FOSSEE INTERNSHIP REPORT

Logical Error Taxonomy Design and Large Language Model Evaluation for Educational Code Submissions

Submitted by:

[**Harshit Mishra**]

Under the guidance of

Dr. Kushal Shah

Dr. Prabhu Ramachandran

FOSSEE Project, IIT Bombay

Internship Duration: October 2025 – March 2026

Acknowledgement

I express my sincere gratitude to the FOSSEE Project at the Indian Institute of Technology Bombay for providing me with the opportunity to work as an Open Source Research Intern in the domain of Python and Large Language Model based educational research.

I am deeply thankful to my mentor, Dr. Kushal Shah, whose constant guidance, constructive feedback, and research-oriented discussions played a significant role in shaping my analytical thinking and experimental methodology. His emphasis on structured reasoning and empirical validation helped me approach complex research problems in a systematic manner.

I sincerely thank Dr. Prabhu Ramachandran for his valuable feedback and guidance, which helped strengthen the direction and quality of this research work.

I would also like to acknowledge the collaborative research environment provided by my fellow interns, whose independent taxonomy explorations and shared technical discussions contributed to a broader understanding of logical error categorisation in programming education.

Finally, I express my gratitude to my academic institution and faculty members for encouraging research participation and supporting my continuous learning journey. This internship experience has significantly influenced my perspective on open-source research, educational technology, and artificial intelligence driven evaluation systems.

Abstract

Programming education at scale requires robust mechanisms for analysing student code submissions and identifying conceptual misunderstandings. Traditional evaluation approaches rely heavily on test-case based correctness assessment, which often fails to capture deeper logical reasoning errors made by learners.

This internship research focuses on the design and empirical evaluation of a structured logical error taxonomy for educational Python programming submissions. The study investigates how modern Large Language Models interpret and classify logical errors under constrained taxonomy-driven prompting frameworks. A multi-iteration experimental protocol was developed to analyse both dominant error prediction behaviour and multi-label reasoning expansion across multiple open-weight and closed-source language models.

A comprehensive experimental pipeline was implemented using Python to automate dataset preprocessing, prompt orchestration, model querying, output validation, retry handling, and structured result storage. The evaluation was conducted on a curated dataset of real-world student submissions obtained from an educational programming platform.

The research further explores inter-model agreement dynamics, taxonomy sensitivity, prediction stability, and anomaly patterns that emerge when models are forced to produce single-label versus multi-label classifications. The findings highlight the inherent ambiguity in logical error identification and demonstrate the significant influence of taxonomy design on model behaviour and agreement metrics.

This work contributes toward developing scalable AI-assisted educational evaluation frameworks and provides insights into the design of pedagogically meaningful logical error categorisation systems.

Chapter 1

Introduction

The rapid expansion of online programming education platforms has led to a substantial increase in the volume of student code submissions requiring evaluation. While automated grading systems efficiently verify functional correctness through predefined test cases, they often lack the capability to interpret the underlying logical reasoning errors present in incorrect solutions. As a result, students may receive limited conceptual feedback, which restricts effective learning progression.

Logical error analysis represents a critical component of pedagogical assessment in programming education. Unlike syntax errors or runtime failures, logical errors arise when a program executes successfully but produces incorrect or suboptimal results due to flawed reasoning, incorrect algorithmic strategy, or misunderstanding of problem specifications. Identifying and categorising such errors requires a deeper semantic understanding of program intent and execution flow.

Recent advancements in Large Language Models have opened new research avenues for automated reasoning-based evaluation of programming tasks. These models exhibit strong capabilities in code comprehension, algorithmic reasoning, and contextual interpretation of natural language problem descriptions. However, their behaviour under constrained classification tasks, particularly in educational contexts, remains an active research area.

This internship research was conducted with the objective of investigating how structured logical error taxonomies influence the reasoning behaviour of Large Language Models when analysing real-world student submissions. The study integrates taxonomy engineering, prompt design, experimental automation, and statistical agreement analysis into a unified research framework.

The work was carried out as part of the FOSSEE initiative, which promotes the development and adoption of open-source computational tools for education. Through this internship, a systematic experimental pipeline was designed to evaluate multi-model reasoning consistency and to explore the feasibility of AI-assisted conceptual feedback systems for large-scale programming education environments.

