

Belana Zwadsich, Ole Schmitt

Mini Project 2:

Consolidate Bug Report Explanations

Advanced Software Engineering
WiSe 24-25



Training the Model: Prepare Data

- adding text complexity column
- chose TTR (type-token-ratio) as metric for diversity of vocabulary
 - low value = tendency to use same words
 - easy to comprehend & implement
 - text length influences score

```
def calculate_ttr(text):  
    # split on non-alpha. chars, lower case  
    tokens = tokenize(text)  
    # unique tokens  
    types = set(tokens)  
    return len(types) / len(tokens) if len(tokens) > 0 else 0
```

Training the Model: Precision & Recall

Bug Report	Precision	Recall
Overall	0.2286	0.0381
HIT03_6	0.44	0.09
HIT04_7	0.08	0.02
HIT05_35	0.25	0.03
HIT06_51	0.11	0.02
HIT07_33	0.60	0.05
HIT08_54	0.14	0.02

Cross-Validation Metrics

- used Random Forest Classifier
- not so great results
- small amount of data
- ground truth heavily skewed (training set - 0: 1840, 1: 420)

Dataset	Precision	Recall
Holdout	0.3333	0.025
HIT01_8	0.25	0.03
HIT02_24	0.50	0.03

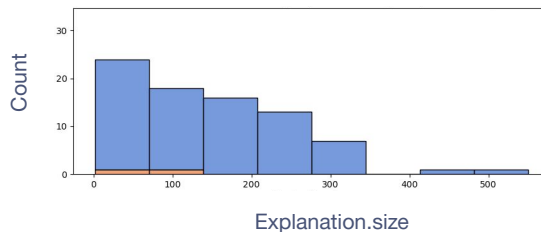
Holdout Metrics

Training the Model: Categorizing Answers

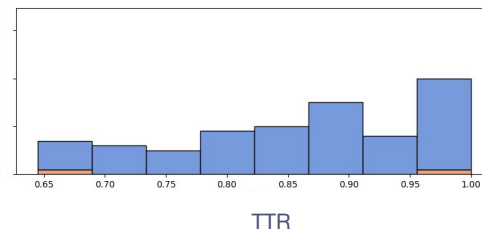
- detection performance on holdout set
 - classifier biased towards predicting 0
 - label 1 predicted in only 6 of 320 cases
- balance / augment dataset

- most explanations short
- TTR high

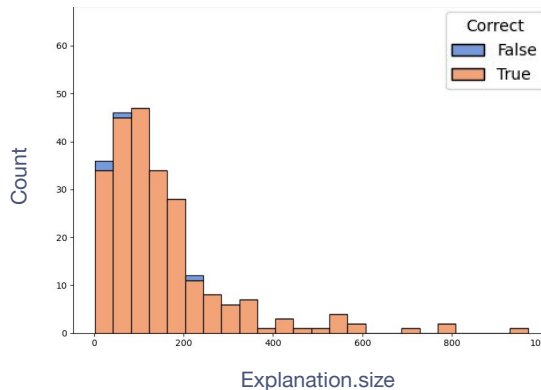
Correct Labels by Explanation Size (GT=1)



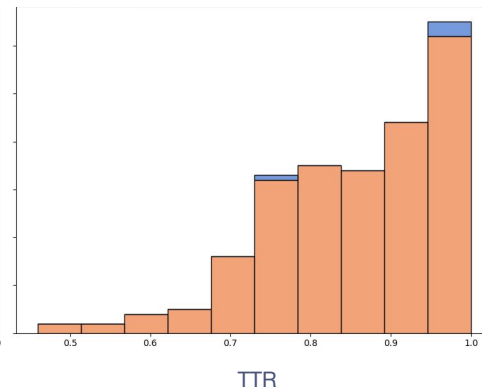
Correct Labels by TTR (GT=1)



Correct Labels by Explanation Size (GT=0)

