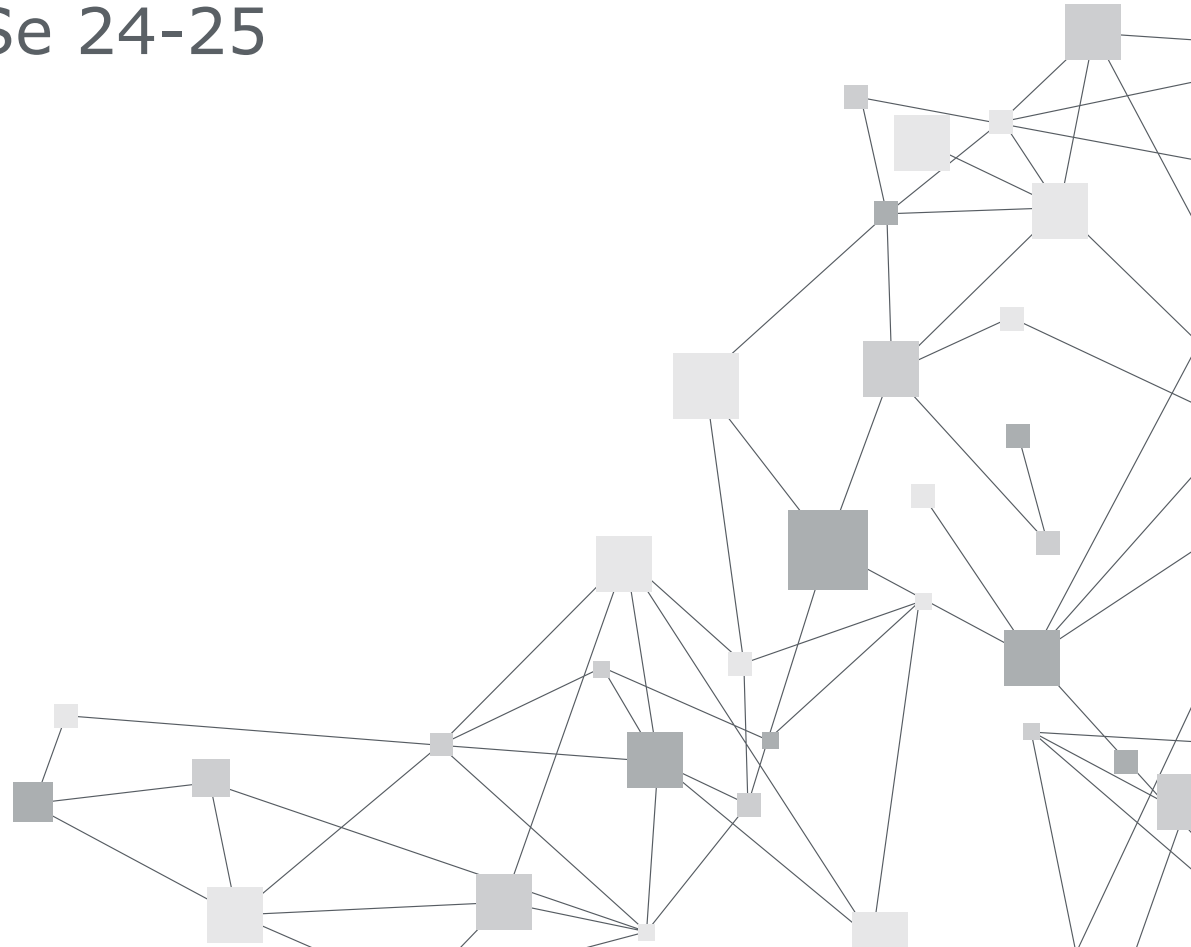
Mini-Project-2

Advanced Software Engineering WiSe 24-25

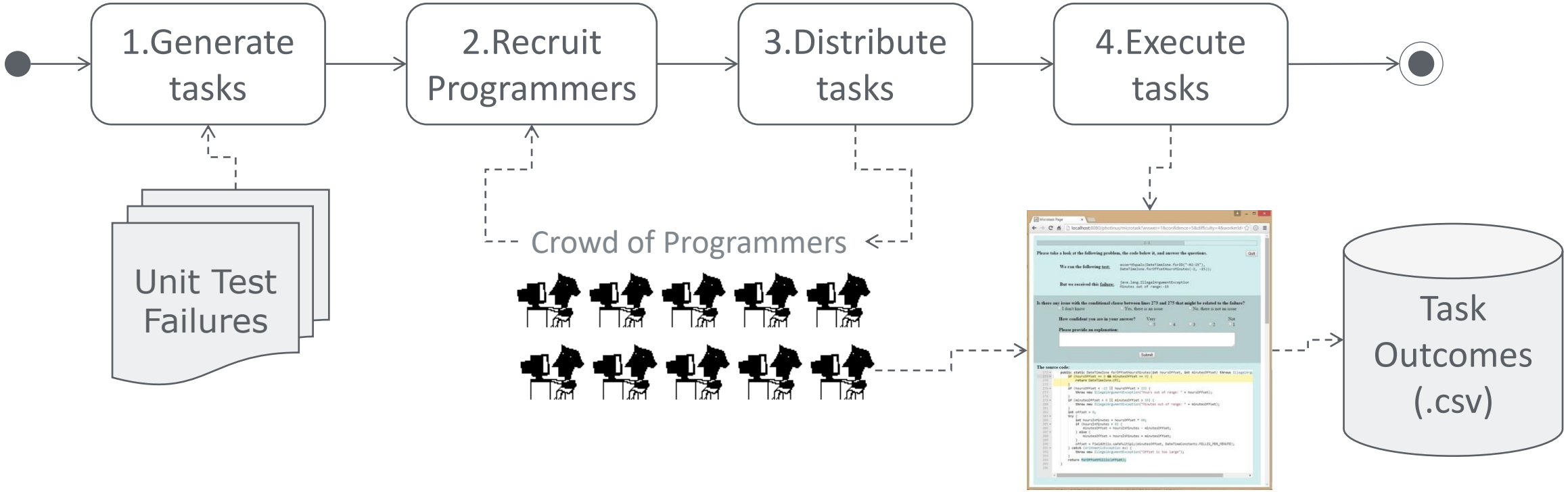
Christian M. Adriano



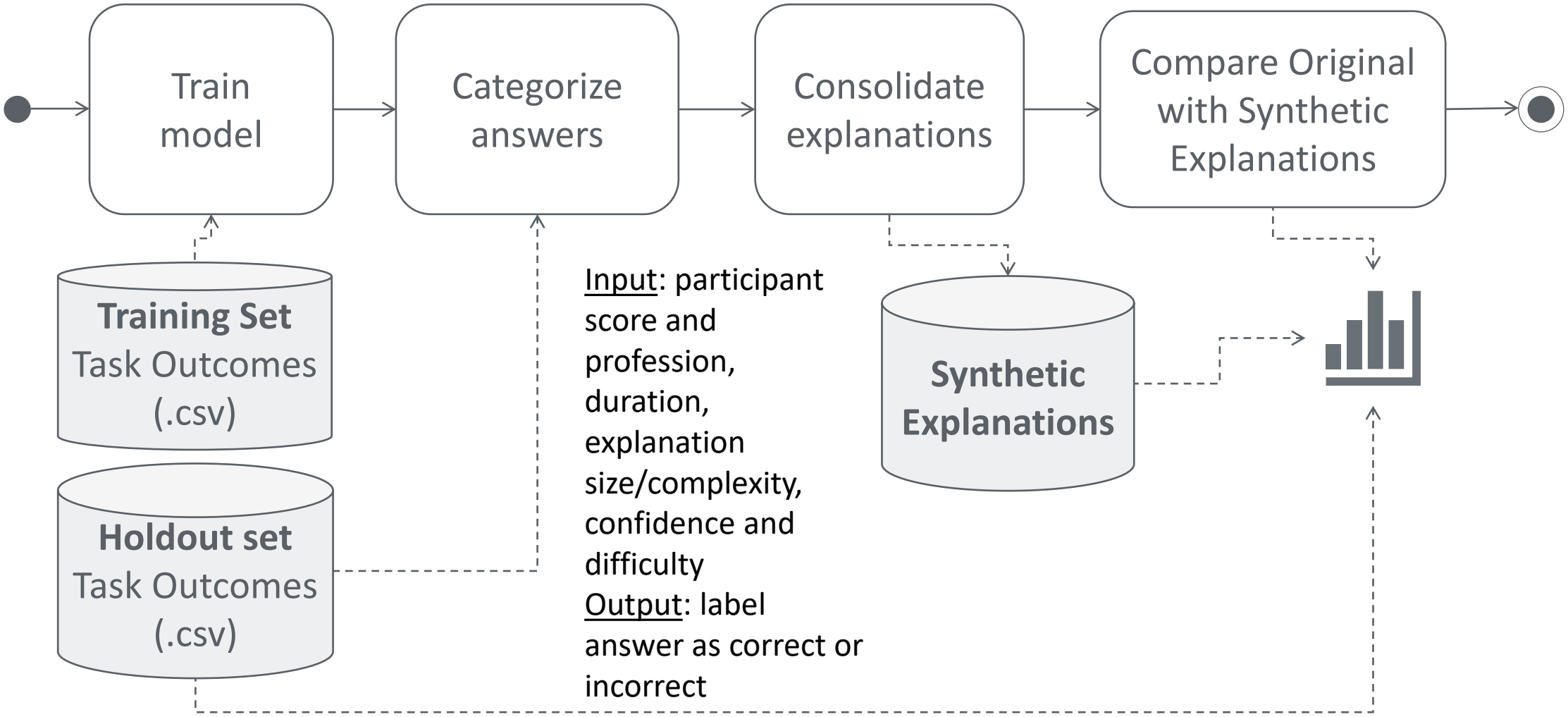
Design IT.
Create Knowledge.

www.hpi.de

How the data was generated



Consolidating bug report explanations



Specifications - I

Prepare data: Split the data set in training and holdout set. For that, choose two bug reports for holdout set, while use the remaining for training.

- For each explanation, add a column with their complexity, e.g., TTR (type-token ratio), Halstead volume, or other score of your choice. Justify it.

Train the Model: choose a decision tree-based method to classify each answer as correct or incorrect. Report per bug report the precision and recall of your classifier (use cross-validation to train and find the best hyperparameters).

- Discuss it. Anything concerning?

Categorize Answers: Use your classifier (on the holdout set!) to label each answer (row).

- For each bug report in the holdout set, report on the precision and recall.
- For the inspection tasks (rows) that host the bug, show the distribution of correct labels by explanation size and complexity.

Specifications -II

Consolidate Explanations: for the correct answers to the inspection tasks hosting the bug, prompt the LLM to 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.
- Try different ways of prompting and report on how the size and complexity of explanations changed.

Compare Explanations: for each LLM generated explanation, compare it with the original ones.

- Assume that the original explanations are the reference, then compute BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation). Discuss your finding.

Reflection

What are the concerns about:

1. guaranteeing the quality of the data
2. keeping the classifier up-to-date in the case of changes in the demographic of programmers or types of bugs
3. testing the output of the classifier and the LLM
4. estimating the quality of the consolidated explanations
5. debugging the integration between the classifier and the LLM

End

Datasets and further instructions will be available on the GitHub repo.