Mini Project 2:

Automation for Consolidating Bug Report Explanations

Smilla Fox, Elena Gensch, Conrad Halle

Train Model - Preprocessing

 +  +  = 

- Answer.explanation:
  - TTR to cover Lexical richness
  - Flesch Kincaid to cover Lexical readability
  - No Halstead volume since it works best for code
- Feature selection (remove label cols and text)
- One-Hot Encoding of List data (e.g. where participants learned to code) with MultiLabelBinarizer
  - additional LLM generated mapping to combine entries like `“Other Books”` and `“there was no \"internet\". I had a TRS-80 for gosh sake. We learned from books! Made of paper. Odd concept”`
- Label Encoding for Categorical Columns

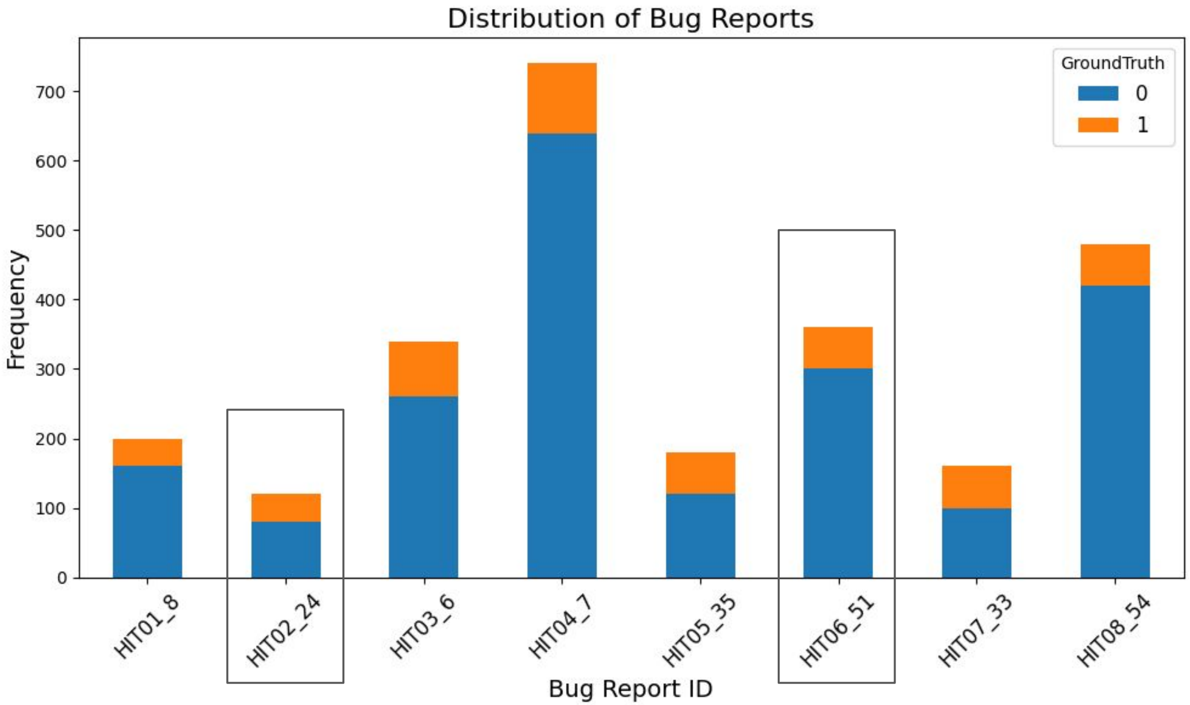
Flesh Kincaid: $$206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$

TTR: $$\frac{numberofuniquewords}{numberofwords}$$

Hidden Slide

Train Model - Split

- Holdout Set:
    - HIT02_24
    - HIT06_51
- Train Set
    - HIT01_8
    - HIT03_6
    - HIT04_7
    - HIT05_35
    - HIT07_33
    - HIT08_54



