Mini-Project 2

ASE 2024

Paul Chevelev · Silvan Verhoeven

Specifications I - Complexity score

Halstead volume: Score for complexity that deals directly with ease of understanding

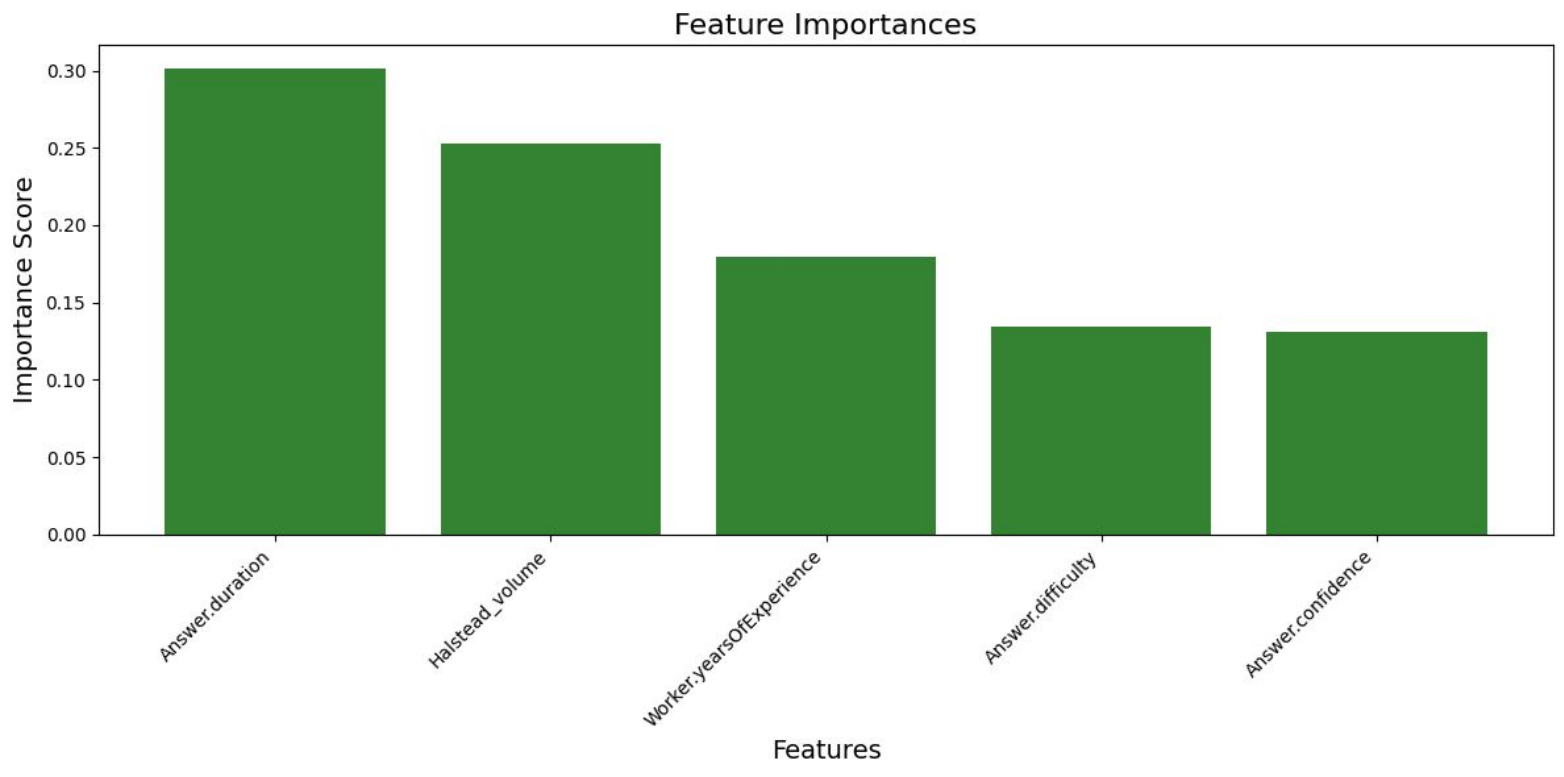
Specifications I - Model

Random Forest