Chapter 2

Motivation and Research Context

Educational institutions increasingly rely on digital platforms to deliver programming instruction to large cohorts of learners. While automated assessment tools provide scalability, they often fail to offer detailed diagnostic insights into the nature of student mistakes. This gap highlights the need for intelligent systems capable of understanding conceptual reasoning patterns rather than merely verifying output correctness.

Logical error classification frameworks can serve as structured pedagogical tools for diagnosing student misunderstandings. However, designing such taxonomies requires balancing conceptual abstraction with structural specificity. Overly coarse taxonomies may fail to capture meaningful distinctions between error types, whereas highly granular taxonomies may introduce ambiguity and reduce evaluator agreement.

The emergence of Large Language Models as reasoning agents introduces the possibility of automating logical error detection at scale. Nevertheless, model behaviour may vary significantly depending on prompt constraints, taxonomy definitions, and classification strategies. Understanding these dynamics is essential before deploying AI-driven evaluation systems in real educational settings.

This research was motivated by the need to empirically analyse model reasoning stability, taxonomy sensitivity, and agreement patterns across multiple classification paradigms. By conducting controlled experiments using real student code submissions, the study aims to contribute toward designing reliable and pedagogically meaningful AI-assisted feedback mechanisms.

Chapter 3

Design of Logical Error Taxonomy

A structured logical error taxonomy was designed as a conceptual framework to categorise reasoning flaws observed in student programming submissions. The taxonomy aimed to balance pedagogical interpretability with computational feasibility, ensuring that each category represented a distinct dimension of logical reasoning failure.

The proposed taxonomy consists of six principal categories along with a neutral class representing the absence of logical error. These categories were defined based on extensive observation of student solution patterns and common conceptual misunderstandings encountered in introductory programming environments.

3.1 Proposed Taxonomy (AXONOMY_TEXT_har)

The proposed taxonomy used in this work is defined as follows:

- **A – Boundary & Indexing:** Errors in loop bounds or indexing (off-by-one, skipped or extra element).
- **B – Conditional / Boolean:** Incorrect condition, wrong boolean operator, reversed logic, unreachable branch.
- **C – State Management:** Incorrect handling of variables across execution (not resetting counters, stale state).
- **D – Algorithmic Strategy:** Fundamental flaw in approach or data structure; logic does not solve the task.
- **E – Edge Case Handling:** Fails only on boundary inputs (empty input, single element, zero, negative values).
- **F – Specification Misunderstanding:** Code does not follow the problem statement, output format, or required function behavior.

- **NONE – No Error:** No logical error present.

3.2 Boundary and Indexing Errors

Boundary-related errors arise when iterative structures fail to correctly traverse the required input domain. These include off-by-one conditions, incomplete traversal ranges, and unintended inclusion of extraneous elements. Such errors often reflect incomplete understanding of loop control variables and termination conditions.

3.3 Conditional and Boolean Reasoning Errors

Conditional logic errors occur when program flow decisions are governed by incorrect predicates. This includes reversed logical comparisons, misuse of boolean operators, and unreachable execution branches. These errors significantly affect program correctness despite syntactic validity.

3.4 State Management Errors

State-related errors emerge when variables that maintain cumulative program context are incorrectly initialised, updated, or reset. These mistakes frequently appear in aggregation problems and iterative algorithm implementations where previous computational state influences subsequent results.

3.5 Algorithmic Strategy Errors

Algorithmic errors represent fundamental flaws in problem-solving approach. Programs exhibiting such errors may execute successfully but fail to implement the conceptual logic required to solve the intended task. This category captures deep reasoning deficiencies rather than surface-level structural issues.

3.6 Edge Case Handling Errors

Programs that perform correctly for typical inputs but fail under boundary scenarios fall into this category. Examples include failure to handle empty datasets, single-element structures, negative values, or extreme numeric ranges. These errors indicate incomplete robustness considerations.

3.7 Specification Interpretation Errors

Specification-related errors arise when the implemented logic deviates from the problem statement requirements. This includes incorrect output formatting, unintended function behaviour, or partial implementation of required features.