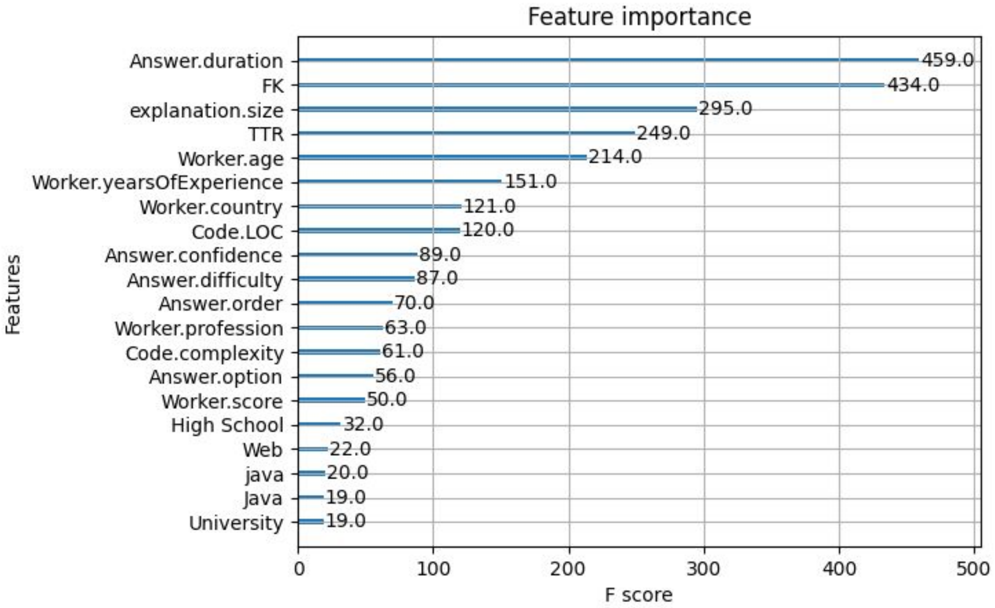
Train Model

- Chosen Methods for categorizing answers
  - Random Forest Classifier
  - **Boosted Random Forest Classifier (XGBoost)**

