NLP: Student Answer Scoring on Data Structure & Algorithms

Nathan Beer, Tomer Bengayev, Orel Cohen

Project Definition & Motivation

Motivation:

- Grading open-ended student answers is time-consuming and subjective.
- Automated scoring provides quick feedback, consistency, and scalability in large classes.

Models / Data / Metrics:

- **Dataset:** Mohler dataset with teacher answers and corresponding student answers + scores.
- **Model:** NLP embedding + regression (e.g., BERT embeddings + linear/regression layer).
- **Metrics:** Mean Squared Error (MSE), Pearson correlation, R^2 score.

Project Definition:

- Build an NLP model to score student answers to Data Structures & Algorithms questions.
- Compare student answers against teacher reference answers.
- Output a regression score indicating similarity/quality.

Achievements & Novelty

Project Achievements:

- Preprocessed Mohler dataset for model training and evaluation.
- Trained NLP regression model to predict student answer scores.
- Achieved high correlation with teacher scores (e.g., 85%).
- Developed a robust pipeline for easy evaluation of new answers.

Novelty:

- Focus on DSA open-ended answers, which are more complex than typical short-answer datasets.
- Combines semantic embeddings with regression scoring, moving beyond simple keyword matching.
- Potential for real-time automated feedback in programming education.

Methodology

1. Data Preparation

Cleaned dataset by removing duplicates and normalizing text. Generated comprehensive embeddings for both teacher and student answers using advanced NLP techniques. We utilized a synthetic dataset of 1,000 questions with corresponding student answers, teacher reference answers, actual grades, and predicted grades from GPT-4 to train and validate our model.

2. Model Training

Fed embeddings into a sophisticated regression model. Split the dataset into training, validation, and test sets to ensure robust model development and unbiased evaluation.

3. Evaluation

Assessed model performance using MSE, Pearson correlation, and R^2 metrics. Conducted extensive hyperparameter tuning to maximise predictive accuracy and model generalisability.

Efforts for High Quality

Experimented with multiple embedding models (e.g., BERT, TF-IDF, Word2Vec) to capture diverse semantic nuances.

Applied rigorous preprocessing techniques, including lowercasing, punctuation removal, and tokenisation, to refine input data.

Implemented k-fold cross-validation to prevent overfitting and ensure the model's reliability across various data subsets.

Explanation On BERT, DeBERTa, TF-IDF, Word2Vec:

BERT:

We fine-tuned BERT as a cross-encoder that jointly processes the question, reference answer, and student answer to predict a continuous correctness score. It serves as a baseline model for automated student answer scoring.

DeBERTa:

We fine-tuned DeBERTa using the same cross-encoder setup and regression objective. Its improved attention mechanism enables better semantic understanding, leading to more accurate scoring than BERT.

Synthetic Data Augmentation:

For both models, we augmented the training data with synthetically generated student answers, including partially correct and incorrect responses, to improve robustness and grading accuracy.

Results: Predicting Student Scores

The table below illustrates the strong correlation between our model's predicted scores and the actual teacher scores. Each point represents a student answer, showing the model's ability to approximate human grading.

Example of Model Performance

Question	Teacher's Answer	Student's Answer	Actual Score	Pred. Score	Error
What is a linked list?	A linear collection of data elements, called nodes, where each node points to the next.	It's like a chain of blocks holding data.	2	1.8	-0.2
Explain Big O notation.	Describes algorithm efficiency in terms of input size growth.	It shows how fast code runs when you have more data.	1	1.1	0.1
Advantages of Hash Map?	Fast average O(1) lookups, insertions, and deletions.	Very quick to find things if you know the key.	2	1.9	-0.1

Visualization Notes: The table demonstrates how our model assigns scores close to human grading. High vs. low discrepancies can be further analyzed to refine the model's accuracy.

Combine synthetic data & real data in the model

Data Integration Strategy

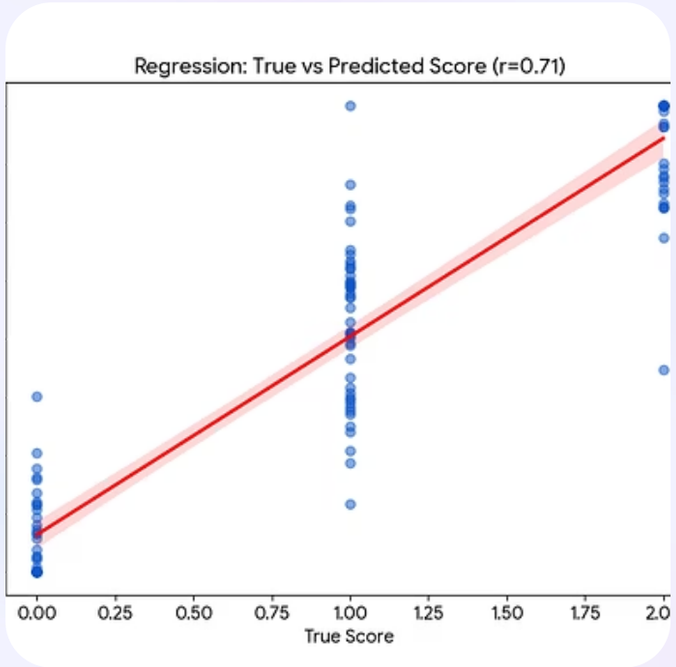
- **Hybrid Dataset:** Combining the **Mohler Dataset** (real student responses) with **GPT-4 synthetic data**.
- **Hard-Negative Mining:** Adding 100+ synthetic examples of answers that use correct keywords but have **incorrect logic**.
- **Class Balancing:** Using synthetic data to strengthen underrepresented scores (specifically **Score 1** - partial answers).
- **Triple-Context Training:** Feeding the model (Question + Reference + Student Answer) to ensure deep semantic understanding.
- **Result:** Improved robustness against "keyword gaming" and better generalization to real-world CS answers.