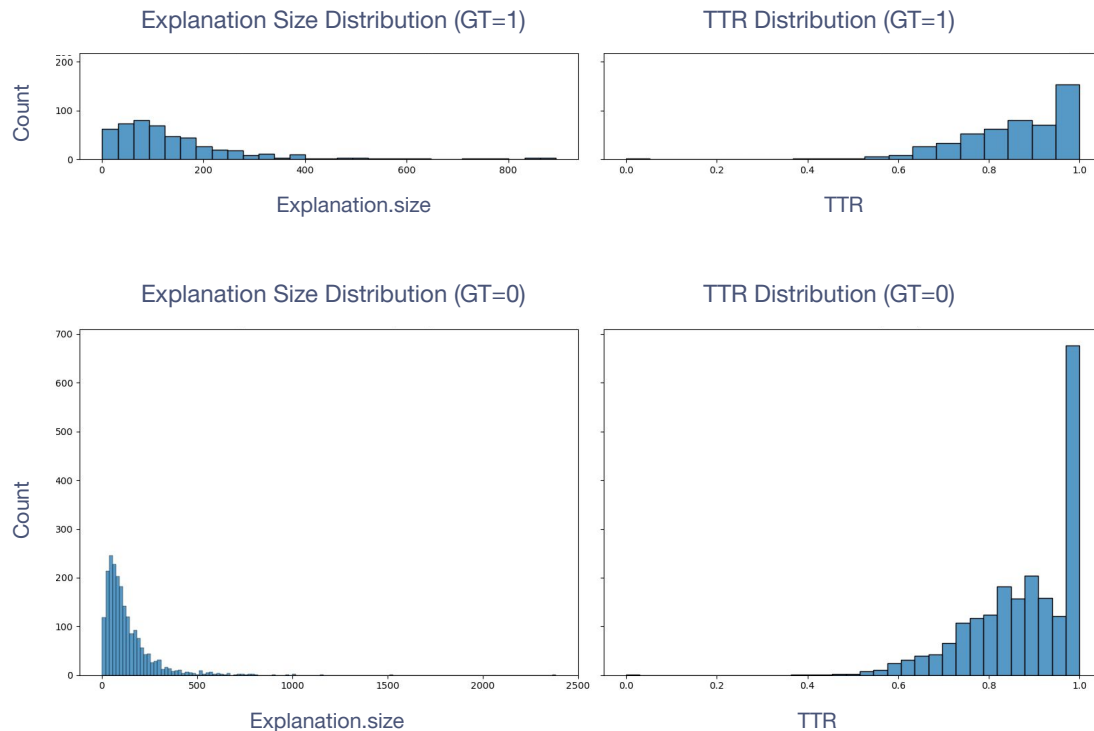


Correct Labels by TTR (GT=0)



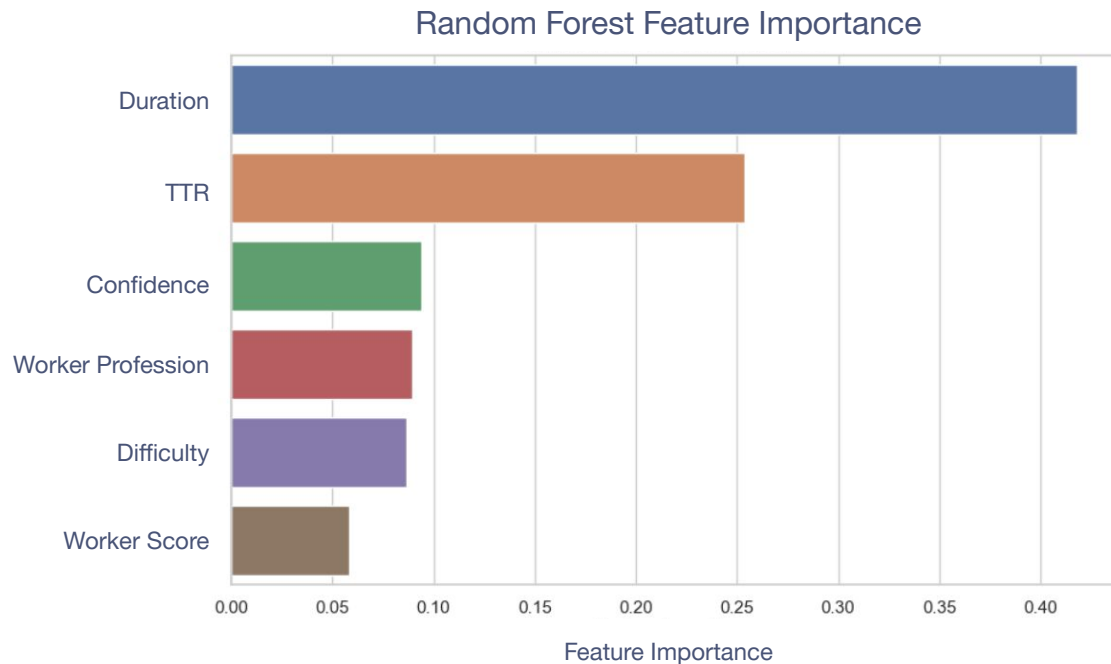
Training the Model: Categorizing Answers

