ChatGPT Incorrectness Detection in Software Reviews

Authors

Minaoar Hossain Tanzil
Junaed Younus Khan
Gias Uddin

Presented by

Nafis Nahian (2105007)
Nafees Ashraf (2105008)
Abrar Zahin Raihan (2105009)

Bangladesh University of Engineering and Technology

Department of Computer Science and Engineering

December 1, 2024

Presentation Outline

1. Introduction
2. Survey of Software Developers
3. CID: ChatGPT Incorrectness Detector
4. Evaluation of CID
5. Conclusion

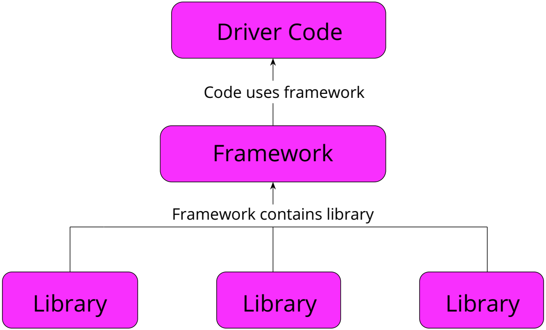
Outline

Background

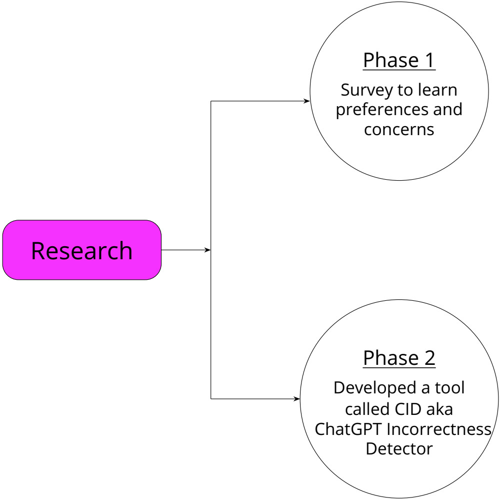
Generative AI tools like ChatGPT are revolutionizing various domains, including software engineering. But can we trust their responses?

Background

Developers are increasingly using ChatGPT for SE tasks like library selection



Overview


Research
Phase 1
Survey to learn preferences and concerns
Phase 2
Developed a tool called CID aka ChatGPT Incorrectness Detector

Overview

- Focus on software library selection as a case study.
- Need for understanding how developers use ChatGPT and their concerns.
- Desire for automated tools to detect incorrectness in ChatGPT's outputs.

Overview

- Focus on software library selection as a case study.
- Need for understanding how developers use ChatGPT and their concerns.
- Desire for automated tools to detect incorrectness in ChatGPT's outputs.

Overview

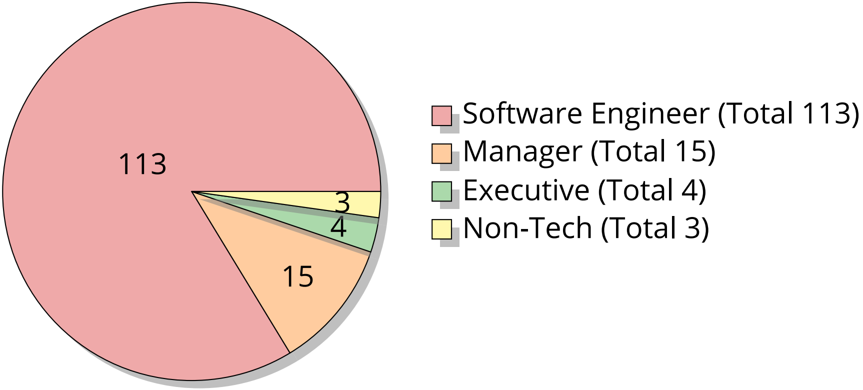
- Focus on software library selection as a case study.
- Need for understanding how developers use ChatGPT and their concerns.
- Desire for automated tools to detect incorrectness in ChatGPT's outputs.

Outline

Survey Overview

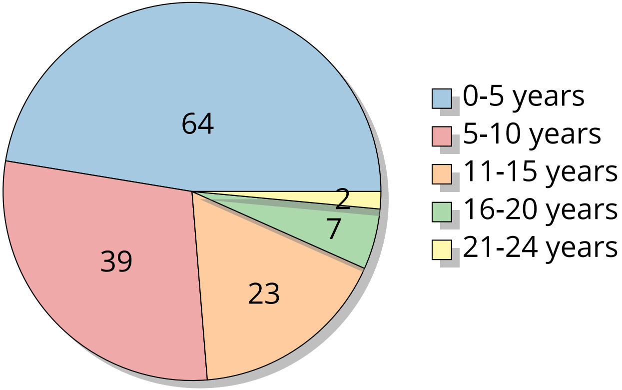
- Conducted a survey with 135 SE practitioners.
- Aimed to answer three Research Questions (RQs):
  - **RQ1:** Why do software developers use ChatGPT?
  - **RQ2:** How much do developers rely on ChatGPT responses?
  - **RQ3:** How do developers verify ChatGPT responses?

Participant Demographics (Current Profession)



Total Participants = 135

Participant Demographics (Years of Experience)



Total Participants = 135

Survey Questions

Table: Survey questions and their mapping to the Research Questions. Here, C/O=Close/Open-ended question, G/S=Generic/Scenario-based question. For scenario-based questions, we used library selection as a case-study.

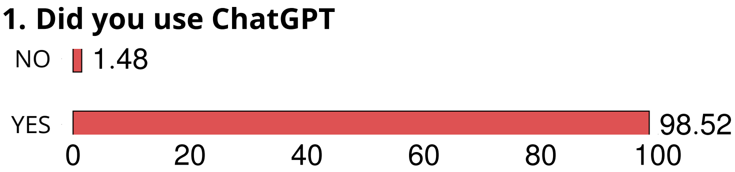
Q#	Questions	O/C	G/S	RQ
1	Did you use ChatGPT?	C	G	1.1
2	In general, which of the cases you used it for?	C	G	1.1
3	As a software professional, how did you or can you use it?	C	G	1.1
4	How would you describe your experience with using it so far?	C	G	1.1
5	How much do you rely on the content/response of ChatGPT?	C	G	1.2
6	Have you considered using ChatGPT to select or compare software libraries? Please share the pros and cons.	O	S	1.2
7	How much would you rely on ChatGPT's response for the given library selection query?	C	S	1.3
8	Would you rely on the ChatGPT's response after further inquiry?	C	S	1.3
9	Do you think the opinion from ChatGPT is correct?	C	S	1.3
10	What can be the ways to improve the reliability of ChatGPT responses?	C	G	1.3

Reasons to use ChatGPT (RQ1)

The answers to the following questions help us to find the answer to Why developers used ChatGPT

1. Did you use ChatGPT?
2. In general, which of the cases you used it for?
3. As a software professional, how did you or can you use it?
4. How would you describe your experience with using it so far?

Reasons to use ChatGPT (RQ1)

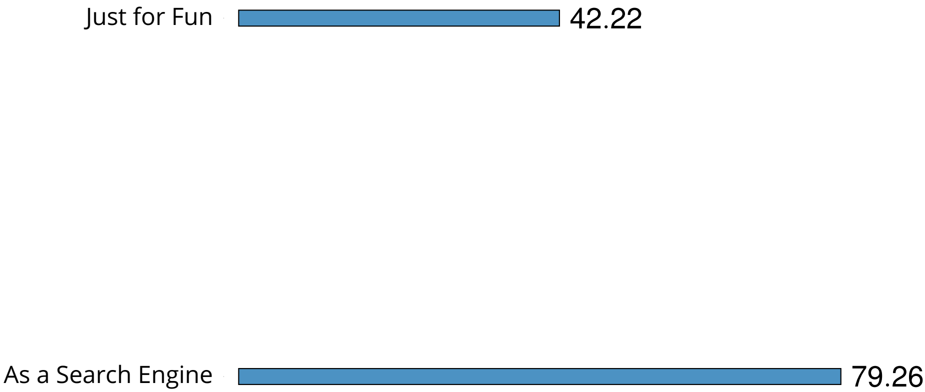


Reasons to use ChatGPT (RQ1)

2. In general, which of the cases you used it for?

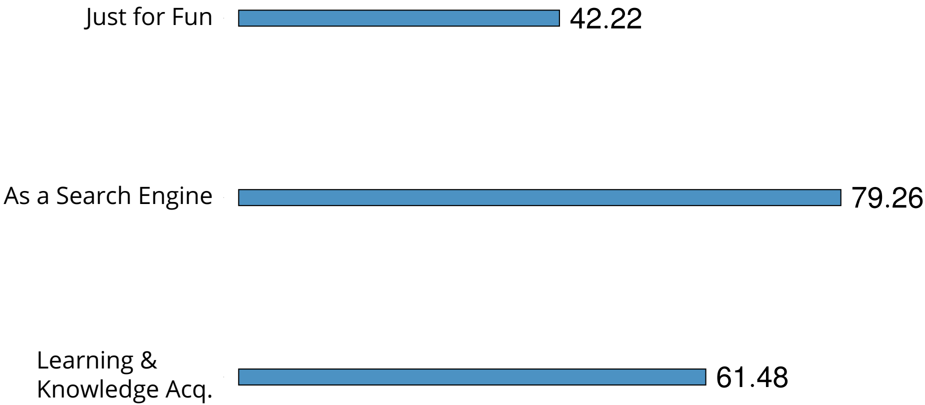
Reasons to use ChatGPT (RQ1)

