Automated Answer Correctness Evaluation

Assessing student answers for correctness in computer science (data structure/algorithms courses)

By: Orel Cohen, Nathan Beer, Tomer Bengayev

Motivating Use Case

The Problem: Manual Grading Burdens

Traditional manual grading assignments is **time-consuming** and **inconsistent**.

Why It Matters: Fair & Efficient Feedback

Ensuring and providing **timely feedback** are crucial for student learning , assistance for practitioners

Challenges

Evaluating answers correctness goes beyond simple keyword matching, requiring **deep understanding** of logic, algorithms, and syntax.

Today

Manual Grading- traditional way which a human examiner compares each answer to the correct answer and determines a score.

Project Task Description

Input

Question, student's answer, correct answer (Text)

Output

Correctness score {0-1}

Our Innovation

Implementation of Triple Context using an End-to-End Transformer architecture (such as BERT), while avoiding the use of manual features and outdated similarity measures, using synthetic data improving diversity and generalization beyond the original MohlerASAG dataset.

Processing Pipeline: Models and Methods

01

Data Preprocessing

MohlerASAG dataset (Hugging Face), input arrangement (scailing, feature extraction), Adding synthetic data using LLM (GPT-4)

02

Modeling & Tokenization

Tokenization, fine-tuned transformer model (BERT/RoBERTa), connecting a single linear layer (Regression Head)

03

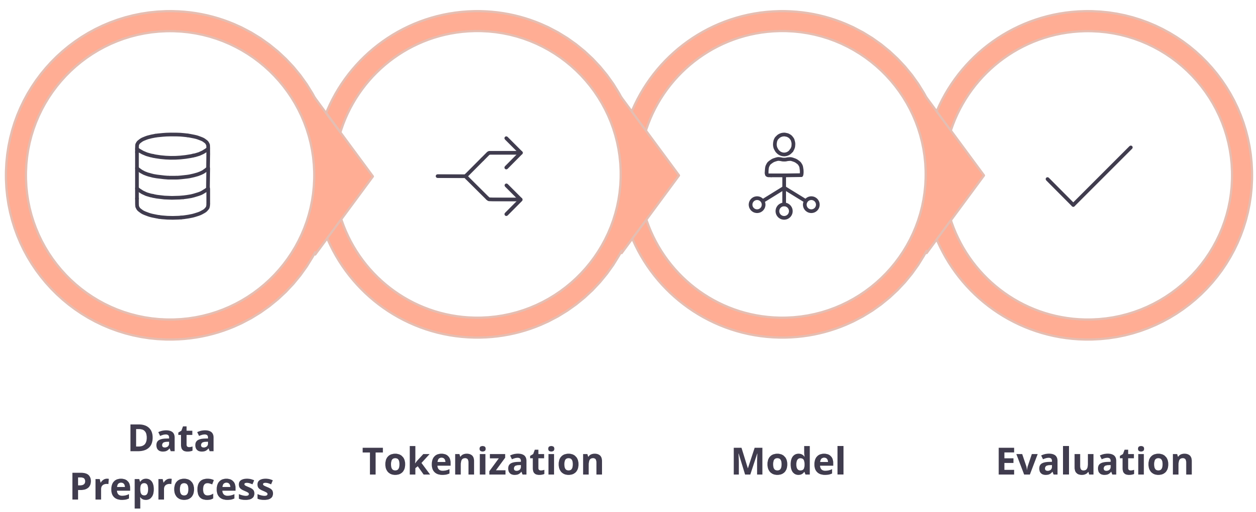
Model Training & Optimization

Training the model with loss function definition (MSE), performing Fine-Tuning

04

Final Evaluation & Output

Normalization with min-max normalization, performance measurement (RMSE), output score {0-1}



Data specification & Generation

Training & Evaluation Data

- Thousands of examples for transforming model (BERT/RoBERTa)
- Each example must consist of 3 input components (question, student answer, *correct answer*)
- **Evaluation data:** Test set consists only from dataset "MohlerASAG" (Huggingface)

Synthetic Data Generation

- Input to LLM: We feed a LLM (GPT-4) the pair (question, correct answer)
- *LLM generates a new answer (student answer)* that has a specific semantic similarity to the correct answer.
- **Label Assignment:** The label (score) is determined by the generation prompt (e.g., prompt for 'High Misconception' = Label 0).
- Using LLM evaluator to check synthetic quality, Filters: Bad logic / Unrelated answers / Slang, manual validation on 20–30 examples

Measuring Success: Metrics and KPIs

1

Metrics

- RMSE
- Pearson Correlation

2

Ground Truth Data

The true score in the "**MohlerASAG"** dataset, after being normalized to a range of 0 to 1.

3

Measurement Protocol

Training: Fine-Tuning of BERT While tracking MSE loss

Comparison: Calculating the metrics (RMSE) by comparing the ground truth scores.

Baseline Comparison:

- Sentence-BERT cosine similarity
- GPT-4 zero-shot scoring

4

Overall Quality & Step Measurement

Training phase: Measuring the Validation Loss (MSE) at each epoch.

Regression phase: Testing the model's bias – by analyzing the error matrix.