3.8 No Logical Error Category

The taxonomy includes a neutral class to represent submissions that exhibit no identifiable logical reasoning flaw under the defined classification scheme.

Chapter 4

Experimental Methodology

The experimental framework was designed to evaluate the logical error classification behaviour of multiple Large Language Models under controlled prompting constraints. The methodology emphasised reproducibility, structured iteration, and systematic result collection.

4.1 Dataset Preparation

A curated dataset consisting of one hundred real-world student programming submissions was utilised for experimentation. Each dataset entry included a textual problem description and a corresponding Python solution submitted by learners in an educational programming environment.

Prior to experimentation, preprocessing steps were applied to remove extraneous input-output scaffolding code that could bias model reasoning. This ensured that evaluation focused primarily on algorithmic and logical constructs rather than formatting artefacts.

4.2 Multi-Taxonomy Evaluation Framework

To analyse taxonomy sensitivity, three independently designed logical error classification schemes were evaluated. Each taxonomy emphasised a different perspective of reasoning analysis, including conceptual abstraction, structural granularity, and pedagogical decomposition.

This multi-taxonomy approach enabled comparative evaluation of model behaviour under varying classification constraints.

4.3 Multi-Iteration Prompting Strategy

A two-stage prompting protocol was implemented to investigate reasoning stability.

4.3.1 Iteration One: Dominant Error Identification

In the first iteration, models were constrained to select a single dominant logical error category. This stage simulated practical educational scenarios where automated systems must provide concise diagnostic feedback.

4.3.2 Iteration Two: Multi-Label Error Expansion

The second iteration relaxed classification constraints, allowing models to identify all applicable logical error categories. This stage enabled analysis of reasoning expansion behaviour and identification of latent conceptual interpretations.

4.4 Automated Experiment Execution Pipeline

A Python-based orchestration system was developed to automate the evaluation workflow. The pipeline performed the following sequential operations:

- Dynamic prompt construction using taxonomy definitions
- Multi-model API querying through a unified routing interface
- Output validation and sanitisation
- Retry handling with exponential backoff strategy
- Incremental result persistence to ensure experiment continuity
- Model status tracking to handle access failures or invalid outputs

The automation framework ensured experimental consistency while enabling large-scale evaluation across multiple reasoning agents.

Chapter 5

Mathematical Framework for Agreement Analysis

Quantitative evaluation of model reasoning behaviour required the formulation of statistical agreement metrics. Multi-label classification outputs were analysed using set-based similarity measures.

5.1 Jaccard Similarity Measure

Agreement between predicted label sets and reference annotations was computed using the Jaccard similarity coefficient defined as:

$$J(P, M) = \frac{|P \cap M|}{|P \cup M|}$$

where P represents the predicted label set generated by a model and M denotes the manually assigned reference label set.

This metric provides a continuous measure of overlap ranging from zero (no agreement) to one (complete agreement).

5.2 Containment-Based Agreement Criteria

Additional evaluation criteria were defined to capture nuanced agreement relationships:

- Prediction Subset Criterion: All predicted labels are contained within the reference annotation set.
- Reference Coverage Criterion: All reference labels are included in the predicted set.
- Partial Overlap Criterion: At least one common label exists between prediction and reference.

These criteria enabled multi-dimensional interpretation of classification behaviour beyond strict exact match evaluation.

5.3 Inter-Model Agreement Analysis

Pairwise agreement between models was evaluated by computing average similarity across all dataset samples. This analysis provided insights into clustering patterns among reasoning agents and highlighted divergence behaviour under structurally granular taxonomies.

5.4 Stability Evaluation Across Iterations

Model stability was analysed by examining whether dominant single-label predictions were retained within subsequent multi-label expansions. This metric served as an indicator of reasoning consistency and confidence calibration.

Chapter 6

Experimental Results and Observations

The experimental evaluation revealed several important behavioural patterns in Large Language Model based logical error classification. These observations provide insight into both the strengths and limitations of taxonomy-driven reasoning systems.

6.1 Dominant Error Prediction Behaviour

During the constrained single-label classification phase, models frequently converged on high-level reasoning categories such as algorithmic strategy or specification interpretation. This indicates that modern language models tend to prioritise conceptual correctness over structural or syntactic reasoning signals.