2. In general, which of the cases you used it for?



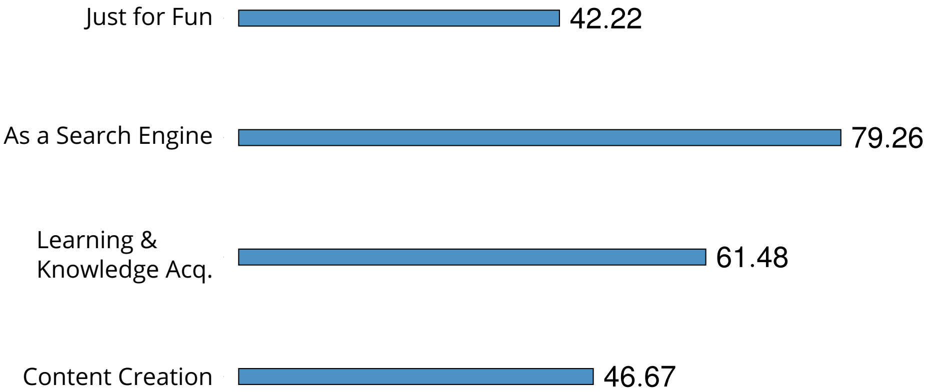
Reasons to use ChatGPT (RQ1)

2. In general, which of the cases you used it for?

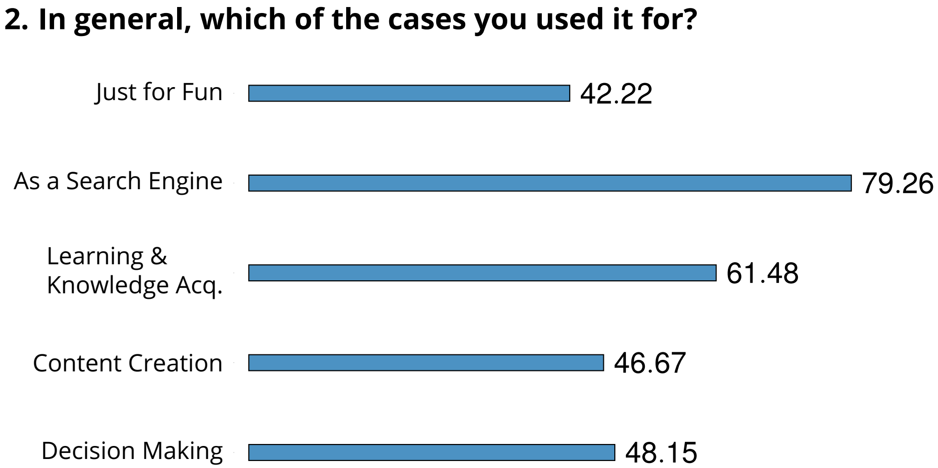


Reasons to use ChatGPT (RQ1)

2. In general, which of the cases you used it for?

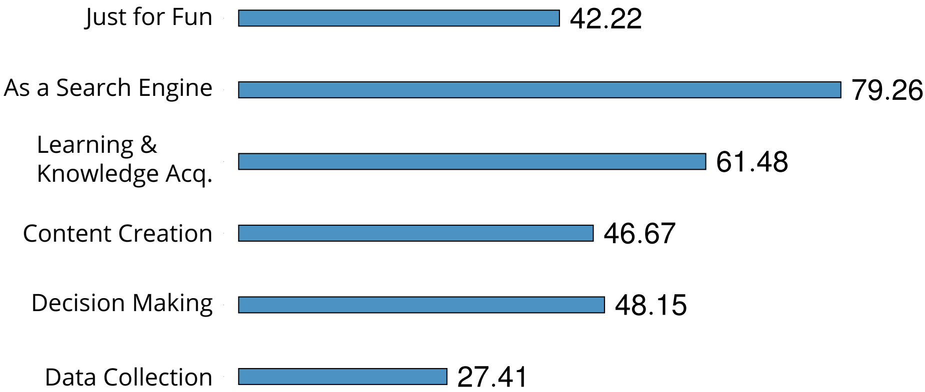


Reasons to use ChatGPT (RQ1)



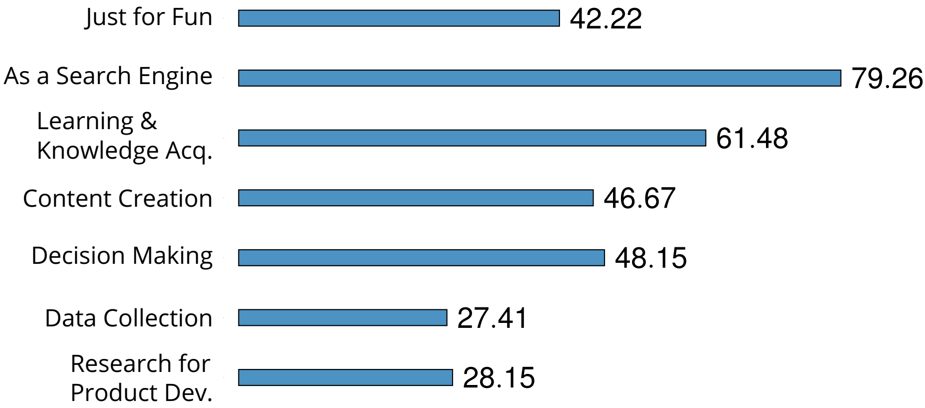
Reasons to use ChatGPT (RQ1)

2. In general, which of the cases you used it for?



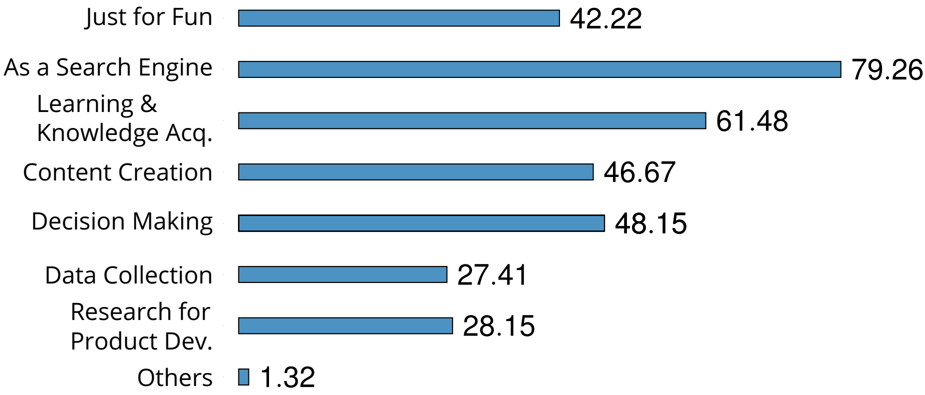
Reasons to use ChatGPT (RQ1)

2. In general, which of the cases you used it for?



Reasons to use ChatGPT (RQ1)

2. In general, which of the cases you used it for?

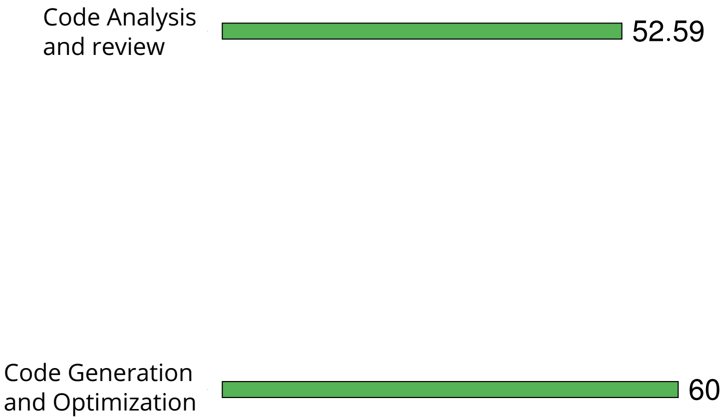


Reasons to use ChatGPT (RQ1)

3. As a software professional, how did you or can you use it?

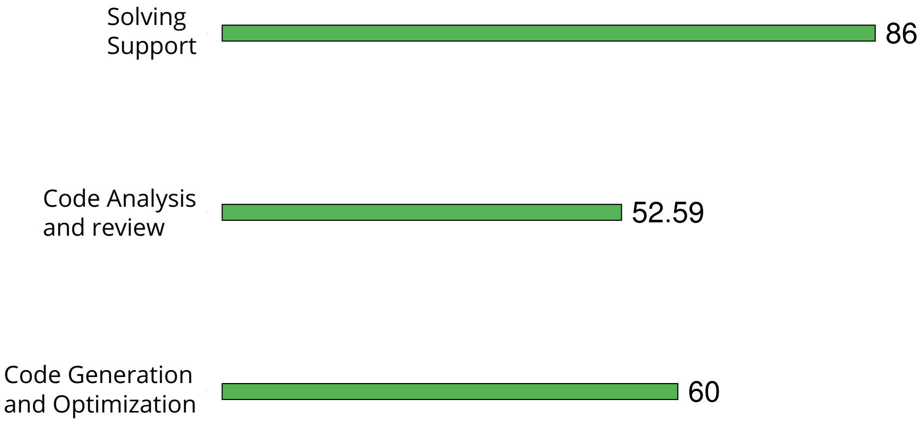
Reasons to use ChatGPT (RQ1)

3. As a software professional, how did you or can you use it?



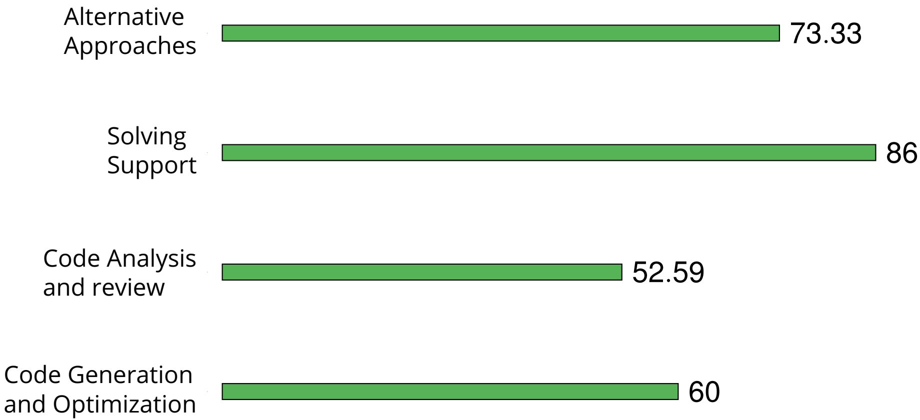
Reasons to use ChatGPT (RQ1)