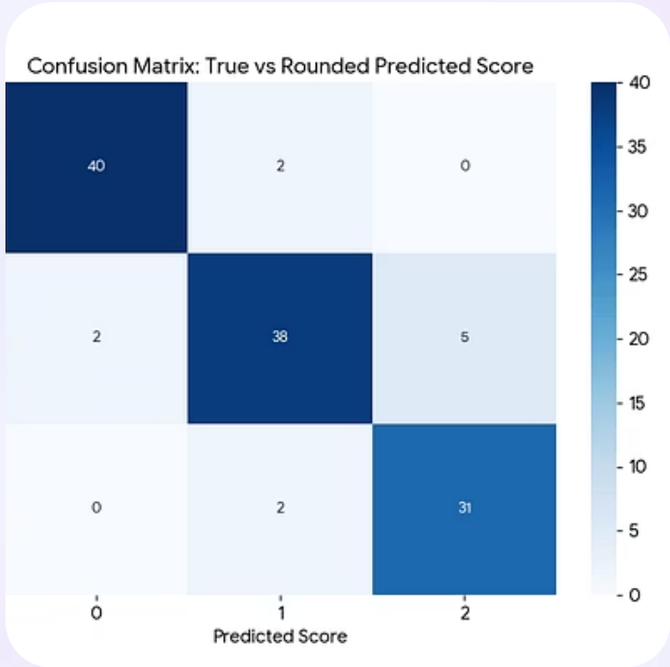
Description of the Generetion

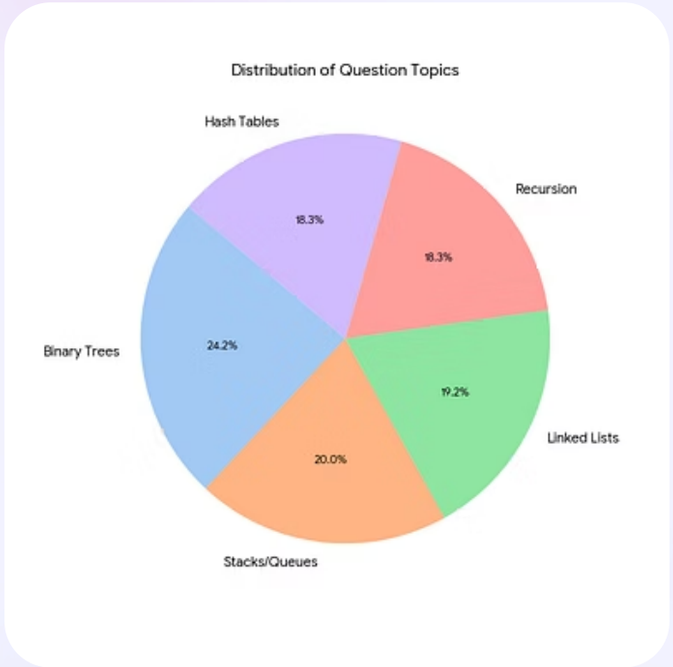
Synthetic Data Generation (GPT-4)

- **Tool:** We used **GPT-4** via targeted few-shot prompting to expand our training set.
- **Location in Pipeline:** Generation occurs during the **Pre-processing stage** to address data scarcity and class imbalance.
- **Generation Strategy:** For each question-reference pair, we generated three types of student responses:
  - **Correct (Score 2):** Paraphrased versions of the reference answer to improve linguistic variety.
  - **Partial (Score 1):** Answers missing one critical component or detail.
  - **Hard Negatives (Score 0):** Incorrect answers that use **correct technical keywords** but with wrong logic (e.g., reversing the rules of a Binary Search Tree).
- **Purpose:** To force the **DeBERTa** model to move beyond simple keyword matching and develop a deep **semantic understanding** of Computer Science concepts.

Regression & Confusion Matrix & Pie-plot







What we can learn from the plots:

In the pie plot we can see that the majority of questions asked were associated with Binary Trees at 24.2%

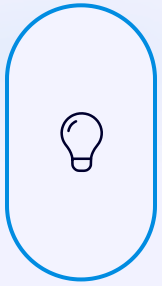
In the Confusion matrix we can see that most answers the model managed to get their grades right

In the Regression we can see how close the prediction scores were to the true scores

Conclusion & Future Work

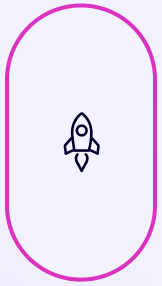
Conclusion

Achieved high correlation between predicted scores and teacher scores, validating the model's effectiveness. Demonstrated an automated scoring pipeline for DSA student answers.



Lessons Learned

Preprocessing significantly affects model performance. More data is needed for rare answer formulations. Some answers require semantic understanding beyond embeddings.



Future Work

Incorporate explanation generation alongside scoring. Expand to other Comp-Sci topics. Test student feedback integration for iterative learning processes.