- detection performance on holdout set
 - classifier biased towards predicting 0
 - label 1 predicted in only 6 of 320 cases
- balance / augment dataset
- most explanations short
 - TTR high



Training the Model: Categorizing Answers

- detection performance on holdout set
 - classifier biased towards predicting 0
 - label 1 predicted in only 6 of 320 cases
- balance / augment dataset
- most explanations short
 - TTR high
 - model relied on TTR as 2nd most important feature
- more data for positive examples



Consolidating Explanations

Compared 5 Prompts for consolidating explanations for HIT02_24 using ChatGPT4o

	Chain of Density [Adams23]	Our prompt	Summarize	Professional Summarizer [Reddit]	Task copied
Word Count	63	348	299	121	511
TTR	0.838	0.277	0.377	0.717	0.227
Readability	-	0	0	++	+
Speed*	-	0	0	++	0
Suggestion	No	Yes	Yes	No	Yes

Adams, Griffin & Fabbri, Alexander & Ladhak, Faisal & Lehman, Eric & Elhadad, Noémie. (2023). From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting. 10.48550/arXiv.2309.04269.

Comparing Explanations: BLEU

- used Chain of Density (CoD) method for summary
- BLEU scores low

→ designed for precise matching & lexical overlap

→ explanation length, structure, word choice varies

→ penalizes shorter summaries (fewer n-gram matches)

→ too much density using CoD?

→ doesn't account for semantic similarity

Issue	BLEU Score
HIT01_8	0.15
HIT02_24	0.00
HIT03_6	0.00
HIT04_7	0.11
HIT05_35	0.00
HIT06_51	0.00
HIT07_33	0.00
HIT08_54	0.00

BLEU Scores per Issue

Comparing Explanations: ROUGE

- used Chain of Density (CoD) method for summary
- ROUGE scores similarly low (for full report see Jupyter Notebook)

→ likely similar reasons

- low scores do not necessarily mean poor summary

Issue	ROUGE Score
HIT01_8	0.13
HIT02_24	0.11
HIT03_6	0.09
HIT04_7	0.07
HIT05_35	0.08
HIT06_51	0.06
HIT07_33	0.11
HIT08_54	0.09

ROUGE Scores per Issue

Reflection

- **Data quality:** crowd-sourced, diverse, uncleaned
- **Classifier update:** results may worsen over time (concept drift), updating / re-training strategy necessary; concept shift detection difficult
- **Output testing of classifier & LLM:** error propagation risk, manual review, monitor metrics, curate golden data set
- **Estimate consolidation quality:** BLEU / ROUGE seem limited, manual review, field study of devs using summaries in the wild
- **Classifier-LLM integration:** manual copy paste, (tested) data exchange pipeline necessary

Anhang

Consolidating Explanations: Chain of Density

You will generate increasingly concise, entity-dense summaries of the above answer explanations.