3. As a software professional, how did you or can you use it?



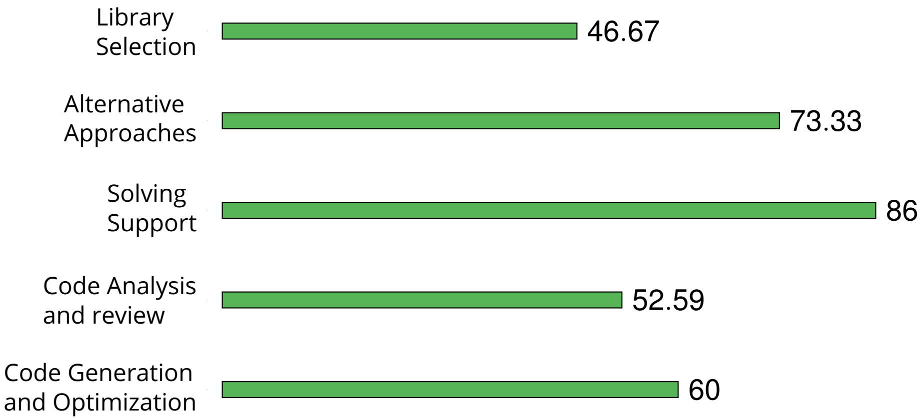
Reasons to use ChatGPT (RQ1)

3. As a software professional, how did you or can you use it?



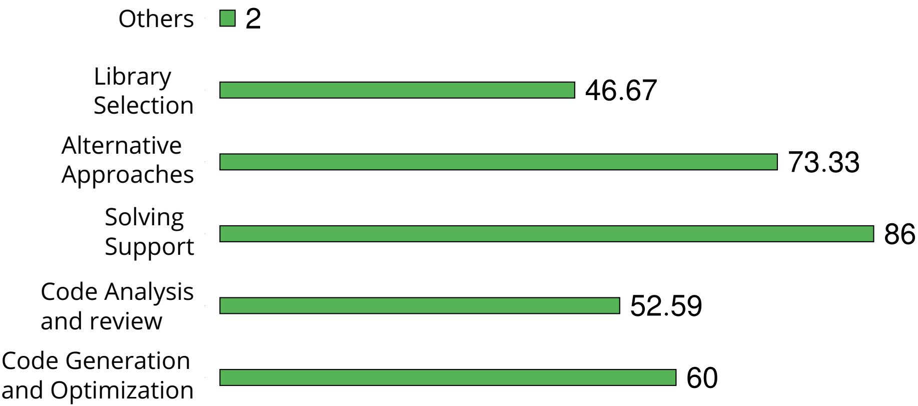
Reasons to use ChatGPT (RQ1)

3. As a software professional, how did you or can you use it?



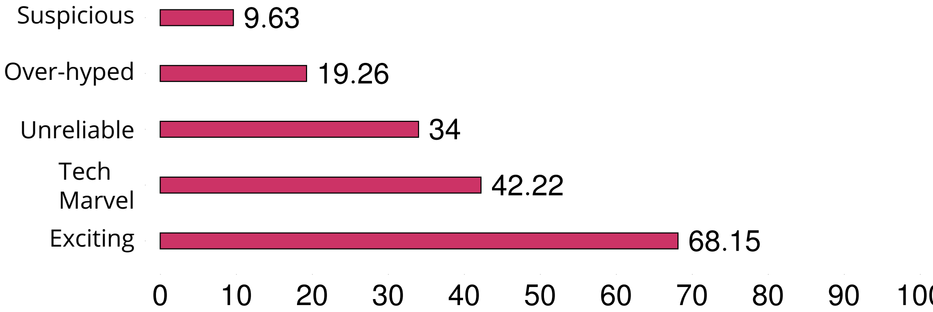
Reasons to use ChatGPT (RQ1)

3. As a software professional, how did you or can you use it?



Reasons to use ChatGPT (RQ1)

4. How would you describe your experience with using it so far?



Summarizing Key Findings for RQ1

- **Usage Purposes:**
  - Code generation and optimization.
  - Problem-solving support.
  - Exploring alternative approaches.
  - Library selection.
- **Experience:**
  - Excitement and recognition of technological advancement.
  - Concerns about reliability and overhyped expectations.

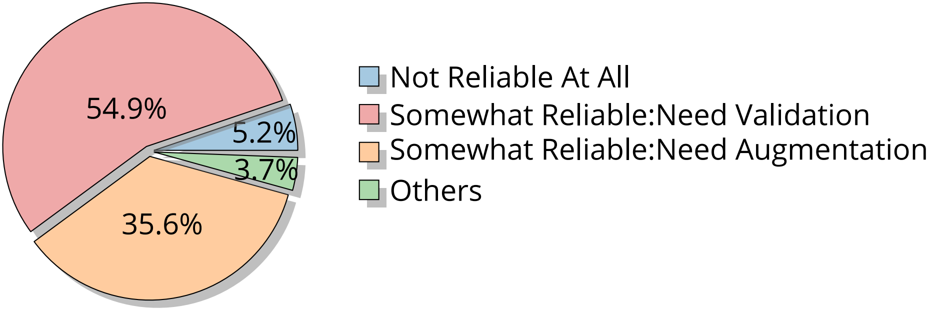
Concerns About ChatGPT Responses (RQ2)

We asked the following questions to the participants to find out how reliable ChatGPT is

1. How much do you rely on the content/response of ChatGPT?
2. Have you considered using ChatGPT to select or compare software libraries? Please share the pros and cons?

Concerns About ChatGPT Responses (RQ2)

1. How much do you rely on the content/response of ChatGPT?



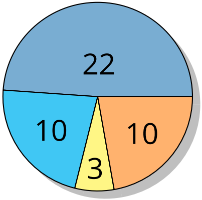
Concerns About ChatGPT Responses (RQ2)

2. Have you considered using ChatGPT to select or compare software libraries? Please share the pros and cons?

- PROS :
  1. Efficient Access to Information
  2. Initial Idea Generation
  3. Personalized Recommendations
  4. Time-Saving
- CONS :
  1. Lack of Up-to-dateness
  2. Contextual Understanding Challenges
  3. Reliability Concerns
  4. Dependence on Prompt
  5. Not Sufficient for Decision-Making
  6. Bias of Training Data

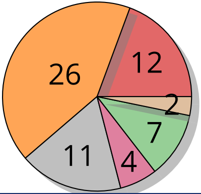
Pros and Cons for Library Selection (RQ2)

Pros : Total 45 responses



- Efficient Access to Info
- Initial Idea Generation
- Personalized Recommendations
- Time Saving

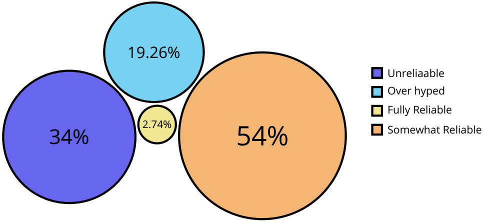
Cons : Total 60 Responses



- Lack of Up-to-date Knowledge
- Reliability Concerns
- Contextual Understanding Challenges
- Dependence on Prompt
- Not Sufficient for Decision Making
- Biased on Training Data

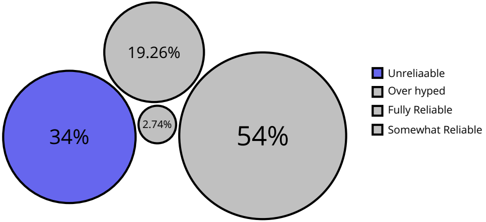
Key Findings for RQ2

- **Reliability Concerns:**
  - Only a small percentage fully trust ChatGPT responses.
  - Majority consider the responses somewhat reliable but require validation.



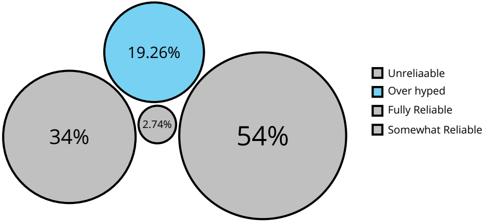
Key Findings for RQ2

- **Reliability Concerns:**
  - Only a small percentage fully trust ChatGPT responses.
  - Majority consider the responses somewhat reliable but require validation.



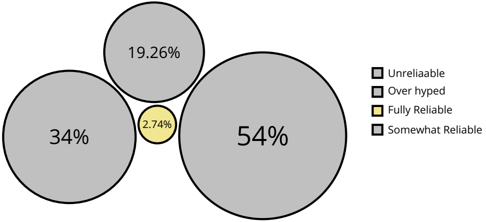
Key Findings for RQ2

- **Reliability Concerns:**
  - Only a small percentage fully trust ChatGPT responses.
  - Majority consider the responses somewhat reliable but require validation.



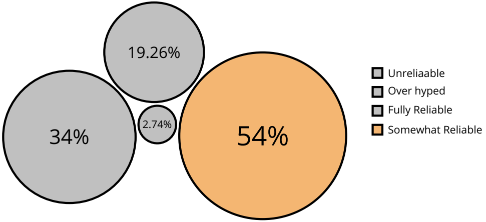
Key Findings for RQ2

- **Reliability Concerns:**
  - Only a small percentage fully trust ChatGPT responses.
  - Majority consider the responses somewhat reliable but require validation.



Key Findings for RQ2

- **Reliability Concerns:**
  - Only a small percentage fully trust ChatGPT responses.
  - Majority consider the responses somewhat reliable but require validation.