However, significant disagreement was observed in submissions containing multiple independent logical flaws. In such cases, different models demonstrated varied anchoring strategies, with some focusing on algorithmic validity while others prioritised adherence to problem specifications.

6.2 Multi-Label Reasoning Expansion

When classification constraints were relaxed, inter-model agreement improved substantially. Models that previously appeared to disagree often produced overlapping multi-label interpretations, suggesting that apparent conflict in single-label settings may arise from forced prioritisation rather than genuine reasoning divergence.

This finding highlights the importance of evaluation protocol design in assessing reasoning systems. Multi-label classification provides a more faithful representation of model understanding in complex logical scenarios.

6.3 Taxonomy Sensitivity Effects

Comparative analysis across multiple taxonomy structures demonstrated that classification agreement is strongly influenced by categorisation granularity. Conceptually broad taxonomies produced higher consensus levels, whereas structurally granular schemes amplified disagreement and label expansion behaviour.

These results emphasise the need for careful taxonomy engineering when designing AI-assisted educational evaluation frameworks. Pedagogical interpretability must be balanced with classification stability to ensure meaningful diagnostic feedback.

6.4 Model Stability Across Iterations

Stability analysis indicated that certain models maintained strong internal consistency, retaining dominant predictions within expanded multi-label outputs. Conversely, other models exhibited reasoning volatility, frequently altering their classification sets upon re-evaluation.

Such behaviour suggests that logical error detection remains an inherently ambiguous task even for advanced reasoning systems. Iterative prompting frameworks may therefore be essential for achieving reliable automated assessment.

6.5 Pairwise Model Agreement Heatmap Analysis

To better understand inter-model reasoning similarity, pairwise agreement matrices were visualised using heatmap representations. These figures illustrate how consistently different language models align in their logical error classifications across taxonomies and prompting iterations.

6.5.1 Iteration-1 Single Label Agreement (Conceptual Taxonomy)

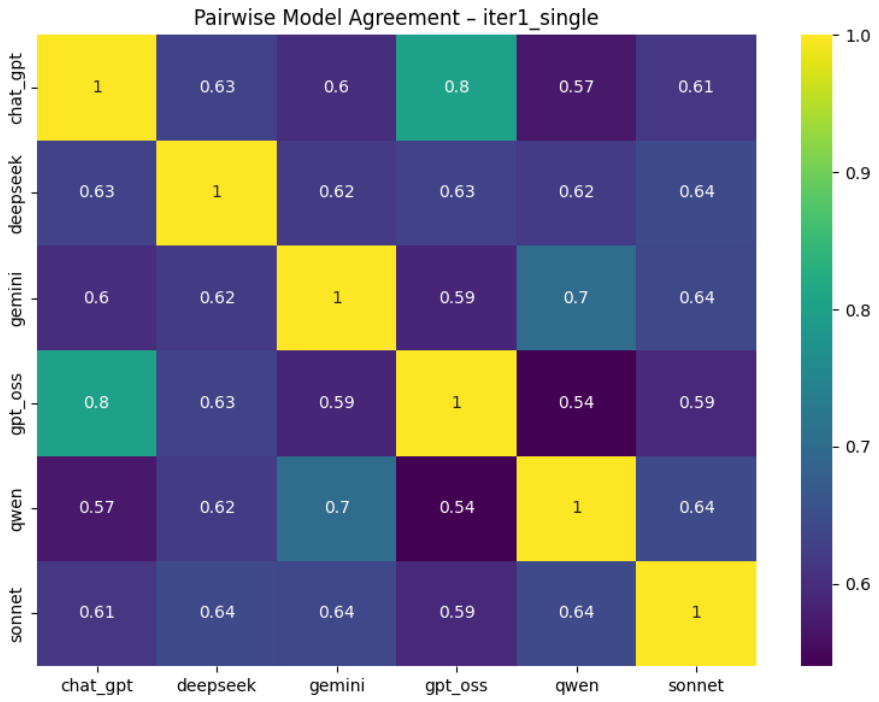

Figure 6.1: Pairwise agreement between models under single-label constraint using the conceptual logical error taxonomy.