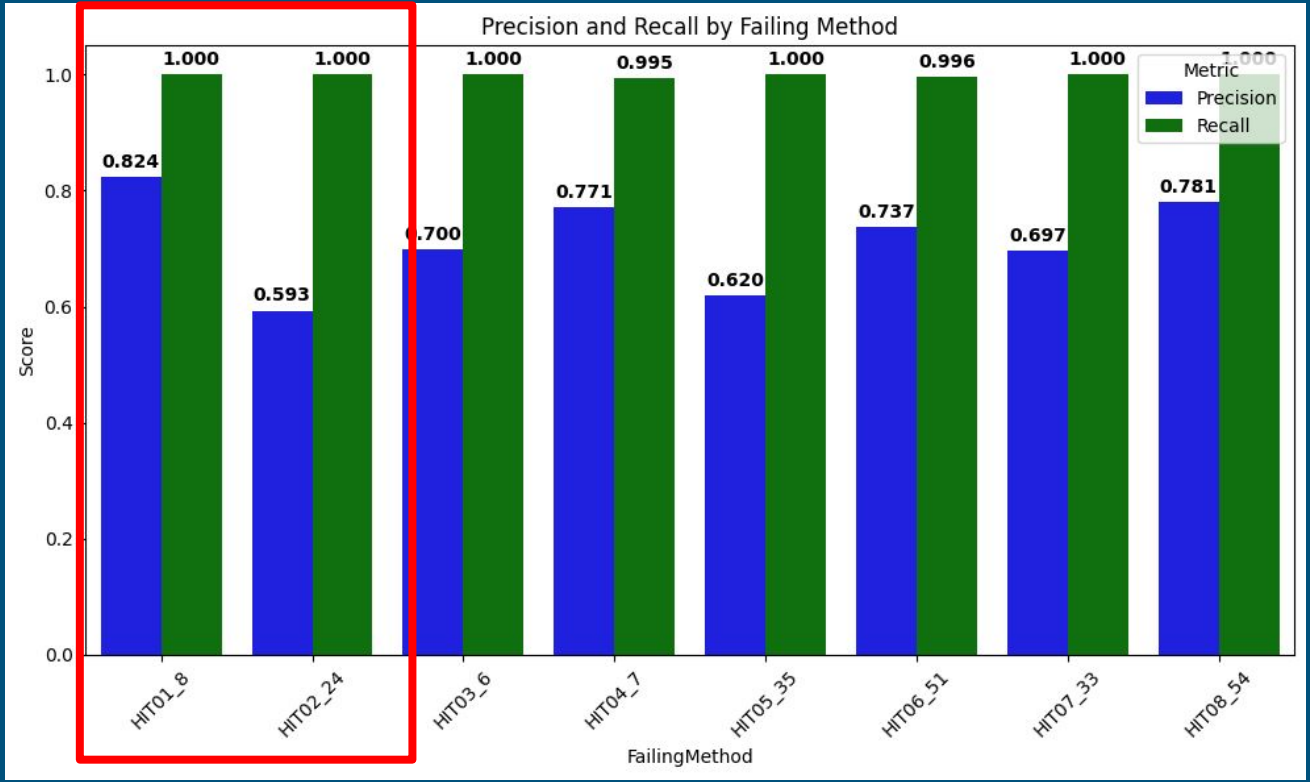
Input: Answer.duration, Answer.confidence, Answer.difficulty, Worker.yearsOfExperience, Halstead_volume of explanation,

Best params:
{'max_depth': 10,
 'max_features': 2,
 'min_samples_leaf': 10,
 'min_samples_split': 2,
 'n_estimators': 75}