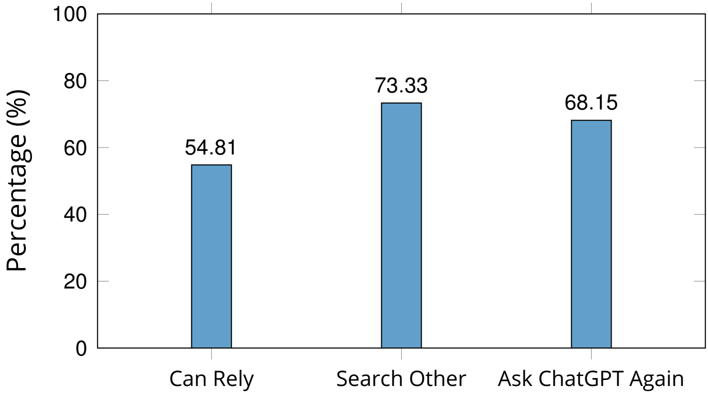


Verification of ChatGPT Responses (RQ3)

Participants were presented with conversations with ChatGPT where it was asked

- Suggestions
- More detail about a specific situation
- Reliability in SE real-world situations

Key Findings



Outline

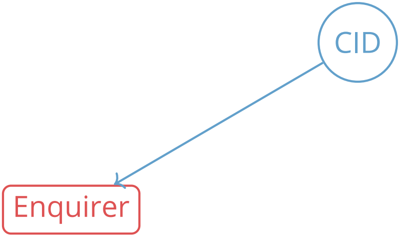
1. Introduction
2. Survey of Software Developers
3. CID: ChatGPT Incorrectness Detector
4. Evaluation of CID
5. Conclusion

Introducing CID

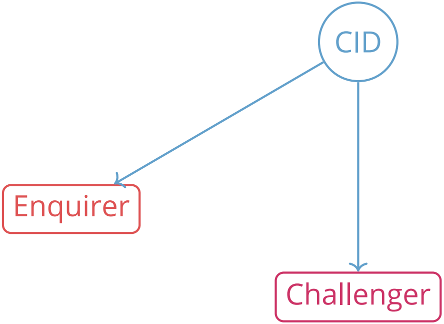
- CID (ChatGPT Incorrectness Detector) tool uses iterative prompting to capture ChatGPT's inconsistency in a similar fashion to an actual Crime Investigation Department (CID).

CID Tool Components

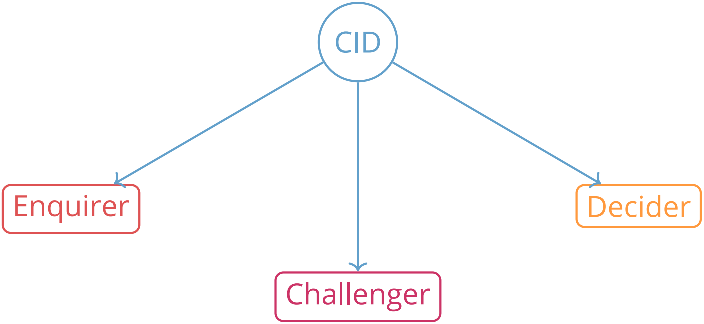
CID Tool Components



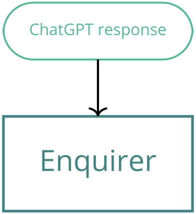
CID Tool Components

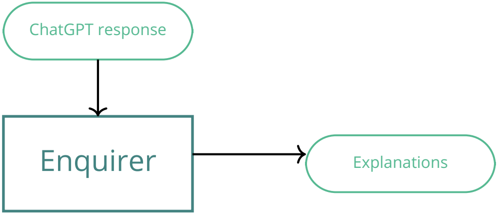


CID Tool Components



Enquirer


ChatGPT response
Enquirer


ChatGPT response
Enquirer
Explanations

ENQUIRER Component

- The ENQUIRER targets to obtain ChatGPT's initial reasoning behind the base-response that can be useful to reveal any inconsistency in the next steps of interrogation.
- Asks ChatGPT to provide separate reasoning for each piece of information by using the following prompt.

> **Enquiring ChatGPT**
>
> Justify your answer. If the answer has multiple pieces of information, provide separate reasoning for each of them.

ENQUIRER Component

- The ENQUIRER targets to obtain ChatGPT's initial reasoning behind the base-response that can be useful to reveal any inconsistency in the next steps of interrogation.
- Asks ChatGPT to provide separate reasoning for each piece of information by using the following prompt.

Enquiring ChatGPT

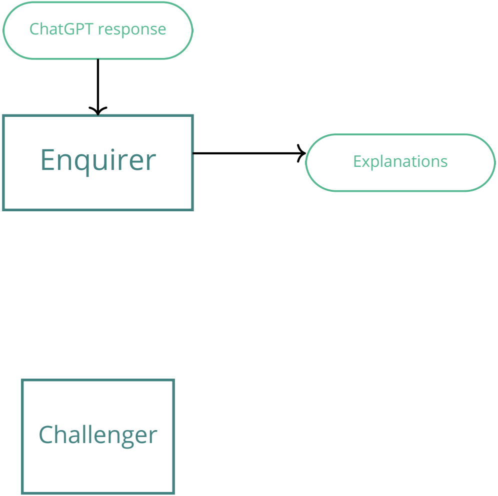
Justify your answer. If the answer has multiple pieces of information, provide separate reasoning for each of them.

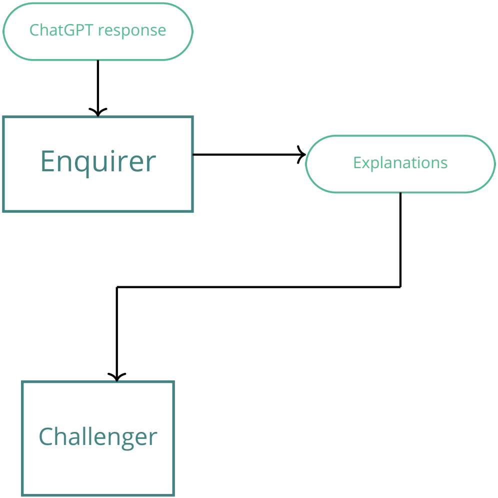
ENQUIRER Component

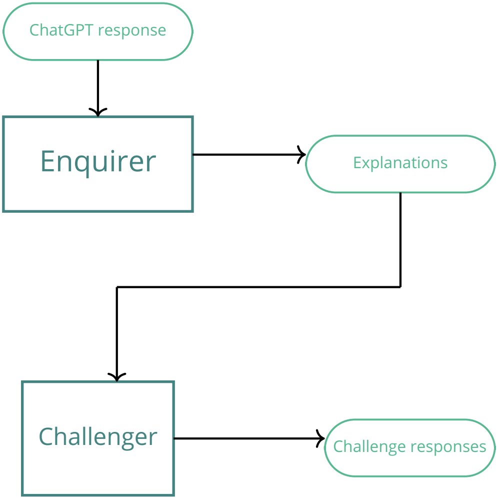
- The ENQUIRER targets to obtain ChatGPT's initial reasoning behind the base-response that can be useful to reveal any inconsistency in the next steps of interrogation.
- Asks ChatGPT to provide separate reasoning for each piece of information by using the following prompt.

Enquiring ChatGPT

Justify your answer. If the answer has multiple pieces of information, provide separate reasoning for each of them.

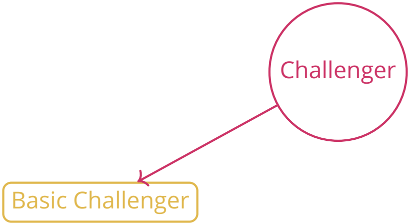

ChatGPT response
Enquirer
Explanations
Challenger


ChatGPT response
Enquirer
Explanations
Challenger

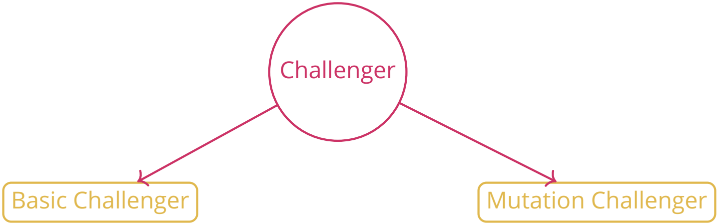

ChatGPT response
Enquirer
Explanations
Challenger
Challenge responses

Challenger Components

Challenger Components



Challenger Components



Basic Challenger

- We first ask three basic challenge questions to ChatGPT: *Why?, How?, Really?* for each explanation (E_i) of its base-response (R_B).
- The basic challenger leverages a separate LLM.
- To replicate a separate LLM, we used ChatGPT with a new separate session. The motive for using a separate session of ChatGPT is to discard the memory of the previous conversation performed.

Basic Challenger

- We first ask three basic challenge questions to ChatGPT: *Why?, How?, Really?* for each explanation (E_i) of its base-response (R_B).
- The basic challenger leverages a separate LLM.
- To replicate a separate LLM, we used ChatGPT with a new separate session. The motive for using a separate session of ChatGPT is to discard the memory of the previous conversation performed.

Basic Challenger

- We first ask three basic challenge questions to ChatGPT: *Why?, How?, Really?* for each explanation (E_i) of its base-response (R_B).
- The basic challenger leverages a separate LLM.
- To replicate a separate LLM, we used ChatGPT with a new separate session. The motive for using a separate session of ChatGPT is to discard the memory of the previous conversation performed.

