol. 2022, p. 6350686, 2022, doi: 10.1155/2022/6350686. [Online]. Available: https://doi.org/10.1155/2022/6350686

[14] A. Svyatkovskiy, S. K. Deng, S. Fu, and N. Sundaresan, *Intellicode Compose: Code Generation Using Transformer*, 2020, p. 1433, doi: 10.1145/3368089.3417058. [Online]. Available: https://doi.org/10.1145/3368089.3417058

[15] C. Zhang, J. Wang, et al., “A survey of automatic source code summarization,” *Symmetry*, vol. 14, no. 3, 2022, doi: 10.3390/sym14030471. [Online]. Available: https://www.mdpi.com/2073-8994/14/3/471

[16] S. Xu, A. Bennett, D. Hoogeveen, J. H. Lau, and T. Baldwin, *Preferred Answer Selection in Stack Overflow: Better Text Representations ... And Metadata, Metadata, Metadata*, 2018, p. 137, doi: 10.18653/v1/W18-6119. [Online]. Available: https://aclanthology.org/W18-6119

[17] L. Cai, H. Wang, et al., *Answerbot: An Answer Summary Generation Tool Based on Stack Overflow*, 2019, p. 1134, doi: 10.1145/3338906.3341186. [Online]. Available: https://doi.org/10.1145/3338906.3341186

[18] C. Zhang, Q. Zhou, et al., “Re<sub>t</sub>rans: combined retrieval and transformer model for source code summarization,” *Entropy*, vol. 24, no. 10, 2022, doi: 10.3390/e24101372. [Online]. Available: https://www.mdpi.com/1099-4300/24/10/1372

[19] do, and Marcelo Maia, *Towards a Question Answering Assistant for Software Development Using a Transformer-Based Language Model*, 2021, pp. 39–42, doi: 10.1109/BotSE52550.2021.00016.

[20] S. Overflow, “Stack overflow developer survey 2022,” Stack Overflow, 2022. [Online]. Available: https://survey.stackoverflow.co/2022/

[21] D. Correa, and A. Sureka, *Chaff From the Wheat: Characterization and Modeling of Deleted Questions on Stack Overflow*, 2014, p. 631, doi: 10.1145/2566486.2568036. [Online]. Available: https://doi.org/10.1145/2566486.2568036