NEXUS.AI

AI-BASED ANSWER SHEET EVALUATION SYSTEM

Introduction

Nexus is an AI-based Answer Sheet Evaluation System. It automates and accelerates the grading process using advanced Natural Language Processing (NLP) Techniques, offering fast and accurate results.

NEXT

INTRODUCTION

Problem Statement

70%

of examiners find manual correction of subjective answers to be tedious and tiresome.

60%

of examiners experience fatigue and reduced attentiveness during manual correction.

20%

variance in marking among different examiners persists, compromising the fairness and reliability of assessments.

NEXT

Findings

20 Lakh

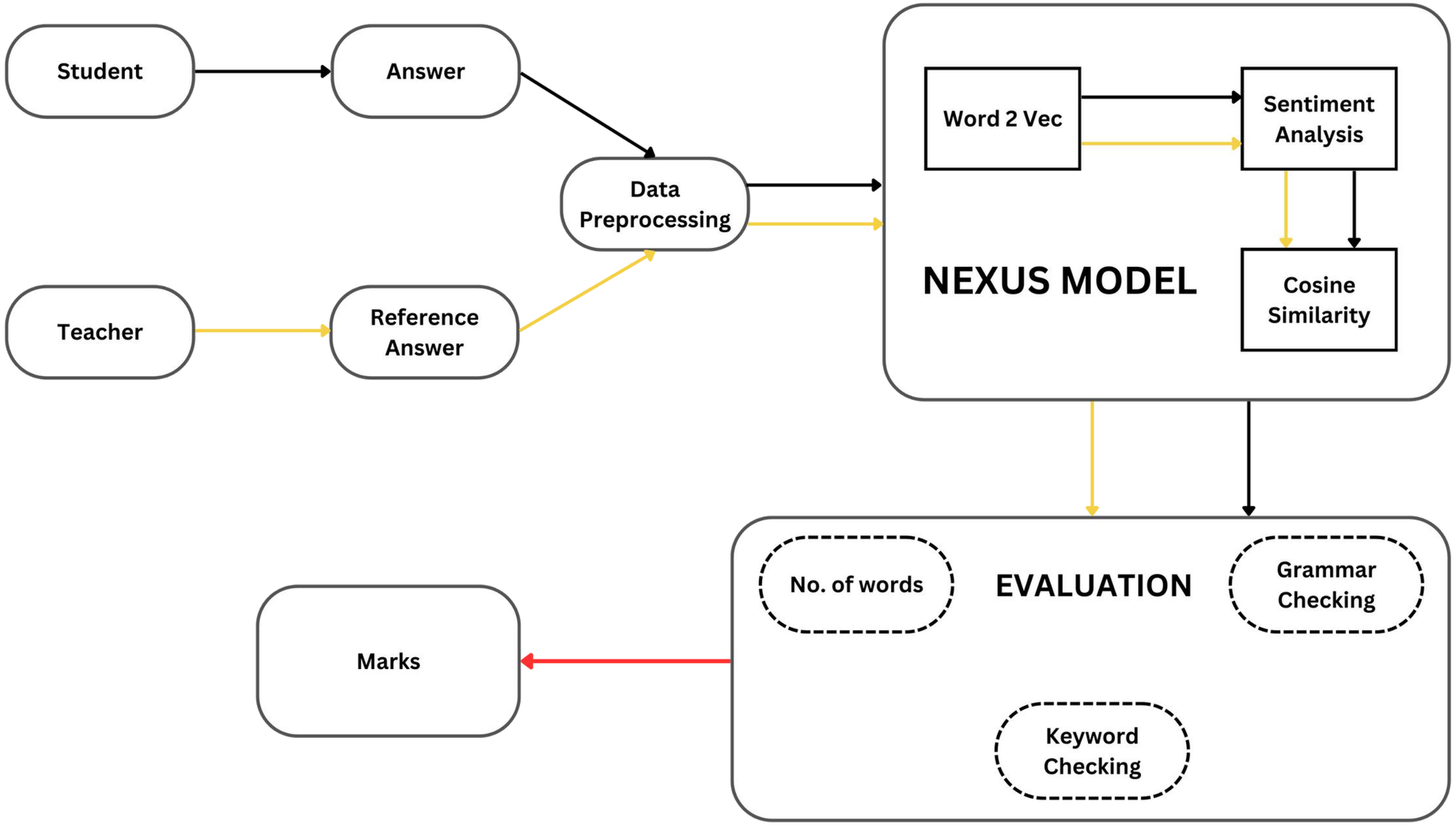
engineering students approx. participate in semester subjective written examinations