Mutation Challenger

- Aims to increase the cognitive load of the model.
- It mutates the basic challenge questions to create mutation challenge questions.
- Employs the *Sentence-level metamorphic testing technique, QAQA*
- It inserts a redundant sentence as a clause to the original (basic challenge) question to generate the mutated question and challenges it.
- Depending on the source of the redundant sentence, the mutation challenger applies two types of metamorphic relation (MR): **Equivalent Question (MR1)** and **Equivalent Test Integration (MR2)**

Mutation Challenger

- Aims to increase the cognitive load of the model.
- It mutates the basic challenge questions to create mutation challenge questions.
- Employs the *Sentence-level metamorphic testing technique, QAQA*
- It inserts a redundant sentence as a clause to the original (basic challenge) question to generate the mutated question and challenges it.
- Depending on the source of the redundant sentence, the mutation challenger applies two types of metamorphic relation (MR): **Equivalent Question (MR1)** and **Equivalent Test Integration (MR2)**

Mutation Challenger

- Aims to increase the cognitive load of the model.
- It mutates the basic challenge questions to create mutation challenge questions.
- Employs the *Sentence-level metamorphic testing technique, QAQA*
- It inserts a redundant sentence as a clause to the original (basic challenge) question to generate the mutated question and challenges it.
- Depending on the source of the redundant sentence, the mutation challenger applies two types of metamorphic relation (MR): **Equivalent Question (MR1)** and **Equivalent Test Integration (MR2)**

Mutation Challenger

- Aims to increase the cognitive load of the model.
- It mutates the basic challenge questions to create mutation challenge questions.
- Employs the *Sentence-level metamorphic testing technique, QAQA*
- It inserts a redundant sentence as a clause to the original (basic challenge) question to generate the mutated question and challenges it.
- Depending on the source of the redundant sentence, the mutation challenger applies two types of metamorphic relation (MR): **Equivalent Question (MR1)** and **Equivalent Test Integration (MR2)**

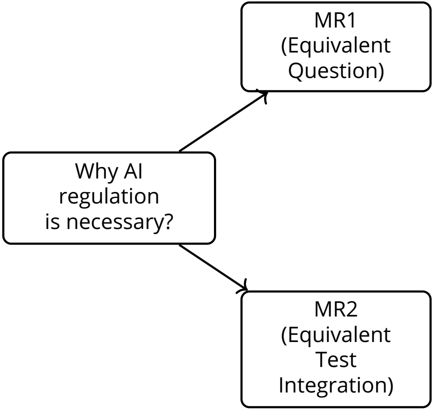
Mutation Challenger

- Aims to increase the cognitive load of the model.
- It mutates the basic challenge questions to create mutation challenge questions.
- Employs the *Sentence-level metamorphic testing technique, QAQA*
- It inserts a redundant sentence as a clause to the original (basic challenge) question to generate the mutated question and challenges it.
- Depending on the source of the redundant sentence, the mutation challenger applies two types of metamorphic relation (MR): **Equivalent Question (MR1)** and **Equivalent Test Integration (MR2)**

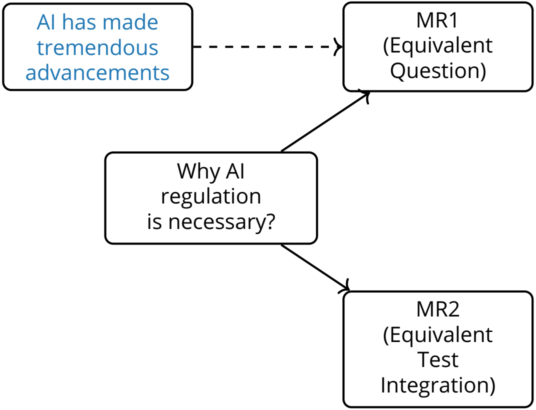
Mutated Questions Flow Example

Why AI
regulation
is necessary?