The agreement levels remain moderate across most model pairs, indicating that forcing a dominant label introduces prioritisation-based divergence in reasoning behaviour.

6.5.2 Iteration-1 Single Label Agreement (Structural Taxonomy)

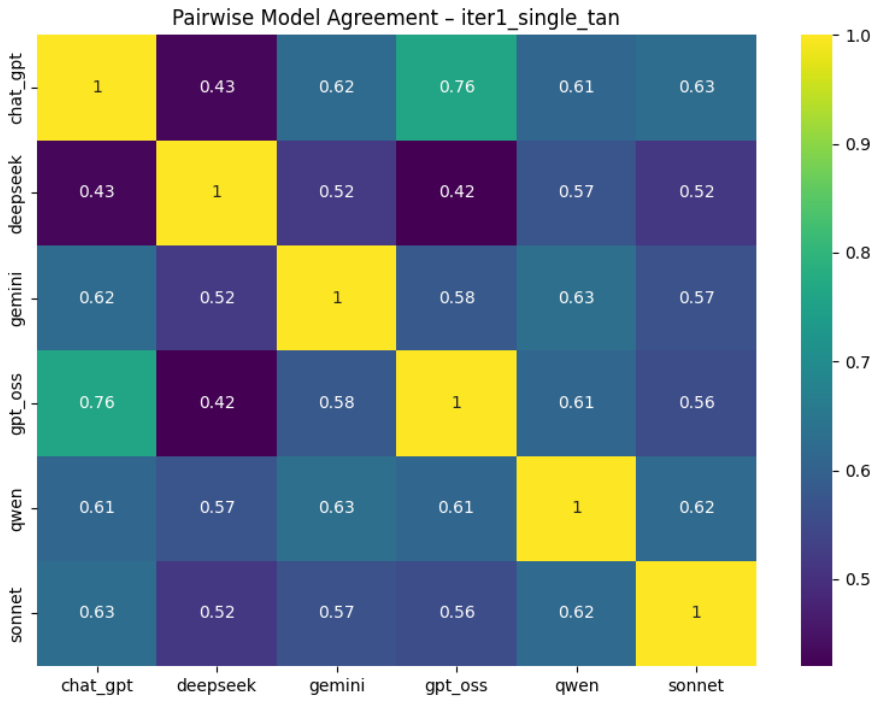

Figure 6.2: Pairwise agreement under structurally granular taxonomy showing increased disagreement patterns.

Structural decomposition of logical errors significantly reduces consensus, suggesting that finer categorisation amplifies interpretation variability.