Specifications I - Model

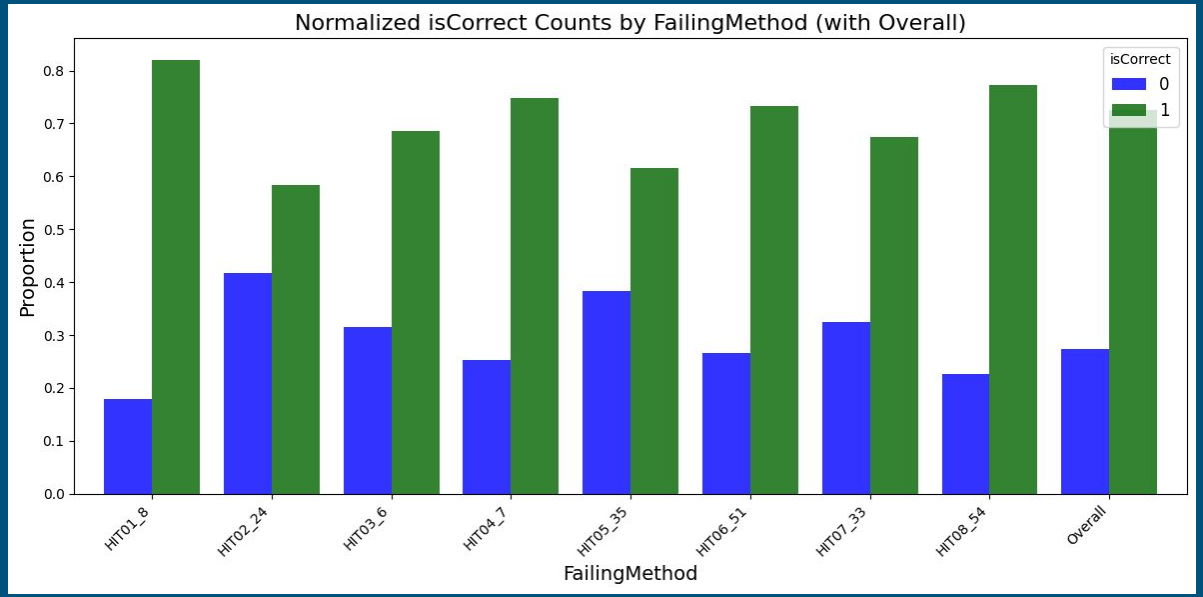


Specifications I - Model

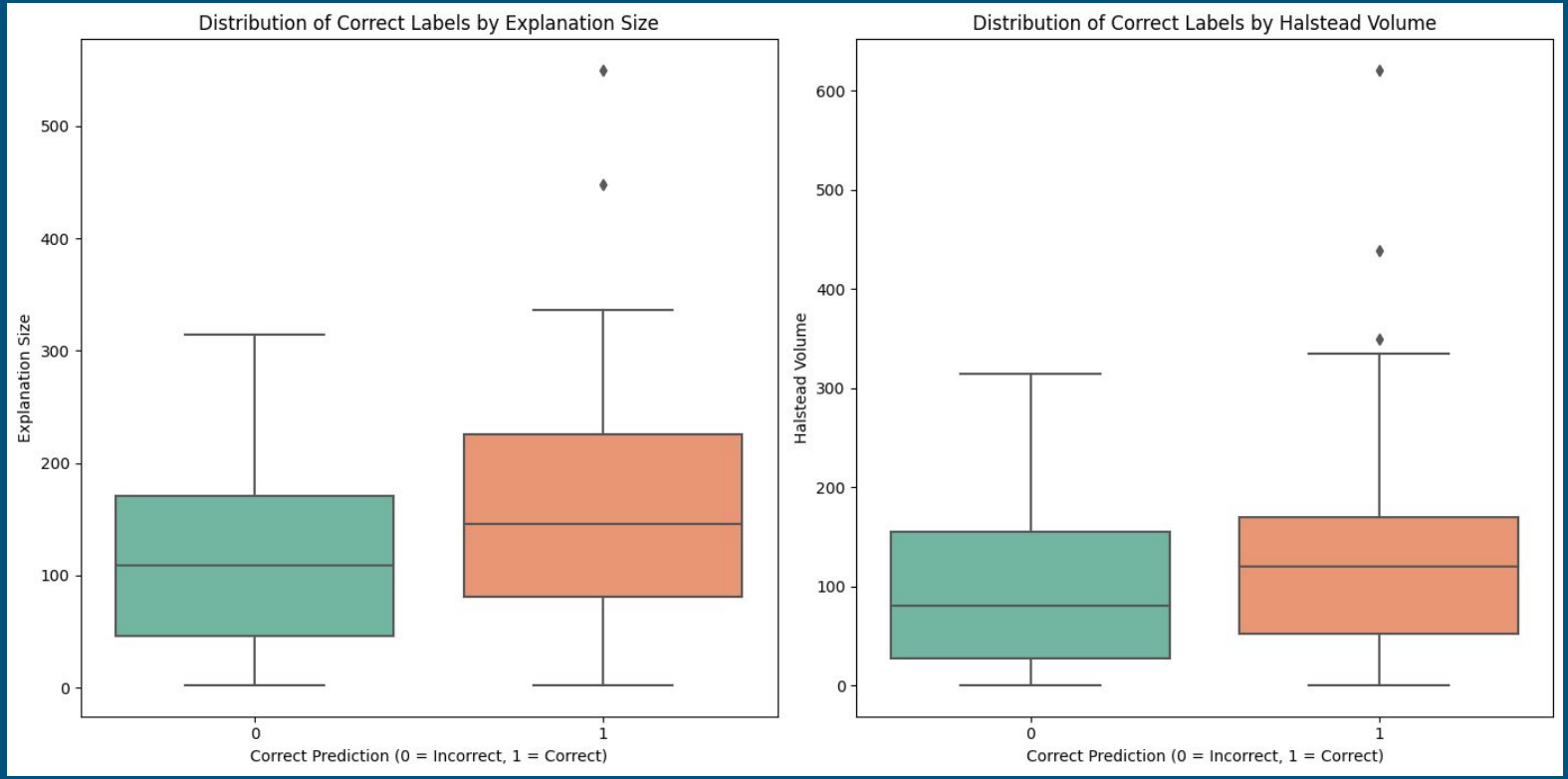


Specifications I - Model

Performance seems alright however recall is nearly always 1 which is due to it nearly always predicting the label to be 1



Specifications I - Inspection task



Specifications II – Consolidate Explanations

Pilot Prompts

- Direct: copied task
  - Intro/Outro in Answer
  - Extensive formatting
  - Elaborate sentences
  - Explanation of code, error root cause, detailed solution steps, additional remarks
- Developer Persona: Like Direct, and speak like a developer
  - Similar to above, without intro/outro
- Short: Allow more deviation from originals; be short
  - Few well formulated, whole sentences
  - Root cause and fix
  - Occasional code blocks
- No Formatting: Like Short, and no code formatting and possibly no complete sentences
  - Similar to above, with mix of complete and sentence fragments
  - Still uses inline code formatting