1.2 Crore

papers are corrected every semester leading to a total of 2.4 crores per year

NEXT

Architecture



Evaluation

By harnessing the power of Artificial Intelligence, Nexus revolutionizes traditional grading methodologies, offering a innovative approach that ensures accuracy, efficiency, and fairness. Let's explore how our model sets a new standard in assessment methodologies.

NEXT➤

EVALUATION

Question Classification

Purpose of Question Classification:

- Ensure accurate assessment by understanding the nature of questions.
- Address the limitations of sentiment analysis for certain question types.

Challenges Addressed:

- Analytical questions may yield different opinions on the same topic.
- Sentiment analysis alone may not suffice due to varying viewpoints.

How to Classify Questions:

- We have used zeroshot classification technique.
- Compared the output with predefined list Analytical, Factual, Inductive.

EVALUATION

Question Classification

Types of Questions

Factual Questions:

- Seek factual information.
- Recognized by question tags: who, where, when, what, or which.
- Example: "What is AI?"

Inductive Questions:

- Require evaluation and inference.
- Question tags include: why, how, justify, show how, etc.
- Example: "Why do we need AI?"

Analytical Questions:

- Involve diverse viewpoints or opinions.
- Not confined to specific question tags.
- Example: "Write your views on AI?"

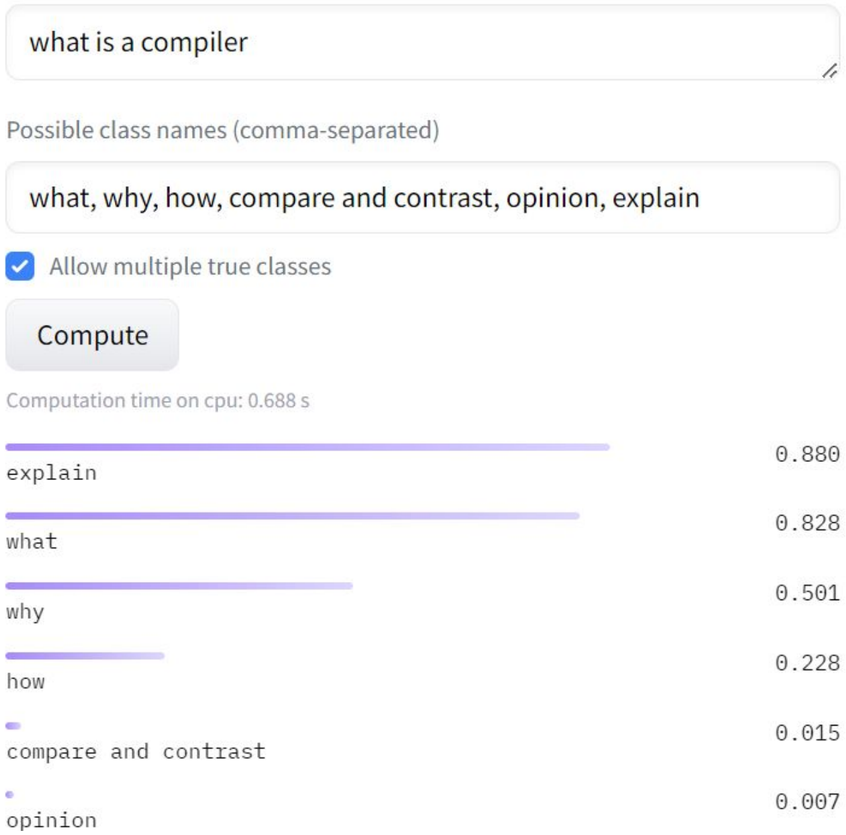
NEXT

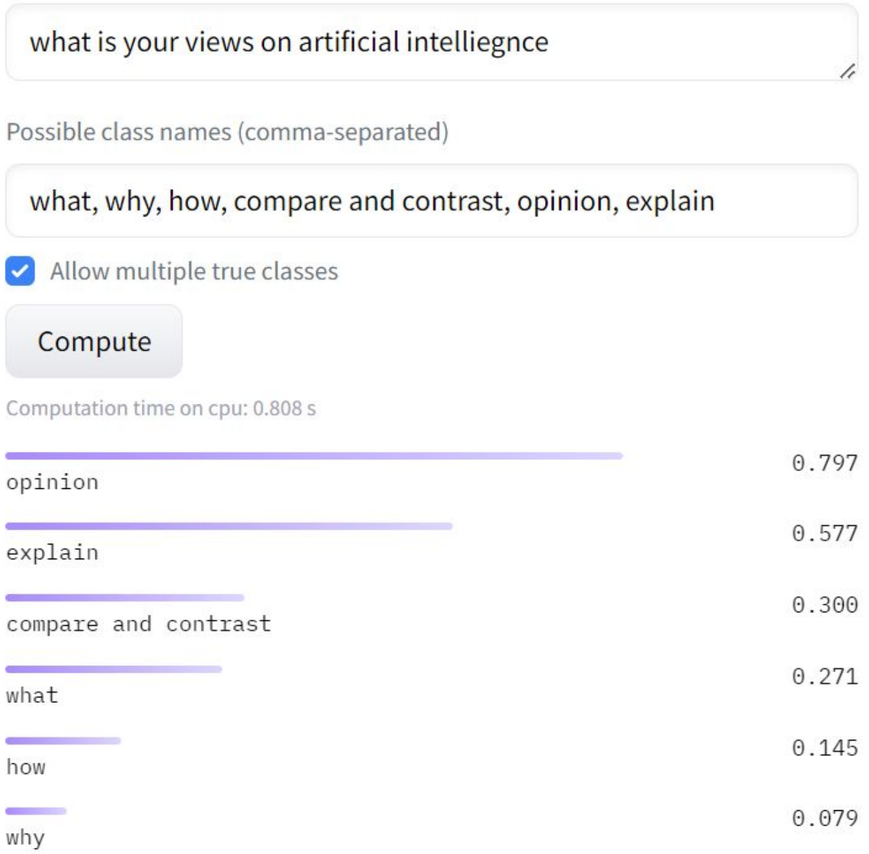
Question Classification

Zero-shot text classification is a task in natural language processing where a model is trained on a set of labeled examples but is then able to classify new examples from previously unseen classes.

EVALUATION

Question Classification





- Classes were carefully selected to ensure the most suitable question classification.
- ‘facebook/bart-large-mnli ‘ model is used for this classification.