6.5.3 Iteration-1 Single Label Agreement (Pedagogical Taxonomy)

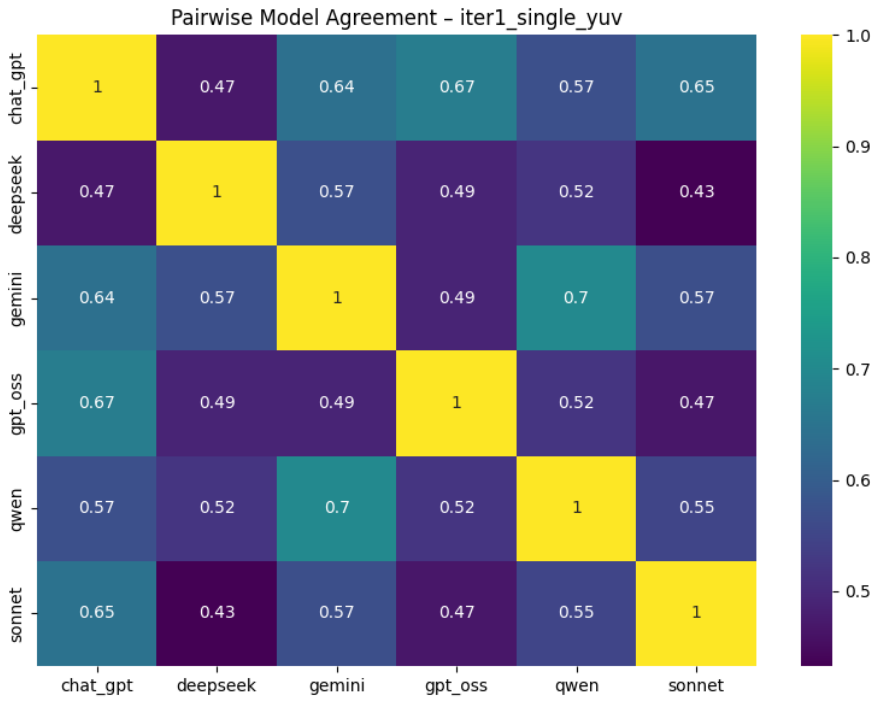

Figure 6.3: Agreement trends using pedagogically oriented taxonomy demonstrating moderate clustering behaviour.

Pedagogical categories produce balanced abstraction, leading to moderate but stable inter-model similarity.

6.5.4 Iteration-2 Multi-Label Agreement (Conceptual Taxonomy)

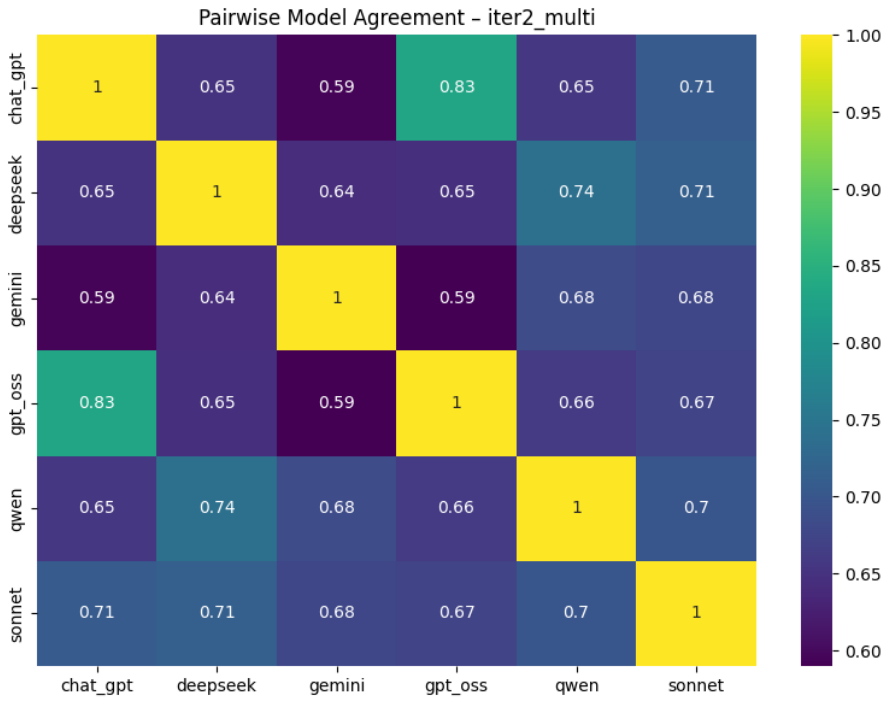

Figure 6.4: Improved agreement levels when models are allowed multi-label reasoning expansion.

Agreement increases significantly in multi-label settings, reinforcing the hypothesis that earlier conflict is largely due to forced prioritisation.