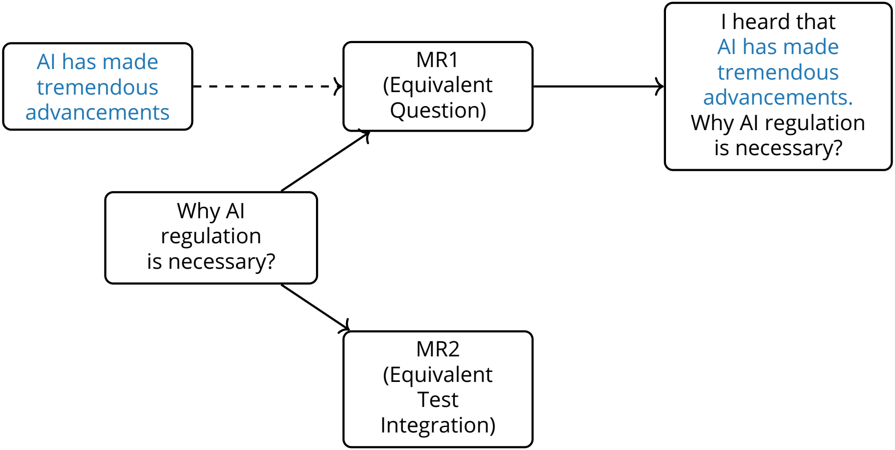
Mutated Questions Flow Example



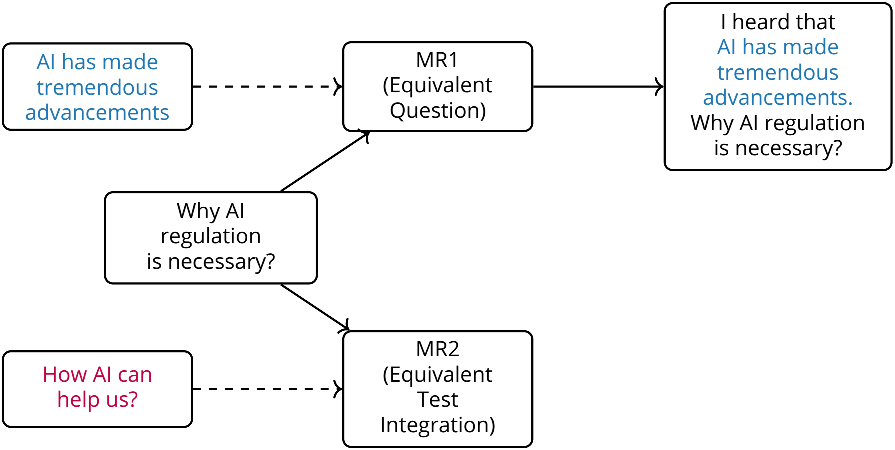
Mutated Questions Flow Example



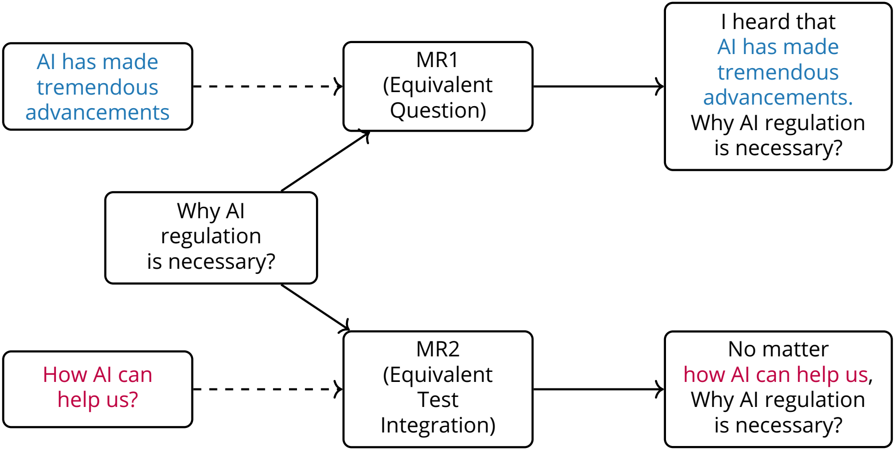
Mutated Questions Flow Example

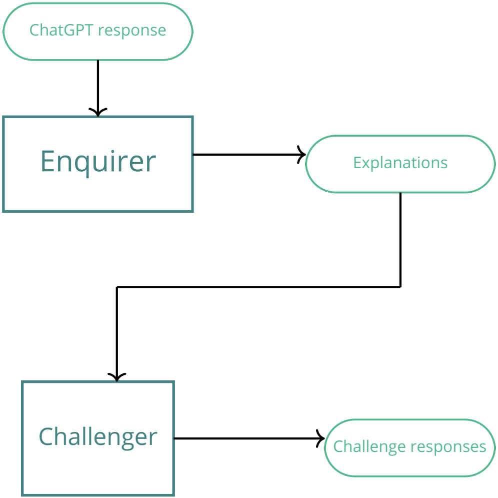


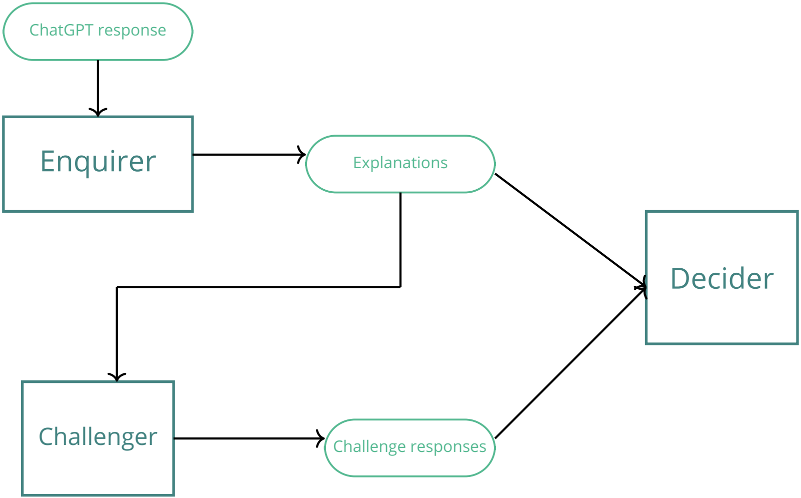
Mutated Questions Flow Example

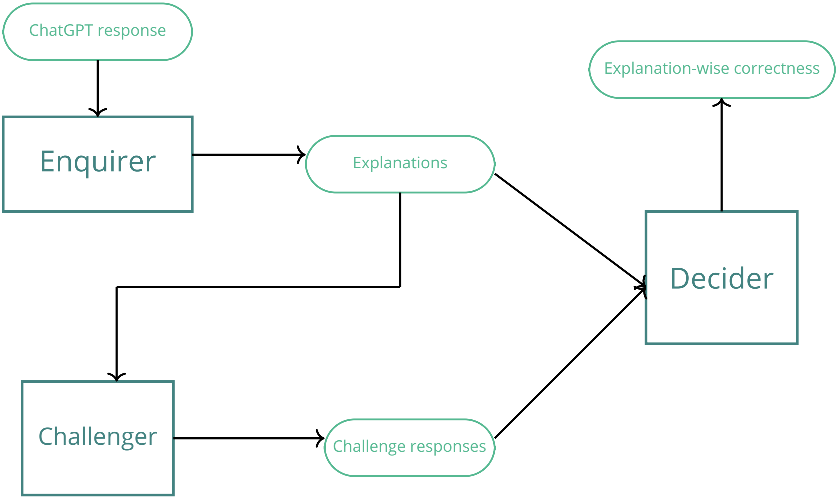


Mutated Questions Flow Example



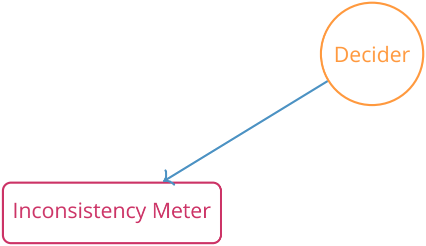

ChatGPT response
Enquirer
Explanations
Challenger
Challenge responses


ChatGPT response
Enquirer
Explanations
Decider
Challenger
Challenge responses

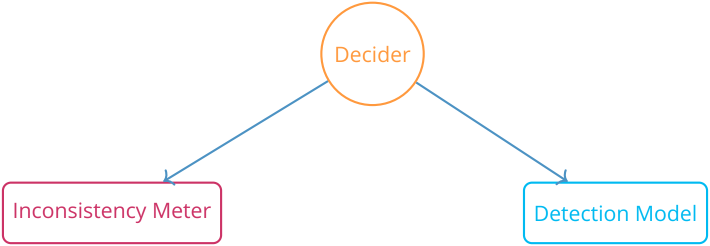

ChatGPT response
Enquirer
Explanations
Explanation-wise correctness
Decider
Challenger
Challenge responses

Decider Modules

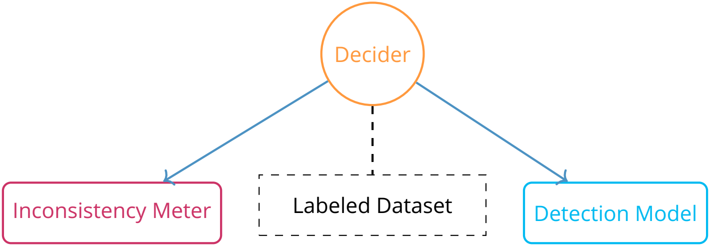
Decider Modules



Decider Modules



Decider Modules



Dataset Creation

- The dataset is generated by interacting with ChatGPT and posing various questions to it.
- we use our ENQUIRER to split each base response into multiple explanations. Finally, these explanations are manually labeled as correct/incorrect by human annotators.

Dataset Creation

- The dataset is generated by interacting with ChatGPT and posing various questions to it.
- we use our ENQUIRER to split each base response into multiple explanations. Finally, these explanations are manually labeled as correct/incorrect by human annotators.

Inconsistency Meter and Detection Model

- Standard similarity scores is computed among ChatGPT responses generated in the ENQUIRY and CHALLENGE phases.
- These scores are used as features for our tool.
- ML model is trained so that they learn the relationship between ChatGPT's incorrectness and inconsistency.
- 24 features from four categories is used to train the model.
    - Explanation-Response ($E_i - R_C$) Similarity
    - Response-Response ($R_C - R_C$) Similarity
    - Question-Response ($Q_C - R_C$) Similarity
    - Question-Question ($Q_C - Q_C$) Similarity

Inconsistency Meter and Detection Model

- Standard similarity scores is computed among ChatGPT responses generated in the ENQUIRY and CHALLENGE phases.
- These scores are used as features for our tool.
- ML model is trained so that they learn the relationship between ChatGPT's incorrectness and inconsistency.
- 24 features from four categories is used to train the model.
  - Explanation-Response ($E_i - R_C$) Similarity
  - Response-Response ($R_C - R_C$) Similarity
  - Question-Response ($Q_C - R_C$) Similarity
  - Question-Question ($Q_C - Q_C$) Similarity

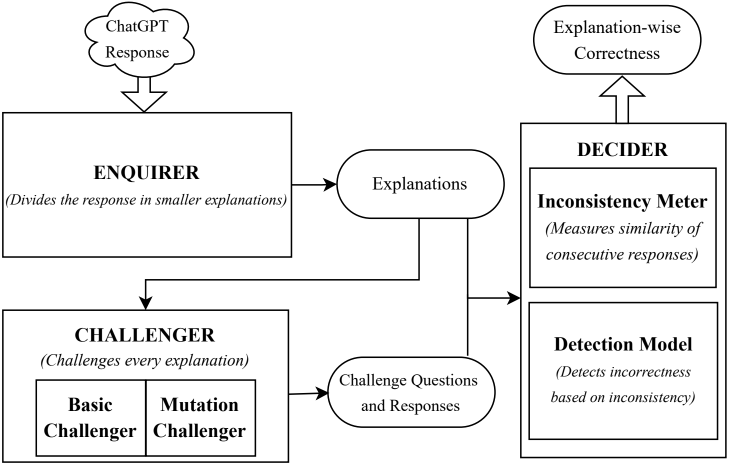
Inconsistency Meter and Detection Model

- Standard similarity scores is computed among ChatGPT responses generated in the ENQUIRY and CHALLENGE phases.
- These scores are used as features for our tool.
- ML model is trained so that they learn the relationship between ChatGPT's incorrectness and inconsistency.
- 24 features from four categories is used to train the model.
  - Explanation-Response ($E_i - R_C$) Similarity
  - Response-Response ($R_C - R_C$) Similarity
  - Question-Response ($Q_C - R_C$) Similarity
  - Question-Question ($Q_C - Q_C$) Similarity

Inconsistency Meter and Detection Model

- Standard similarity scores is computed among ChatGPT responses generated in the ENQUIRY and CHALLENGE phases.
- These scores are used as features for our tool.
- ML model is trained so that they learn the relationship between ChatGPT's incorrectness and inconsistency.
- 24 features from four categories is used to train the model.
  - Explanation-Response ($E_i - R_C$) Similarity
  - Response-Response ($R_C - R_C$) Similarity
  - Question-Response ($Q_C - R_C$) Similarity
  - Question-Question ($Q_C - Q_C$) Similarity

CID Tool Overview



Outline

1. Introduction
2. Survey of Software Developers
3. CID: ChatGPT Incorrectness Detector
4. Evaluation of CID
5. Conclusion

Evaluation of CID

- **Research Questions**
  - **RQ4**: How accurate is CID in detecting incorrect responses?
  - **RQ5**: How do the base and mutation challenge prompts impact performance?

Benchmark Study Setup

- **Context**: Software Library Selection Task
- **Dataset Collection**:
  - Collected 100 Stack Overflow (SO) posts
  - Focused on text processing libraries: **spaCy**, **NLTK**, **GSON**
  - Covered aspects like ease of use, performance, stability, etc.
- **Base Questions**:
  - Formulated questions based on SO posts
  - Example: *"How easy is it to use the library strictly based on the following conversation?"*

Visualizing the Benchmark Setup

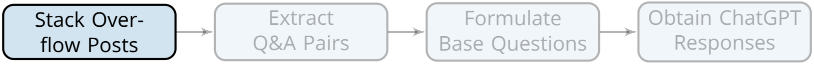


Figure: Flowchart of Benchmark Study Setup

Visualizing the Benchmark Setup

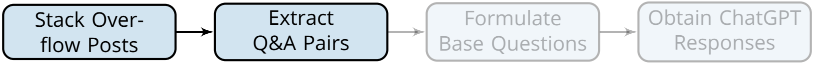


Figure: Flowchart of Benchmark Study Setup

Visualizing the Benchmark Setup

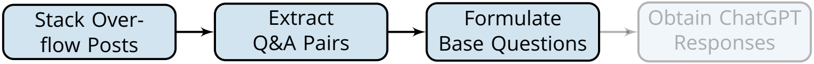


Figure: Flowchart of Benchmark Study Setup

Visualizing the Benchmark Setup

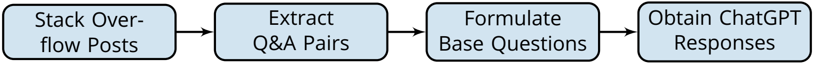


Figure: Flowchart of Benchmark Study Setup

CID Components Recap

ENQUIRER

- Extracts explanations from ChatGPT's base responses

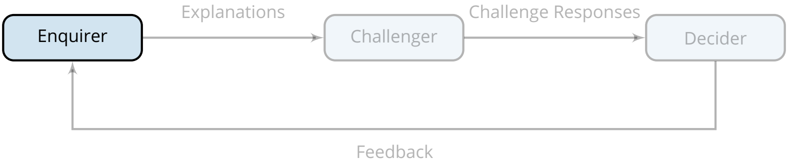


Figure: Interaction between CID Components

CID Components Recap

CHALLENGER

- Poses basic and mutated challenge questions
- Uses metamorphic relationships to mutate questions

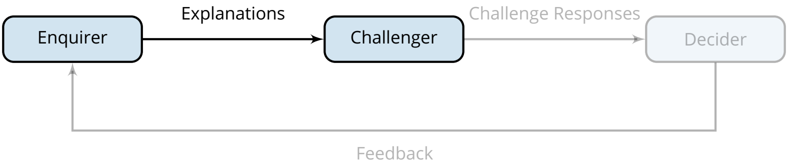


Figure: Interaction between CID Components

CID Components Recap

DECIDER

- Analyzes inconsistencies
- Employs ML techniques to detect incorrectness

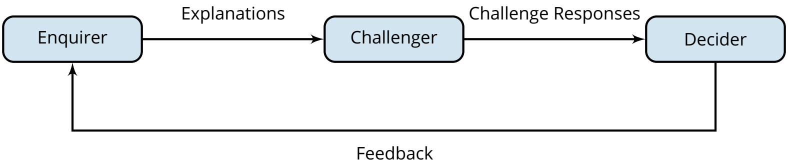


Figure: Interaction between CID Components

Explanation Generation

Process:

- ChatGPT provides base responses to base questions
- ENQUIRER requests separate explanations for each piece of information

Explanation Generation

Outcome:

- Generated 341 explanations from 100 posts
- **Labeling**:
  - **276 explanations (81%) labeled as correct**
  - 65 explanations (19%) labeled as incorrect

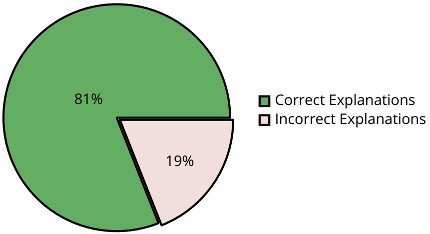


Figure: Distribution of Correct and Incorrect Explanations

Explanation Generation

Outcome:

- Generated 341 explanations from 100 posts
- **Labeling**:
  - 276 explanations (81%) labeled as correct
  - **65 explanations (19%) labeled as incorrect**

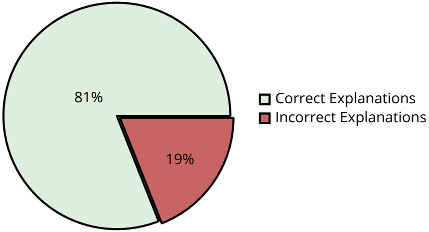


Figure: Distribution of Correct and Incorrect Explanations

Incorrectness Detection Performance (RQ4)

- **Machine Learning Models Evaluated**:
  - Logistic Regression (LR)
  - Random Forest (RF)
  - Support Vector Machine (SVM)
- **Performance Metrics**:
  - Precision (P), Recall (R), F1-Score (F1), Accuracy (A)

Model	P	R	A	F1
Logistic Regression (LR)	0.74	0.65	0.65	0.68
Random Forest (RF)	0.73	0.65	0.65	0.68
Support Vector Machine (SVM)	**0.74**	**0.75**	**0.75**	**0.74**

Table: ML model performance to detect ChatGPT incorrectness

Visualizing Model Performance

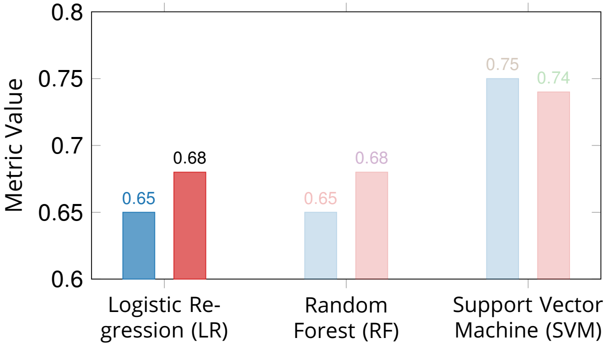


Figure: Comparison of ML Model Performances

Visualizing Model Performance

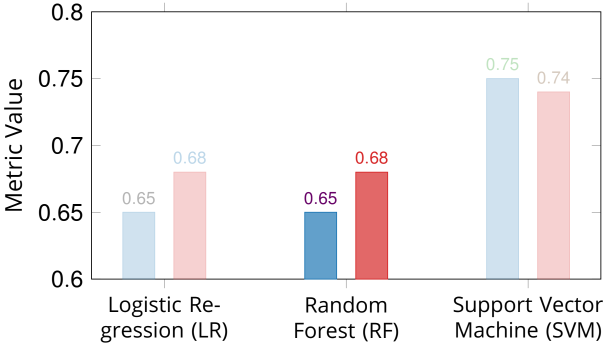


Figure: Comparison of ML Model Performances

Visualizing Model Performance

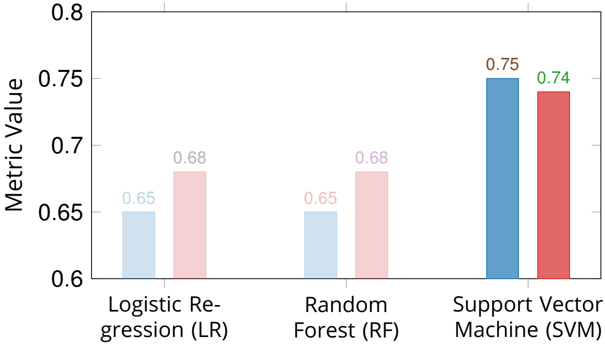


Figure: Comparison of ML Model Performances

Misclassification Analysis

- **Total Misclassifications**: 86 out of 341 explanations

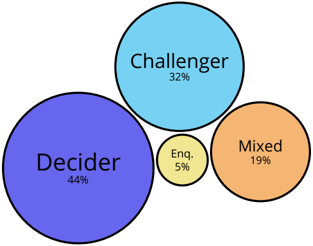


Figure: Distribution of Error Sources

Misclassification Analysis

- **Total Misclassifications**: 86 out of 341 explanations
- **Error Sources** (44% of errors)
  - **Decider Component** (44% of errors)
    - Similarity calculation issues
    - Difficulty detecting unanimous incorrect responses

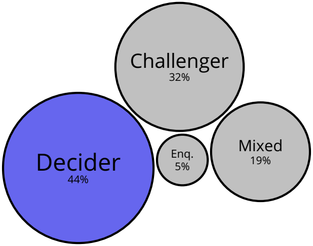


Figure: Distribution of Error Sources

Misclassification Analysis

- **Total Misclassifications**: 86 out of 341 explanations
- **Error Sources** (44% of errors)
  - **Challenger Component** (32% of errors)
    - Misdirected challenges
    - Out-of-scope questions

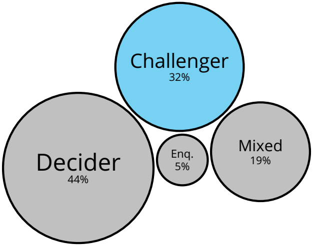


Figure: Distribution of Error Sources

Misclassification Analysis

- **Total Misclassifications**: 86 out of 341 explanations
- **Error Sources** (44% of errors)
  - **Enquirer Component** (5% of errors)
    - Convoluted explanations with multiple opinions

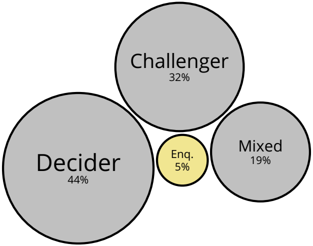


Figure: Distribution of Error Sources

Misclassification Analysis

- **Total Misclassifications**: 86 out of 341 explanations
- **Error Sources** (44% of errors)
  - **Mixed Sources** (19% of errors)
    - Continuous incorrect reasoning by ChatGPT
    - Generic issues (unclear information)

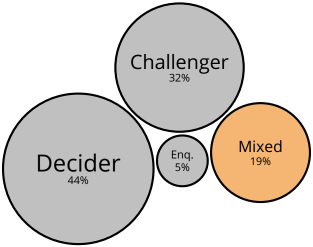


Figure: Distribution of Error Sources

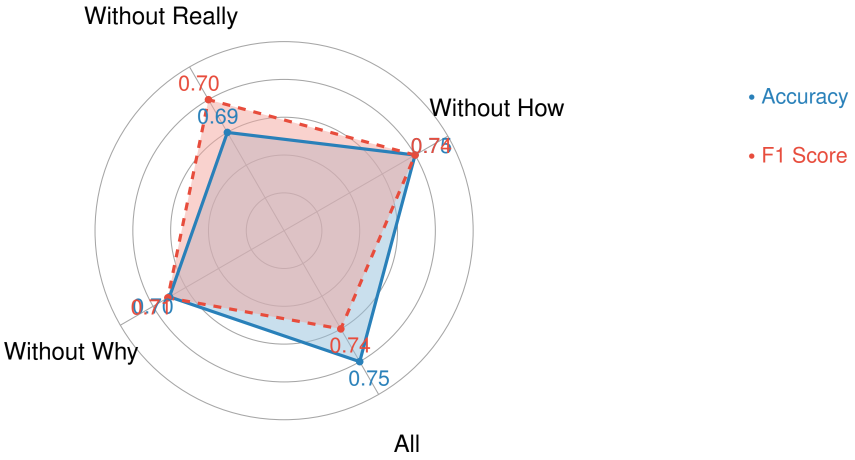
Impact of Challenge Prompts (RQ5)

- **Experiment Setup**:
  - Evaluated the impact of individual challenge prompts
  - Tested performance by excluding each question type

Scenarios	Accuracy	F1-score
With all questions	0.75	0.74
Without *How* questions	0.75	0.75
Without *Really* questions	0.69	0.70
Without *Why* questions	0.70	0.71

Table: Impact of individual challenge prompts

Visualizing Impact of Challenge Prompts



Impact of Mutation & Basic Challenges

- **Key Findings**:
  - Excluding mutation challenges reduced accuracy by 16% (from 75% to 63%)
  - Excluding basic challenges reduced accuracy by 8% (from 75% to 69%)

Scenarios	Accuracy	F1-score
With Basic and Mutation Challenges	0.75	0.74
Without Mutation Challenges	0.63	0.65
Without Basic Challenges	0.69	0.69