NEXT

Question Classification

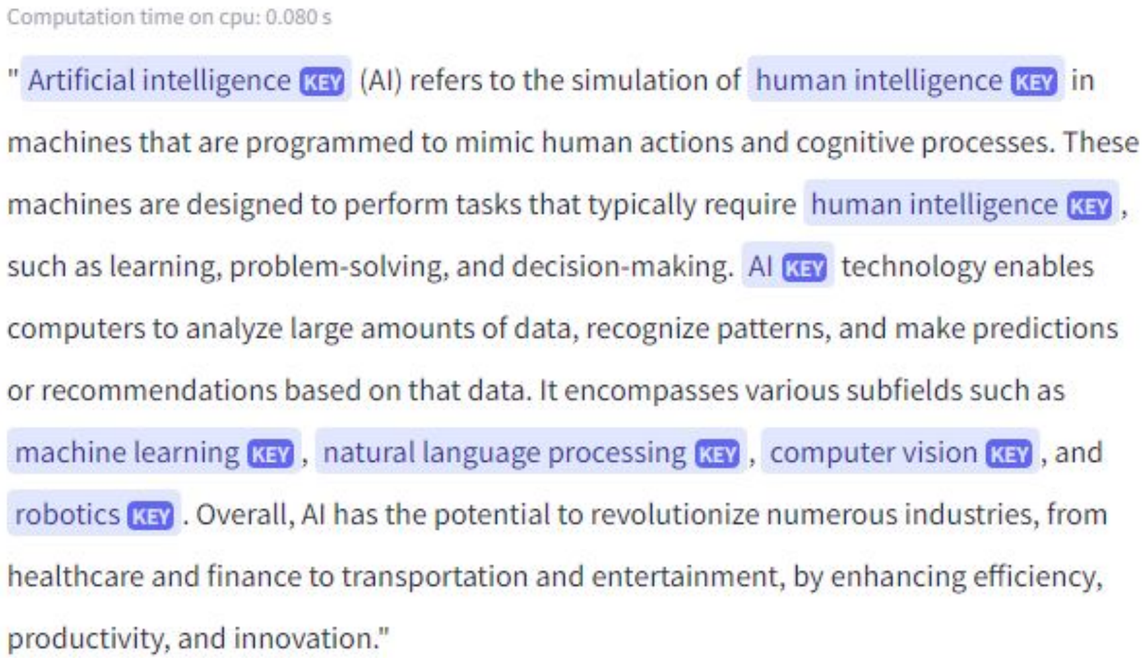
```
def question_classification(questions):
        factual=['what','explain']
        inductive=['why','how']
        Analytical=['Distinguish','compare and contrast','opinion','views','advantages','disadvantages']
```

- Class label extracted from zeroshot classification model will be used to identify a the question class.
- Different evaluation techniques will be employed based on the question class.
- For Inductive and Factual questions sentiment analysis will be used.
- Analytical questions will involve comparing student and reference answers based on extracted keywords.
- Keywords extracted from both student and reference answers will be compared using Jaccard similarity.

EVALUATION

Question Classification

- Model : bert-keyword-extractor.
- BERT is a transformers model pretrained on a large corpus of English data in a self-supervised fashion.
- Using this model we will extract keywords from the reference answer and student answer.



NEXT

Sentiment Analysis

To evaluate a answer we will use a sentiment analysis model to get the sentiment of the user

- This sentiment will be used to measure the similarities between two answers.
- Sentiment Analysis is used mainly on factual and inductive questions to check whether the student and reference ans have same sentiment.
- If both answers have same sentiment then similarity score will be positive.
- If sentiments of answers are different then similarity score will be negative.
- Sentiment analysis is not used for evaluating analytical questions.

NEXT

NLP MODELS

Word2Vec

For sentence embedding we have used a pretrained model called all-mpnet-base-v2

- Which is a sentence-transformers model It maps sentences & paragraphs to a 768 dimensional dense vector space.
- Sentence-Transformers is a Python framework sentence embeddings.
- Using this framework we can compute sentence / text embeddings for more than 100 languages. These embeddings can then be compared e.g. with cosine-similarity to find sentences with a similar meaning.
- Given an input text, it ouptuts a vector which captures the semantic information of student answer.

NEXT

NLP MODELS

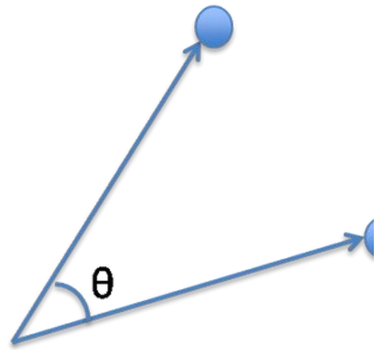
Word2Vec

- From student answer and reference answer embeddings

  we will measure similarity between the two embeddings.

- Similarity metric we will use is cosine similarity.

$$sim(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



- Cosine similarity measure the distance between two vectors.

- This similarity score will be used to get **sa_simarks.**

NEXT

EVALUATION

Answer Length Checking

Introduction:

- Incorporates analysis of answer length as a crucial component in evaluating student responses.
- Depending upon the question mark there will be expected no.of words in student answer.

