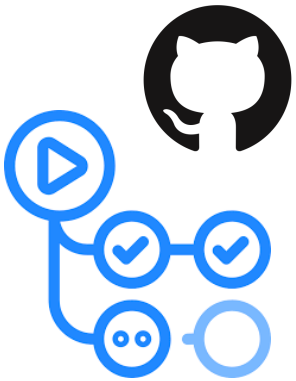
Explaining GitHub Actions Failures with Large Language Models

Challenges, Insights and Limitations

Pablo Valenzuela-Toledo, Chuyue Wu, Sandro Hernandez, Alexander Boll, Roman Machacek, Sebastiano Panichella, Timo Kehrer

Date Published: April 2025

Presented by: Amaan Ahmed



Problem Being Solved

- **GitHub Actions = Continuous Integration / Continuous Deployment Automations.**
  - Runs workflows whenever code is pushed, to test, build, and deploy software.
- **Frequent failures.**
  - Workflows often fail, causes can be misconfigurations, hidden dependencies or environment issues.
- **Debugging is difficult.**
  - Error logs are long, unstructured and not user-friendly.
- **Time-consuming and frustrating.**
  - Developers must scroll through and make sense of hundreds of lines of output to figure out what's wrong.
- **Question: Can LLMs help *explain* these failures in the form of natural summaries, so debugging is faster and more efficient?**

New Idea

Metrics and Research Questions

- **To what extent do LLMs correctly describe the context of GA run failures according to developers?**
  - Are the explanations technically sound?
- **To what extent do developers find LLM generated explanations for GA run failures clear and concise?**
  - How understandable are the explanations? Do the explanations contain only essential information?
- **To what extent are the descriptions of GA run failures considered actionable by developers?**
  - Do the explanations provide specific and relevant information which developers can implement into a solution easily?

TABLE I
ATTRIBUTES USED FOR EVALUATING LLM-GENERATED EXPLANATIONS OF GITHUB ACTIONS RUN FAILURES [17], [18].

Attribute	Definition
Correctness	Measures the accuracy and reliability of the LLM-generated explanations in describing the actual behavior of the system, ensuring information is free from misleading content and inspires confidence in the diagnosis provided.
Conciseness	Reflects whether the explanation is efficient and avoids unnecessary information, presenting only essential details to understand and resolve the issue effectively.
Clarity	Assesses whether the explanation is presented in a clear and understandable manner, enabling developers to readily grasp the issue and the suggested steps.
Actionability	Assesses whether the explanation provides clear, step-by-step guidance that is directly implementable, enabling developers to efficiently address and resolve the failure without needing further clarification or external resources.

New Idea

Survey Study

- **Developers shown GA run failure logs with LLM-generated explanations.**
- **Developers asked to evaluate explanations through close-ended questions and open-ended questions.**
  - Close-ended questions to answer RQ1 and RQ2, through a Likert Scale Rating (Strongly Agree - Neutral - Strongly Disagree)
  - Open-ended questions to answer RQ3, through free-text
- **Authors invited 811 developers, 31 responded back.**
  - Responses for Questions 11 and 12 were then manually categorised by the authors with respect to themes/*sub-attributes*.

RQ1 RQ2 RQ3

Survey Statements & Questions

(1) The explanation accurately reflects the details and context of the GitHub Actions run failure.
(2) The run failure explanation is helpful.
(3) There is a low likelihood of a misleading explanation.
(4) The explanation accurately diagnoses the run failure.
(5) The explanation contains no inappropriate or incorrect content.
(6) There is evident sound diagnostic reasoning.
(7) The explanation clearly and understandably communicates the run failure.
(8) The explanation clearly outlines the subsequent steps to take.
(9) The explanation specifically addresses my needs without being too general.
(10) I am confident in the diagnosis provided by the run failure explanation.
⋆(11) What attributes make an error explanation valuable and effective for addressing issues related to GitHub Actions runs?
⋆(12) Do you have any additional comments or suggestions on how we can enhance our run failure explanations?

New Idea

Survey Tool

- **Custom-built platform called LogExp, to conduct the survey efficiently and uniformally.**
- **Displayed the log and static LLM explanations side by side.**
- **Reduced bias and ensured structured evaluation.**

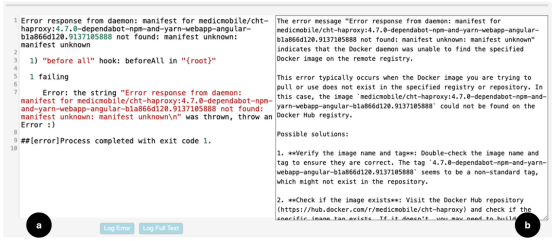


Fig. 2. Partial view of the LogExp tool's interface. The log is displayed on the left, allowing participants to choose between viewing a summary or the full log. On the right, the corresponding textual explanation generated by the LLM is presented. Below these sections, participants encountered statements and questions specific to each case.