Repeat the following 2 steps 5 times.

Step 1. Identify 1-3 informative entities (";" delimited) from the article that are missing from the previously generated summary.

Step 2. Write a new, denser summary of identical length which covers every entity and detail from the previous summary plus the missing entities.

A missing entity is:

- Relevant to the main answer explanations,
- Specific yet concise (5 words or fewer),
- Novel (not in the previous summary),
- Faithful (present in the answer explanations),
- Anywhere (can be located anywhere in the answer explanations).

Guidelines:

- The first summary should be long (4-5 sentences, ~80 words) yet highly non-specific, containing little information beyond the entities marked as missing. Use overly verbose language and fillers (e.g., "these answer explanations discuss") to reach ~80 words.
- Make every word count: rewrite the previous summary to improve flow and make space for additional entities.
- Make space with fusion, compression, and removal of uninformative phrases like "the article discusses".
- The summaries should become highly dense and concise yet self-contained, i.e., easily understood without the answer explanations.
- Missing entities can appear anywhere in the new summary.
- Never drop entities from the previous summary. If space cannot be made, add fewer new entities.

Remember, use the exact same number of words for each summary.

Adapted from: Adams, Griffin & Tabrizi, Alexander & Ladhak, Taisir & Lehman, Eric & Elhadad, Noémie. (2023). From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting. 10.48550/arXiv.2309.04269.

Consolidating Explanations: Chain of Density Result

The "g" variable, scaled within a 255 range and cast to int, impacts the Color class, triggering `IllegalArgumentException` if bounds fail. `this.lowerBound` and `this.upperBound` inadequacies prevent sanitization, allowing negatives. Misusing "value" over "v" (line 117) worsens errors. Reliance on `Math.max()` and `Math.min()`, double types, and input consistency is emphasized. The `Color(float, float, float)` constructor and `RuntimeException` encapsulate broader dependency issues in parameter handling.

Consolidating Explanations: Our Prompt

I have a list of explanations of where there is a bug in a given code statement. Please merge the explanations in a way that minimizes redundant information, while keeping the information that would be necessary for someone else to fix the bug. Types of information that, if present in the explanation, should be preserved - how the program works, how the failure is happening, what is problem in the code, etc.
Be short, concise, do not use verbose language.

Consolidating Explanations: Summarize

Summarize

Consolidating Explanations: Professional Summarizer

As a professional summarizer, create a concise and comprehensive summary of the provided text, be it an article, post, conversation, or passage, while adhering to these guidelines:

1. Craft a summary that is detailed, thorough, in-depth, and complex, while maintaining clarity and conciseness.
2. Incorporate main ideas and essential information, eliminating extraneous language and focusing on critical aspects.
3. Rely strictly on the provided text, without including external information.
4. Format the summary in paragraph form for easy understanding.

Consolidating Explanations: Professional Summarizer Result

The provided text discusses a technical issue involving the use of the "g" variable in a coding context, where it is scaled within a range of 255 and cast to an integer, potentially causing problems in the Color class. Specifically, an `IllegalArgumentException` is triggered when values exceed defined bounds. The failure to properly sanitize the `this.lowerBound` and `this.upperBound` variables allows negative values, compounding the issue. Additionally, incorrect usage of "value" instead of "v" on line 117 introduces further complications. The text underscores the importance of using methods like `Math.max()` and `Math.min()` with consistent double-type inputs to ensure reliability. Broader challenges are identified with parameter handling in the `Color(float, float, float)` constructor, with dependencies further obscured by encapsulating issues in a `RuntimeException`.

Consolidating Explanations: Copied Task

generate a single explanation by merging the participants' explanations in a way that minimizes redundant information, while keeping the information that would be necessary for someone else to fix the bug.

Types of information that, if present in the explanation, should be preserved - how the program works, how the failure is happening, what is problem in the code, etc.