6.5.5 Iteration-2 Multi-Label Agreement (Structural Taxonomy)

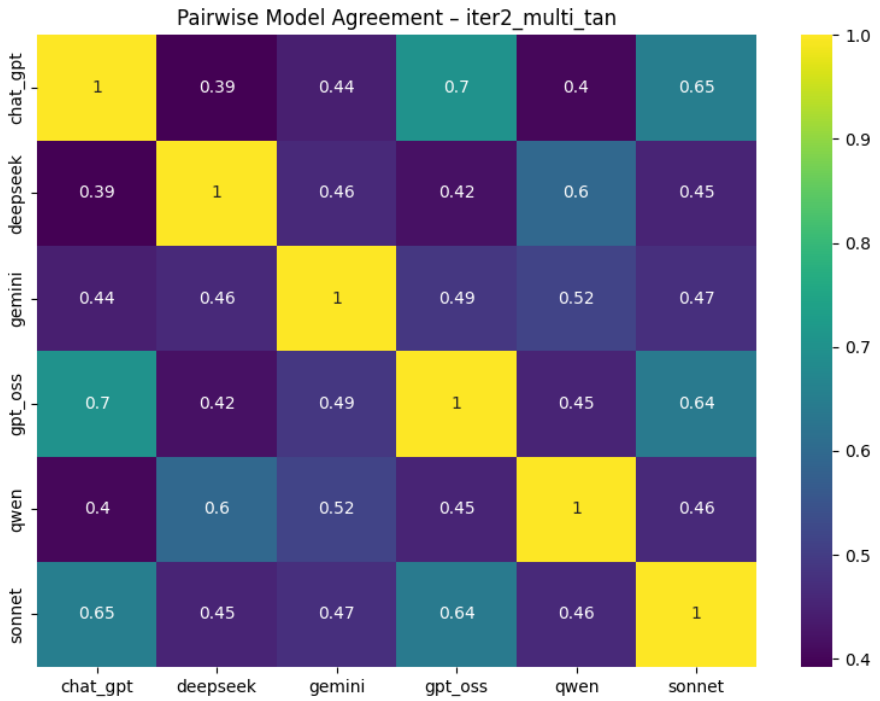

Figure 6.5: Structural taxonomy continues to exhibit fragmented agreement even in multi-label conditions.

Despite reasoning expansion, structural granularity sustains disagreement, highlighting taxonomy sensitivity.

6.5.6 Iteration-2 Multi-Label Agreement (Pedagogical Taxonomy)

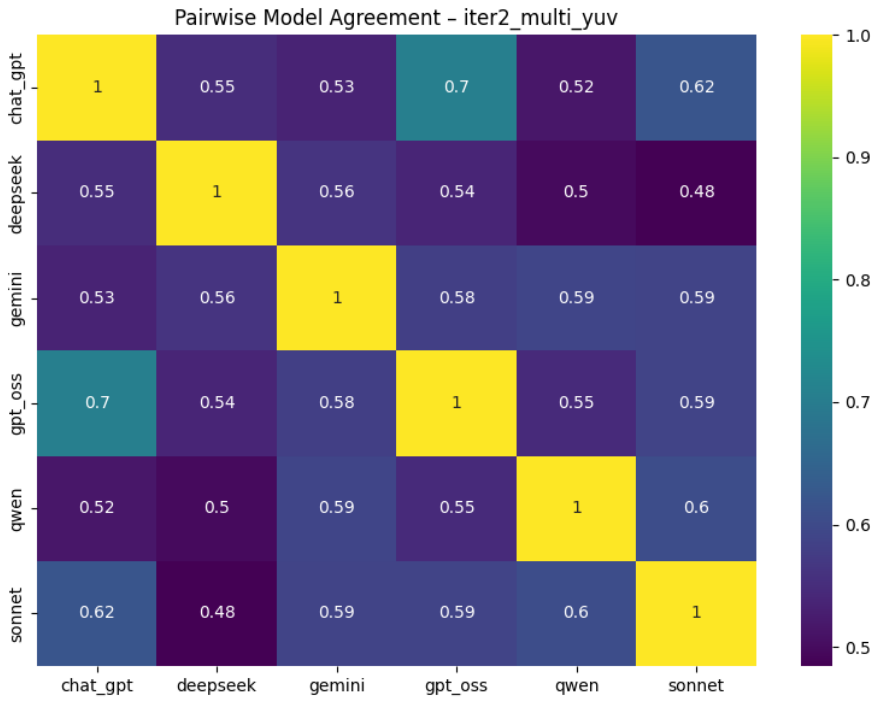

Figure 6.6: Pedagogical taxonomy agreement trends under multi-label classification.

This configuration demonstrates moderate convergence, indicating conceptual overlap among model reasoning patterns.

Chapter 7

Discussion

The findings from this research contribute toward a deeper understanding of how artificial intelligence systems interpret logical reasoning errors in educational programming contexts. While language models demonstrate impressive capability in code comprehension, their classification behaviour is significantly shaped by prompt structure, taxonomy design, and output constraints.

The observed sensitivity to taxonomy granularity suggests that educational feedback systems must be carefully calibrated to avoid over-fragmentation of conceptual error categories. Excessively detailed classification schemes may introduce unnecessary ambiguity, thereby reducing agreement reliability.

