



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

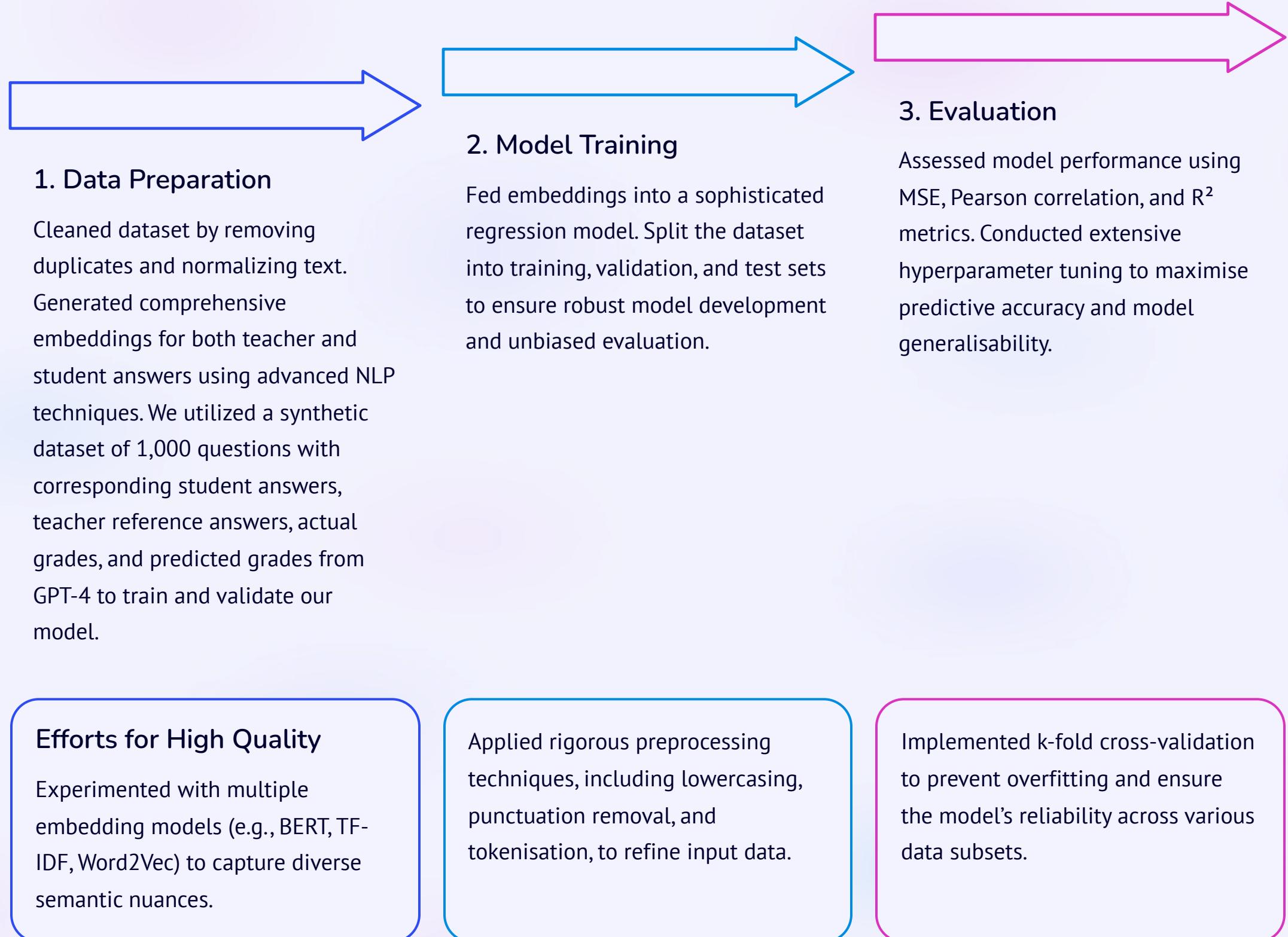
Project Achievements:

- 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



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

TF-IDF: a statistical method that represents text by measuring how important a word is in a document compared to the whole dataset. It does not understand meaning or context and ignores word order.