Scores

	BLEU	ROUGE
Short	{'HIT01_8': 0.248, 'HIT02_24': 0.04, 'HIT03_6': 0.113, 'HIT04_7': 0.134, 'HIT05_35': 0.163, 'HIT06_51': 0.042, 'HIT07_33': 0.25, 'HIT08_54': 0.2}	{'HIT01_8': {'rouge1': p=0.31, r=0.64, 'rouge2': p=0.13, r=0.29, 'rougeL': p=0.29, r=0.6}, 'HIT02_24': {'rouge1': p=0.34, r=0.51, 'rouge2': p=0.07, r=0.36, 'rougeL': p=0.20, r=0.30}, 'HIT03_6': {'rouge1': p=0.47, r=0.41, 'rouge2': p=0.10, r=0.43, 'rougeL': p=0.17, r=0.42}, 'HIT04_7': {'rouge1': p=0.44, r=0.51, 'rouge2': p=0.12, r=0.16, 'rougeL': p=0.21, r=0.24}, 'HIT05_35': {'rouge1': p=0.39, r=0.48, 'rouge2': p=0.12, r=0.31, 'rougeL': p=0.24, r=0.3}, 'HIT06_51': {'rouge1': p=0.45, r=0.46, 'rouge2': p=0.13, r=0.09, 'rougeL': p=0.25, r=0.26}, 'HIT07_33': {'rouge1': p=0.33, r=0.52, 'rouge2': p=0.22, r=0.23, 'rougeL': p=0.30, r=0.31}, 'HIT08_54': {'rouge1': p=0.54, r=0.38, 'rouge2': p=0.10, r=0.25, 'rougeL': p=0.2, r=0.48}}
No Formatting	{'HIT01_8': 0.211, 'HIT02_24': 0, 'HIT03_6': 0, 'HIT04_7': 0, 'HIT05_35': 0.2, 'HIT06_51': 0.013, 'HIT07_33': 0.08, 'HIT08_54': 0}	{'HIT01_8': {'rouge1': p=0.35, r=0.64, 'rouge2': p=0.13, r=0.25, 'rougeL': p=0.30, r=0.56}, 'HIT02_24': {'rouge1': p=0.00, r=0.00, 'rouge2': p=0.00, r=0.00, 'rougeL': p=0.00, r=0.00}, 'HIT03_6': {'rouge1': p=0.00, r=0.00, 'rouge2': p=0.00, r=0.00, 'rougeL': p=0.00, r=0.00}, 'HIT04_7': {'rouge1': p=0.00, r=0.00, 'rouge2': p=0.00, r=0.00, 'rougeL': p=0.00, r=0.00}, 'HIT05_35': {'rouge1': p=0.29, r=0.70, 'rouge2': p=0.15, r=0.37, 'rougeL': p=0.31, r=0.28}, 'HIT06_51': {'rouge1': p=0.35, r=0.17, 'rouge2': p=0.12, r=0.00, 'rougeL': p=0.31, r=0.15}, 'HIT07_33': {'rouge1': p=0.48, r=0.52, 'rouge2': p=0.14, r=0.12, 'rougeL': p=0.30, r=0.32}, 'HIT08_54': {'rouge1': p=0.00, r=0.00, 'rouge2': p=0.00, r=0.00, 'rougeL': p=0.000 r=0.00}}

Reflection Task – Concerns

Guaranteeing Data Quality

- Diversity
  - bug types
  - programmers
  - explanations (similar way of expression needed)
- Resource intense; skilled workers and time needed
- Interpretability; clear scope of explanation needed
- Quality of explanations?
  - Explanation leads to fix/diagnosis

Up-to-date Classifier

- Changes in programmer demographic: new expressions and programming languages
  - Gradual shifts in explanations
  - Possibly different
  - → Track demographic changes and regularly retrain the classifier
- Bug type changes: classifier agnostic to actual content
  - Relies on workers' properties and explanations
  - Gathering high quality data is difficult

Output Quality Testing (classifier and LLM)

- For LLM: Simple metrics give first indications
  - output length and complexity
  - amount of markup (dividers, code blocks, …)
  - …
- End-to-end evaluation
- Test difficult scenarios
  - Long bug reports, complex bug reports, multiple bugs per report…

Quality Estimation of consolidated Explanations

- No “gold standard” to compare to
  - Rely on survey data and previous scores (ROUGE, BLEU, …)
- Other options
  - Evaluation by other LLM
  - Expert evaluation
  - “Success rate” of explanation

(Integration Debugging (Classifier <> LLM))

- Bad LLM outputs can have two sources
  - LLM changes
  - Degraded classifier (garbage in, garbage out)
- Mitigation strategy
  - Change one part at a time and rolling release, monitor consistently
  - Version changes – rollback on degradation