Furthermore, the improvement in agreement under multi-label conditions indicates that student code submissions often contain layered reasoning mistakes rather than singular dominant errors. Automated systems designed for conceptual feedback should therefore consider hierarchical or probabilistic classification strategies instead of rigid single-label outputs.

Another important implication concerns human annotation consistency. Cases where strong inter-model agreement contradicted manual labels highlight the possibility of subjective interpretation in logical error categorisation. This suggests that collaborative annotation protocols and consensus-based validation may be necessary in future educational dataset development.

Chapter 8

Internship Learning Experience

This internship provided a comprehensive exposure to research-oriented software development within an open-source academic ecosystem. Working in a remote collaborative environment required disciplined time management, independent problem solving, and continuous engagement with technical literature.

A significant learning outcome involved the practical implementation of large-scale experimental pipelines. Designing automated evaluation systems demanded careful attention to reproducibility, error handling, and data integrity considerations. The experience of integrating multiple language model interfaces and managing iterative experiment execution enhanced my understanding of real-world research engineering workflows.

The process of developing and refining a logical error taxonomy also contributed to strengthening my conceptual clarity regarding programming pedagogy. Analysing diverse student solution patterns enabled a deeper appreciation of how learners approach algorithmic problem solving and where conceptual misunderstandings typically arise.

Collaborative discussions with fellow interns working on alternative taxonomy perspectives broadened my analytical viewpoint. Observing differences in categorisation philosophy reinforced the importance of evaluation framework design in research studies involving human-interpretable classifications.

8.1 Technical Skills Acquired

- Advanced Python based research pipeline development
- Prompt engineering for structured reasoning tasks
- Multi-model API orchestration and experiment automation
- Statistical agreement analysis using set-based similarity metrics
- Data preprocessing and experimental reproducibility design

- Research documentation and analytical interpretation writing

8.2 Challenges Encountered

One of the primary challenges during this internship involved managing the ambiguity inherent in logical error classification. Unlike deterministic software engineering tasks, research experimentation required iterative refinement of hypotheses, evaluation criteria, and analytical interpretations.

Another challenge involved ensuring stability of automated experiments when interacting with multiple external model interfaces. Implementing retry strategies, output validation mechanisms, and incremental result storage systems was essential to maintain experimental continuity.

Chapter 9

Conclusion

This research internship successfully explored the feasibility of taxonomy-driven logical error classification using modern Large Language Models. The study demonstrated that while language models possess strong conceptual reasoning capabilities, their classification behaviour is highly dependent on evaluation protocol design and taxonomy structure.

Multi-iteration prompting strategies were found to provide valuable insights into reasoning stability and agreement dynamics. The results highlight the potential of AI-assisted conceptual feedback systems in large-scale programming education, while also underscoring the need for careful pedagogical alignment.

The experimental pipeline developed during this internship establishes a foundation for future research in automated educational assessment and intelligent tutoring systems.

Chapter 10

Future Work

Future research directions emerging from this work include:

- Development of hierarchical logical error taxonomies integrating conceptual and structural perspectives
- Exploration of probabilistic multi-label classification frameworks for educational feedback
- Integration of execution trace analysis with language model reasoning outputs
- Creation of large benchmark datasets with consensus-validated human annotations
- Investigation of adaptive prompting strategies based on student learning profiles

Advancements in these areas may contribute toward building scalable intelligent tutoring environments capable of providing personalised conceptual guidance to learners.

Appendix A

Summary of Experimental Pipeline

The automated evaluation system developed during this internship followed a structured workflow designed to ensure reproducibility and scalability.

1. Dataset ingestion and preprocessing of student code submissions
2. Removal of auxiliary input-output scaffolding elements
3. Dynamic construction of taxonomy-driven prompts
4. Sequential querying of multiple language models
5. Validation and normalisation of classification outputs
6. Retry handling using exponential backoff strategies
7. Row-wise persistence of experimental results for fault tolerance
8. Statistical agreement computation and analytical reporting

This modular pipeline architecture enables extension to additional taxonomies, datasets, and reasoning models in future research studies.