Results

Research Question 1 and 2

- **Correctness ~ 80%+ agreement**
  - Most developers found explanations accurate, logically coherent and precise.
- **Clarity and Conciseness ~ 75% - 80% agreement**
  - Over 80% of the participants found the explanations easy to understand.
  - ~75% of the participants found the explanations specific, and not overly broad with unnecessary details.

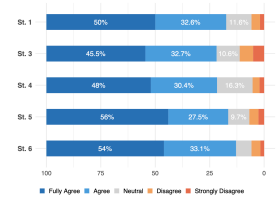


Fig. 3. The stacked bar chart shows the levels of agreement of participants to our statements (1), (3), (4), (5), and (6).

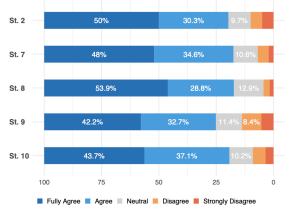


Fig. 4. The stacked bar chart shows the levels of agreement of participants to our statements (2), (7), (8), (9), and (10).

Results

Research Question 3

- To address actionability, **the authors defined 5 categories/sub-attributes.**
- **Clarity of Explanation:**
  - Was the explanation written clearly enough to be followed?
- **Actionable Guidance**
  - Did the explanation suggest a fix or concrete step on what the developer should do next?
- **Specificity of Content**
  - Was the explanation targeted to the actual error, or just a general comment?
- **Contextual Relevance**
  - Did the explanation provide additional context or external links to resources that may help understand the problem more fully?
- **Conciseness**
  - Was the explanation brief yet informative? Were the solution steps concisely presented?

Answer to RQ3

Effective explanations for GitHub Actions run failures include five key attributes: *clarity*, which provides straightforward information; *actionable guidance*, offering precise steps for resolution; *specificity*, adapting explanations to the technical context; *contextual relevance*, adding links or details about dependencies; and *conciseness*, ensuring only essential information is presented.

> *I believe that a useful error explanation should get straight to the point without saying a lot of unnecessary things. Furthermore the explanation should be easily understandable by people that are just getting started so they can become better at understanding errors. Last but not least the steps to fix the issue shouldn't be too general because then a google search is better. [ID:5]"*

> *1. Cutting fluff, going straight to the point. 2. Possible steps to take to fix the problem. [ID:2]"*

Positives

- **Great potential for LLMs to reduce human effort.**
  - Not just error logs, but this research can inspire efforts to summarise other large volume unstructured data.
- **Systematic approach, attributes tie in with research questions seamlessly.**
  - Clear structure to the paper, methodology was easy to follow.
- **Consistent evaluation with balanced design and reduced bias.**
  - Custom-built LogExp tool and data collected through both close-ended statements and open-ended qs.
- **Separate section for defending validity.**
  - Authors go into depth about their design choices and provide reasonings in defence.
- **Replication package and raw data provision.**
  - The work can be replicated and verified.

Negatives

- **Very low response rate and indeterminant sample size.**
  - 31 out of 811 developers responded. Authors chose responses that were at least 70% completed.
- **Logs presented were simplified, not real unstructured log swamps.**
  - The examples included in the paper are of very simple logs, not logs awash in a text swamp.
- **Unclear if LLMs ran on full logs or separated excerpts.**
  - The tool only present an explanation of a particular log, it is not clear whether it was provided a single log or the entire unstructured log.
- **Participant selection bias.**
  - Only skilled developers with prior experience in handling GA run failures were selected.

Future Work

- **Increase sample size and participant diversity.**
  - Include less experienced developers and learn from their feedback.
- **Test on realistic, unstructured log 'swamps'.**
  - Token limit could be an issue here.
- **Explore fine-tuned LLMs trained on CI/CD data.**
  - The paper identifies poor performance on CI/CD data with the general LLMs used.
- **Extend beyond GitHub Actions to other CI/CD tools.**
  - Jenkins, Azure DevOps, Azure Pipelines.

Rating: 4/5

Great application of LLMs to make human life easier.

Discussion Points

- **Can LLMs be trusted for this task if they sometimes give confident but wrong or misleading explanations?**
  - How do we mediate this? What sort of manual intervention can help but maintain reduced effort?
- **Is conciseness more important than actionability?**
  - What trade-offs can we feasibly undertake here?
- **How do we balance the risk of developer over-reliance on LLMs versus their productivity benefits?**
  - Developers already employ LLMs to generate code, will this affect their debugging skills as well? What skills would developers require then?

Thank you.