Word2Vec: a neural model that learns vector representations of words based on their surrounding words. It captures semantic relationships between words, but each word has a single fixed meaning regardless of context.

BERT: a transformer-based model that understands language using context from both the left and right sides of a word. This allows the same word to have different meanings in different sentences, giving much deeper language understanding.

DeBERTa: an improved version of BERT that separates word meaning from word position and uses a more advanced attention mechanism, resulting in better accuracy and performance on many NLP tasks.

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

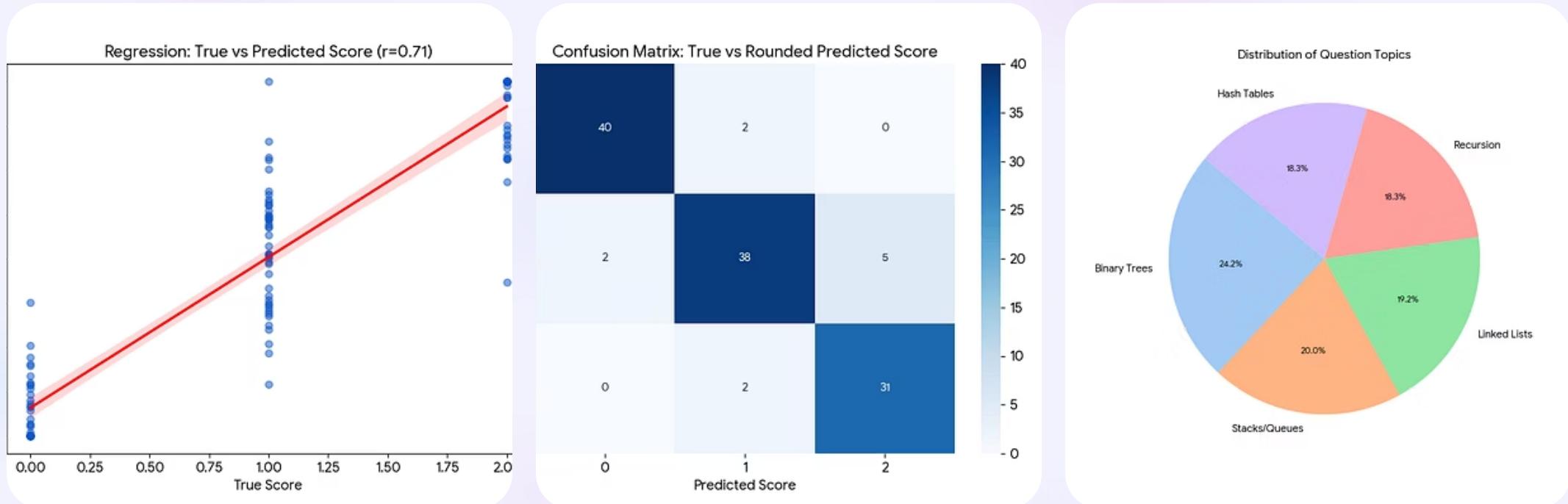
- **Hybrid Dataset:** Combining the **Mohler Dataset** (real student responses) with **GPT-4 synthetic data**.
- **Hard-Negative Mining:** Adding 100+ synthetic examples of answers that use correct keywords but have **incorrect logic**.
- **Class Balancing:** Using synthetic data to strengthen underrepresented 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 Generation

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



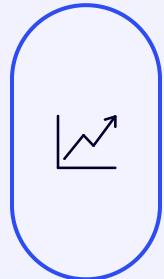
What we can learn from the plots:

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

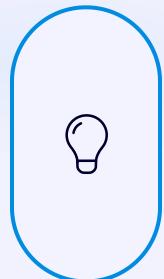
In the Regression we can see how close the prediction scores were to the true scores

Conclusion & Future Work



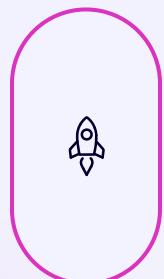
Conclusion

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.