Table: Impact of mutation & basic challenges

Visualizing Impact of Mutation & Basic Challenges

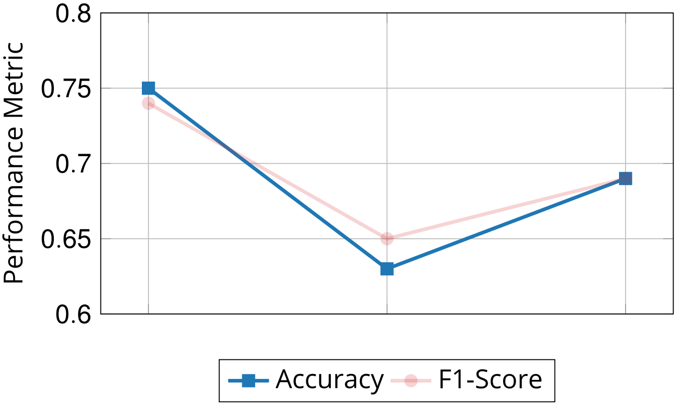


Figure: Impact of Mutation and Basic Challenges

Visualizing Impact of Mutation & Basic Challenges

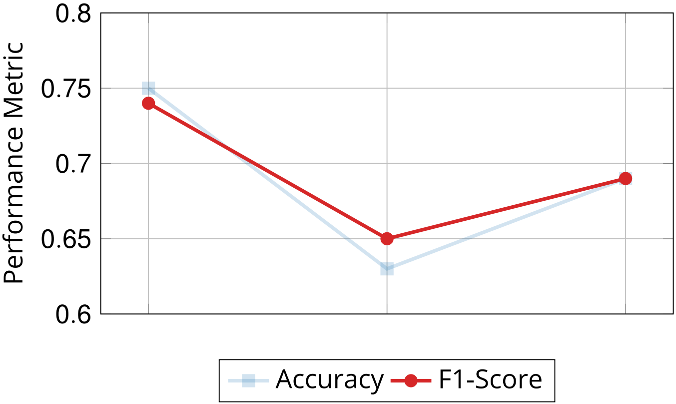


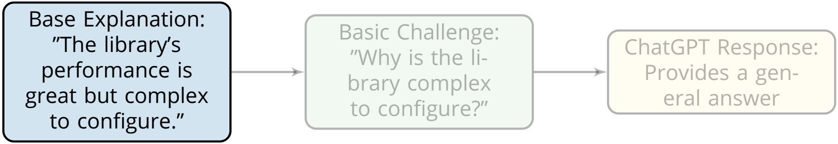
Figure: Impact of Mutation and Basic Challenges

Example of Mutation Impact

- **Scenario**:
  - Base Explanation: ”The library’s performance is great but complex to configure.”

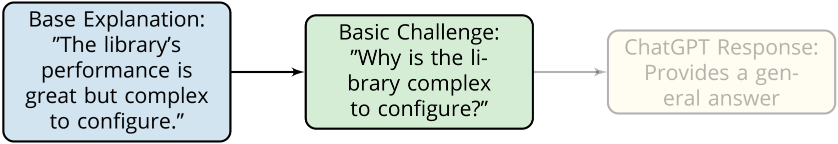
Example of Mutation Impact - Basic Challenge

- **Basic Challenge**:
  - Question: ”Why is the library complex to configure?”
  - ChatGPT Response: Provides a general answer



Example of Mutation Impact - Basic Challenge

- **Basic Challenge**:
  - Question: ”Why is the library complex to configure?”
  - ChatGPT Response: Provides a general answer



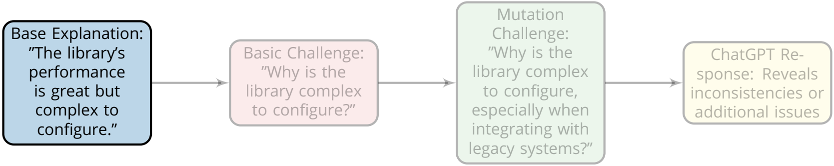
Example of Mutation Impact - Basic Challenge

- **Basic Challenge**:
  - Question: ”Why is the library complex to configure?”
  - ChatGPT Response: Provides a general answer

Base Explanation: ”The library’s performance is great but complex to configure.” → Basic Challenge: ”Why is the library complex to configure?” → ChatGPT Response: Provides a general answer

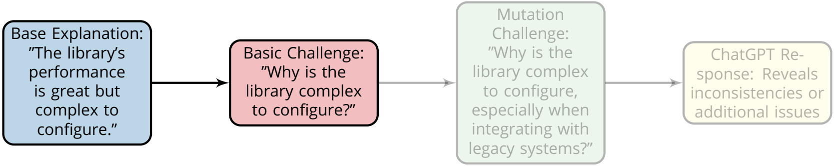
Example of Mutation Impact - Mutation Challenge

- **Mutation Challenge**:
  - Question: ”Why is the library complex to configure, especially when integrating with legacy systems?”
  - ChatGPT Response: Reveals inconsistencies or additional issues



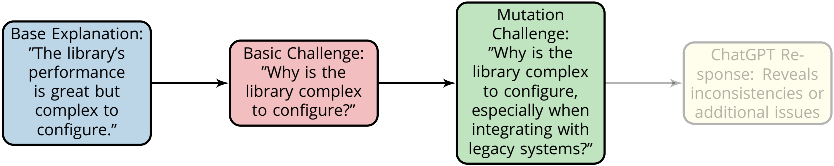
Example of Mutation Impact - Mutation Challenge

- **Mutation Challenge**:
  - Question: "Why is the library complex to configure, especially when integrating with legacy systems?"
  - ChatGPT Response: Reveals inconsistencies or additional issues



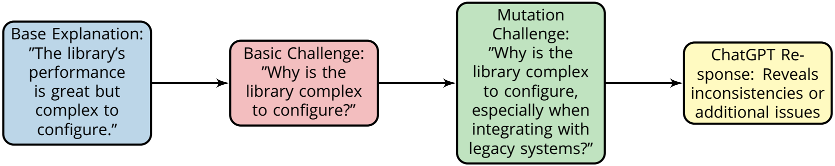
Example of Mutation Impact - Mutation Challenge

- **Mutation Challenge**:
  - Question: ”Why is the library complex to configure, especially when integrating with legacy systems?”
  - ChatGPT Response: Reveals inconsistencies or additional issues



Example of Mutation Impact - Mutation Challenge

- **Mutation Challenge**:
  - Question: ”Why is the library complex to configure, especially when integrating with legacy systems?”
  - ChatGPT Response: Reveals inconsistencies or additional issues



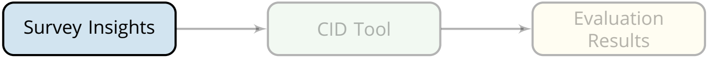
Limitations of CID

- **Dataset and Labeling Constraints**
  - Relies on Stack Overflow data
  - Variability and potential bias in human-annotated labels
- **Similarity Measurement Challenges**
  - Current metrics may struggle with complex or nuanced responses.
  - Potential for misclassifications due to inadequate similarity assessments.
- **Limited Scope and Generalizability**
  - Evaluated primarily on software library selection tasks.
  - Effectiveness on other SE tasks remains unexplored.

Outline

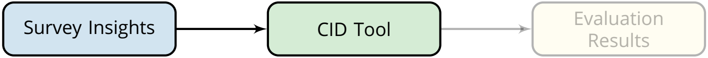
Conclusion

- **Recap of Research**:
  - Explored developers' reliance on ChatGPT
  - Identified concerns about response correctness



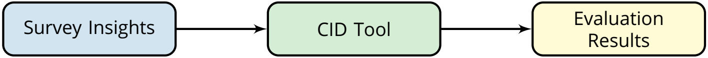
Conclusion

- **Recap of Research**:
  - Developed CID to detect incorrect ChatGPT responses



Conclusion

- **Recap of Research**:
  - Evaluted the performance of CID with relevant metrics



Limitations

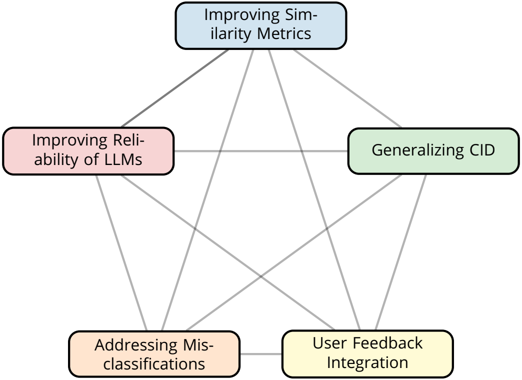
Dataset De-pendence

Similarity Measures

Scope of Evaluation

Summary of Limitations

Future Work



Future Work Roadmap

References I

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying.
arXiv preprint arXiv:2304.13734, 2023.

Myeongjun Jang and Thomas Lukasiewicz.
Consistency analysis of ChatGPT.
arXiv preprint arXiv:2303.06273, 2023.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales.
SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models.
arXiv preprint arXiv:2303.08896, 2023.

References II

Philip Feldman, James R. Foulds, and Shimei Pan.
Trapping LLM hallucinations using tagged context prompts.
arXiv preprint arXiv:2306.06085, 2023.

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

References III

Enrique Larios Vargas, Maurício Aniche, Christoph Treude, Magiel Bruntink, and Georgios Gousios.

Selecting third-party libraries: The practitioners' perspective.

In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 245–256, 2020.

Gias Uddin and Foutse Khomh.

OPINER: An opinion search and summarization engine for APIs.

In *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*, pages 978–983, 2017.

References IV

Gias Uddin, Olga Baysal, Latifa Guerrouj, and Foutse Khomh.
Understanding how and why developers seek and analyze API-related opinions.
IEEE Transactions on Software Engineering, 47(4):694–735, 2019.

Han Wang, Chunyang Chen, Zhenchang Xing, and John Grundy.
DiffTech: A tool for differencing similar technologies from question-and-answer discussions.
In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1576–1580, 2020.

References V

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh.
Beyond accuracy: Behavioral testing of NLP models with CheckList.
arXiv preprint arXiv:2005.04118, 2020.

Qingchao Shen, Junjie Chen, Jie M Zhang, Haoyu Wang, Shuang Liu, and Menghan Tian.
Natural test generation for precise testing of question answering software. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. pages 1–12.

Link to Orignal Paper

ChatGPT Incorrectness Detection in Software Reviews

Thank You!