- Gridsearch using 5 fold CV on the training data for hyperparameter tuning
  - ```
    param_grid = {
      'n_estimators': [150, 200, 500, 1000],
      'max_depth': [2,3,4,5],
      'subsample': [0.7],
      'colsample_bytree': [0.5, 0.7, 0.9],
      'gamma': [0,0.4,0.9],}
    ```
  - Best Parameters:
    - colsample_bytree: 0.7
    - gamma: 0
    - max_depth: 3,
    - n_estimators: 500,
    - subsample: 0.9

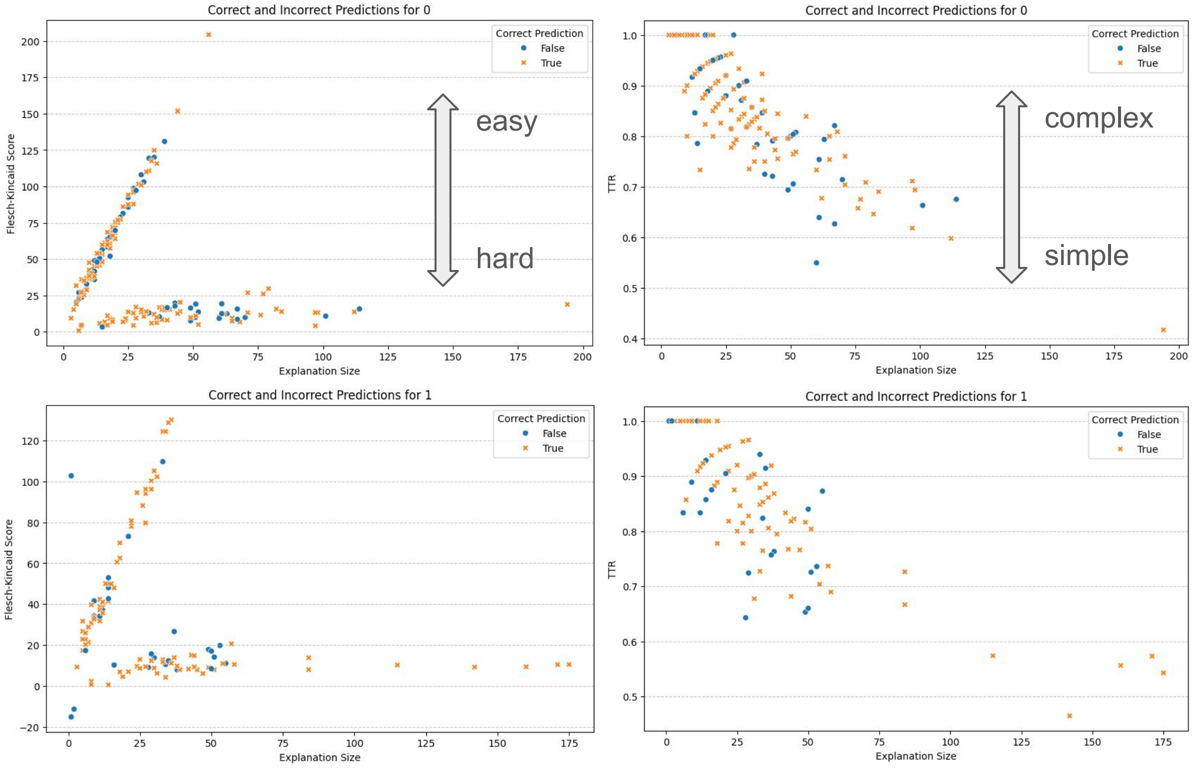
Bug Report	Samples	Train Precision	Train Recall	Train Accuracy
2	340	0.9524	1.0	0.9882
3	740	0.8850	1.0	0.9824
4	180	0.8000	1.0	0.9167
5	360	0.8696	1.0	0.9750
6	160	0.9524	1.0	0.9813
7	480	0.8333	1.0	0.9750
    ```

Model evaluation



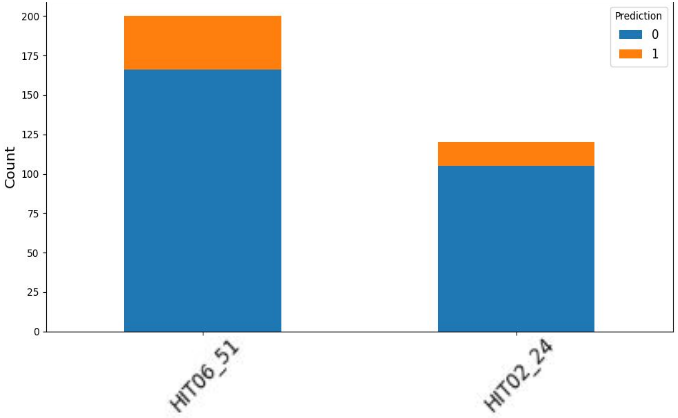
Metric	Bug Report 1	Bug Report 2
Test Precision	0.2400	0.8571
Test Recall	0.1500	0.3000
Test Accuracy	0.7350	0.7500
True Positives	6	12
False Positives	19	2
False Negatives	34	28
True Negatives	141	78

Analyze Prediction Results

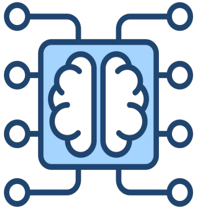


Consolidated explanations

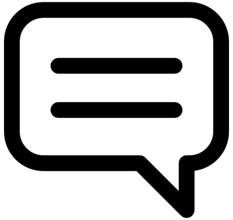
Holdout set predictions



LLM:
- GPT-4
- LLama 3.1. 405B



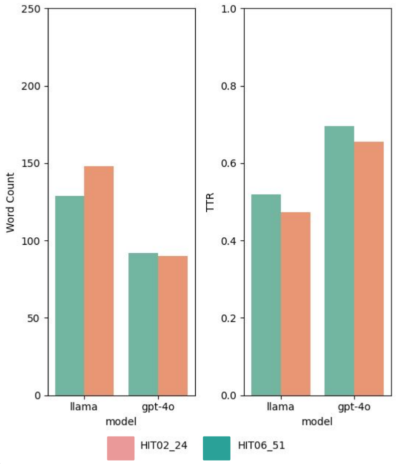
Consolidated explanation



Vanilla

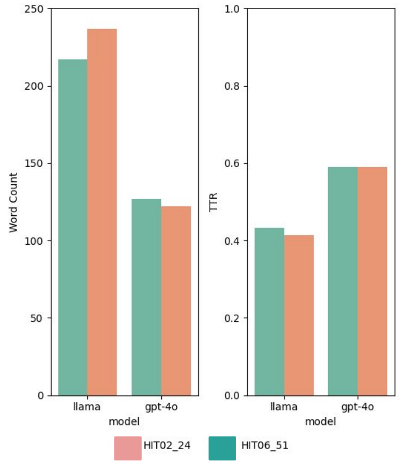
Summarize

<EXPLANATIONS>



Generated Prompt with GPT-4o

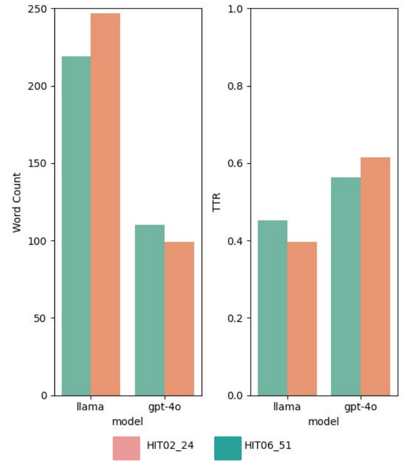
You are a highly skilled AI assistant designed to analyze and conso- lidate explanations for software bugs. Your task is to minimize …



Generated CoT Prompt with GPT-4o

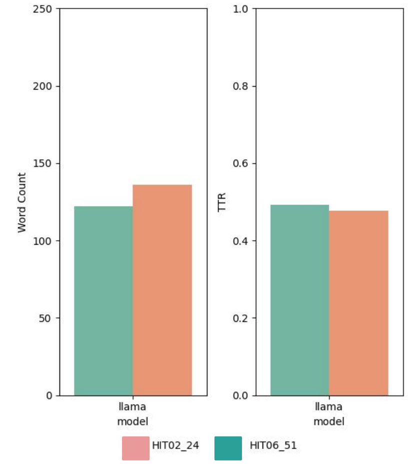
… Step-by-step reasoning to consolidate the explanation:

1. Analyze each expla- nation to identify over- lapping information …



CoT with DSPy

dspy.ChainOfThought



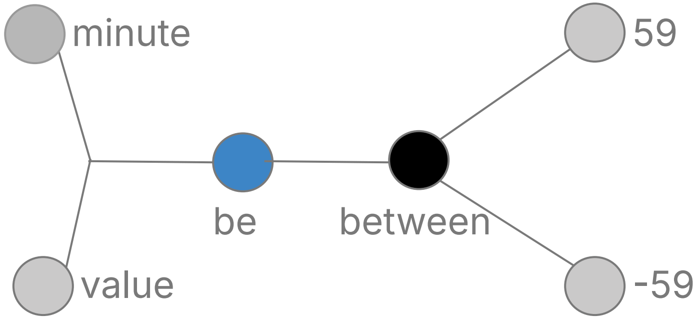
Compare Explanations - Understanding the Metrics

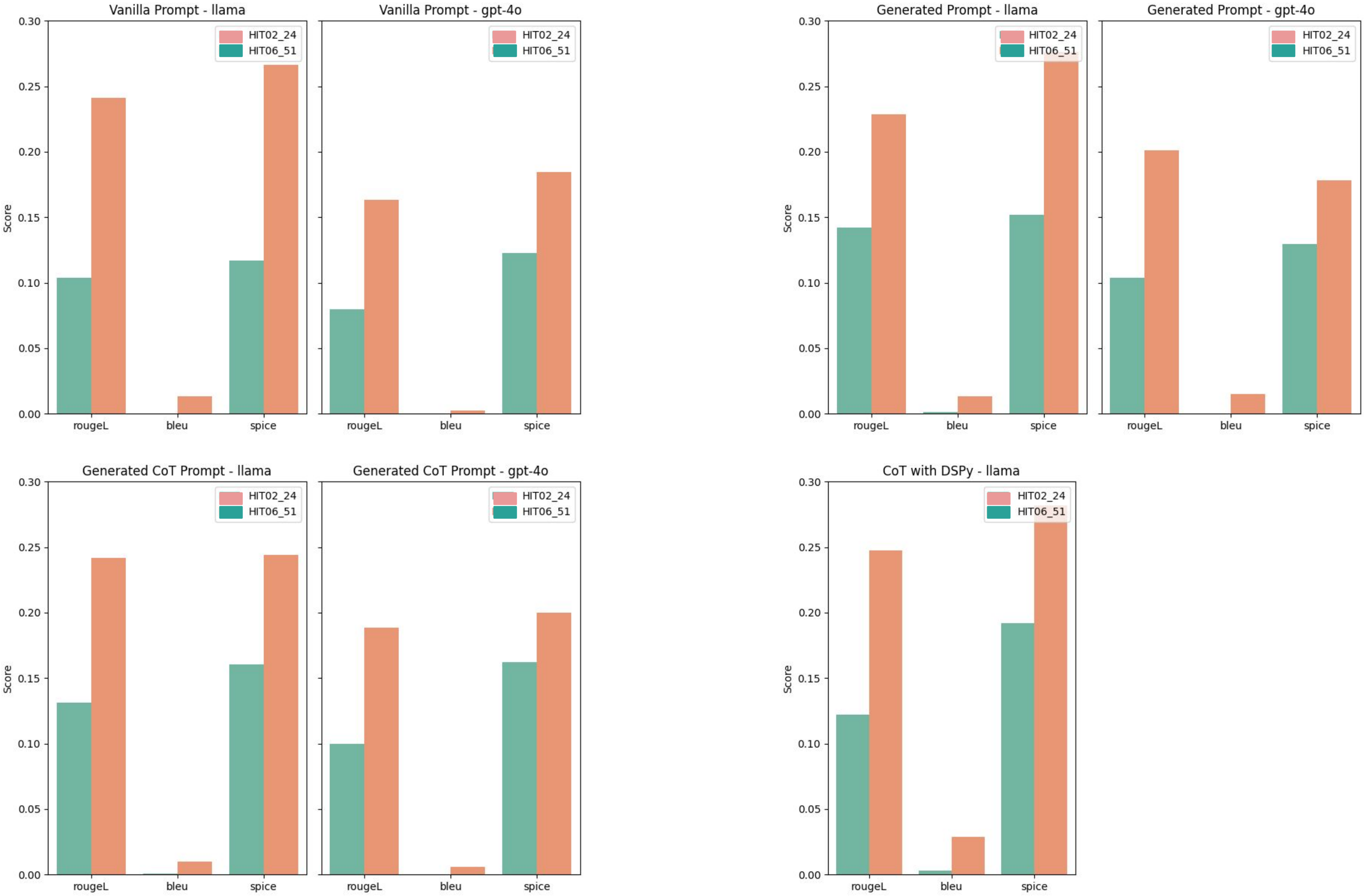
Bleu, Rouge: N-Gram overlap

The check should be updated to allow values between 59 and -59

The minute value should be between -59 and +59

SPICe: Identify relations between objects and their attributes (tuple overlap)




Vanilla Prompt - llama
Vanilla Prompt - gpt-4o
Generated Prompt - llama
Generated Prompt - gpt-4o
Generated CoT Prompt - llama
Generated CoT Prompt - gpt-4o
CoT with DSPy - llama
HIT02_24
HIT06_51
Score
0.00
0.05
0.10
0.15
0.20
0.25
0.30
rougeL
bleu
spice

Compare Explanations - Findings

reference captions are the joined positive predicted explanations for each bug rep.

- ChatGPT generates normally shorter and but more complex answers than LLAMA
- explanation size of CoT with Dspy smaller than with the generated CoT and less complex
- no huge difference for gpt-4o between generated prompt and generated CoT prompt
- BLEU penalizes the low precision (main aspects are repeated (higher recall), but maybe also other/new aspect)
- best values with the CoT prompt of the dspy module
- handles better the HIT06_51 report (more tp reports in the explanation)
    - HIT02_24 has low recall and

Hidden Slide

Reflection

What are the concerns about:

1. guaranteeing the quality of the data:

- Bias and Representativeness (e.g. gender, origin)
- Relevance of specific features (poor text-based features instead of standardized categories)

2. keeping the classifier up-to-date in the case of changes in the demographic of programmers or types of bugs

- if there are large changes it would be necessary to train a new classifier on an updated dataset
- especially when the changes relate to features that are important in the classification

3. testing the output of the classifier and the LLM

- output of the classifier can be tested more easily due to ground truth labels
- test dataset should be large enough

Reflection

4. estimating the quality of the consolidated explanations

- no summarization ground truth to compare output to
- metrics do not capture nuanced aspects of text quality(e.g. removing irrelavant parts is always penalized)

5. debugging the integration between the classifier and the LLM

- automated with Llama
- Errors in the classifier’s predictions cascade into the LLM