Wordmark Method:

- Calculation of Wordmark considers:
  - Expected No.of words for the question (Qwords)
  - Question mark (Qmark)
  - Number of words in the student's answer (sanwords)
- Utilizes weightage ratios w1, w2, and w3 to split Qmark for computing the final weighted score (wmark).

$$w_{\text{mark}} = \frac{Q_{\text{mark}} \times w_3}{(w_1 + w_2 + w_3)}$$

NEXT

EVALUATION

Answer Length Checking

```
def wordmark(Qwords,Qmark,sa_nwords):
    wmark=Qmark * w2/(w1+w2+w3)
    if sa_nwords >= Qwords:
        return wmark
    else:
        penalty=wmark/Qwords
        wmark= penalty*sa_nwords
        return wmark
```

Penalty for Shorter Answers:

- If the student's answer is shorter than the expected length, a penalty is applied proportional to the shortfall in word count.
- Ensures appropriate penalization for incomplete coverage of the question's content.

EVALUATION

Grammar Checking

Introduction:

- Implemented as a vital component for ensuring accuracy and coherence of student responses.
- We used online languagetool to get grammatical errors.

Process Overview:

- Begins with assessing the number of words in student denoted by **Qwords** and the associated mark for grammar, denoted by Qmark.
- Defines a threshold for grammatical errors (*glevel*), which will determine allowed grammatical errors in student answer.

Calculation of Grammar Mark ($g$mark**):**

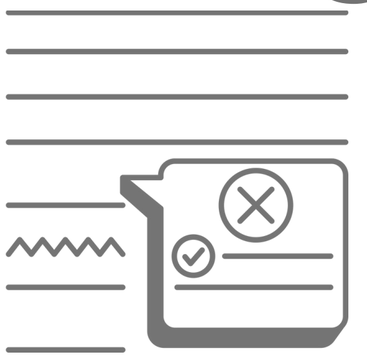
$$g\text{mark} = q\text{mark} \times \frac{w_3}{w_1+w_2+w_3}$$

- w1, w2, and w3 are the weightage ratios for splitting a question_mark.

NEXT

EVALUATION

Grammar Checking



Comparison Against Threshold:

- Compares the grammatical error count for each student response ($sa\_gerrors[i]$) against the predetermined threshold ($g$level×Qwords).
- If the errors exceed the threshold, adjusts the grammar mark:

$$g_{\text{mark}} = g_{\text{mark}} - \frac{\text{grammatic\_error}}{Q_{\text{words}}}$$

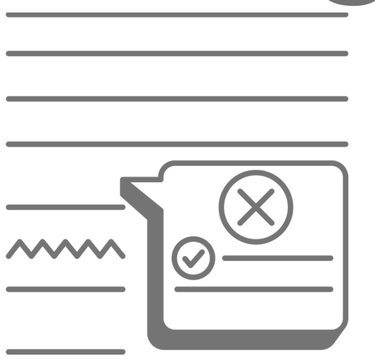
Resulting Grammar Mark (sa_gmark**):**

- Provides a quantitative measure of grammatical errors for each student response.

NEXT

EVALUATION

Grammar Checking



```
glevel=0.1
def grammermark(Qwords,Qmark,grammatic_error):
  gmark=Qmark * w3/(w1+w2+w3)
  if grammatic_error <= glevel*Qwords:
    return gmark
  else:
    gmark= gmark - (grammatic_error / Qwords)
    return gmark
sa_gmark=[]
for i in range(len(sa)):
  sagm=grammermark(Qwords,Qmark,sa_gerrors[i])
  sa_gmark.append(sagm)
print("Corresponding  grammar mark for each student:",sa_gmark)
print(Qwords)
```

```
No.of Errors in Each student Answer: [2, 1, 1]
Corresponding  grammar mark for each student: [0.8666666666666667, 1.0, 1.0]
15
```

NEXT

EVALUATION

Converting To Marks

Components of Scoring:

- Semantic Marks:
  - Assess how well students understand the question.
- Word Marks:
  - Evaluate if students write an adequate number of words compared to the expected length.
- Grammar Marks:
  - Analyze the correctness of grammar and punctuation in student responses.

```
final_mark=[]
for i,j,k in zip(sa_gmark,sa_wordmark,sim_marks):
  final_mark.append(i+j+k)

print(final_mark)
print(qmarks)
```

```
[2.4706797122955324]
[3]
```

NEXT

Results

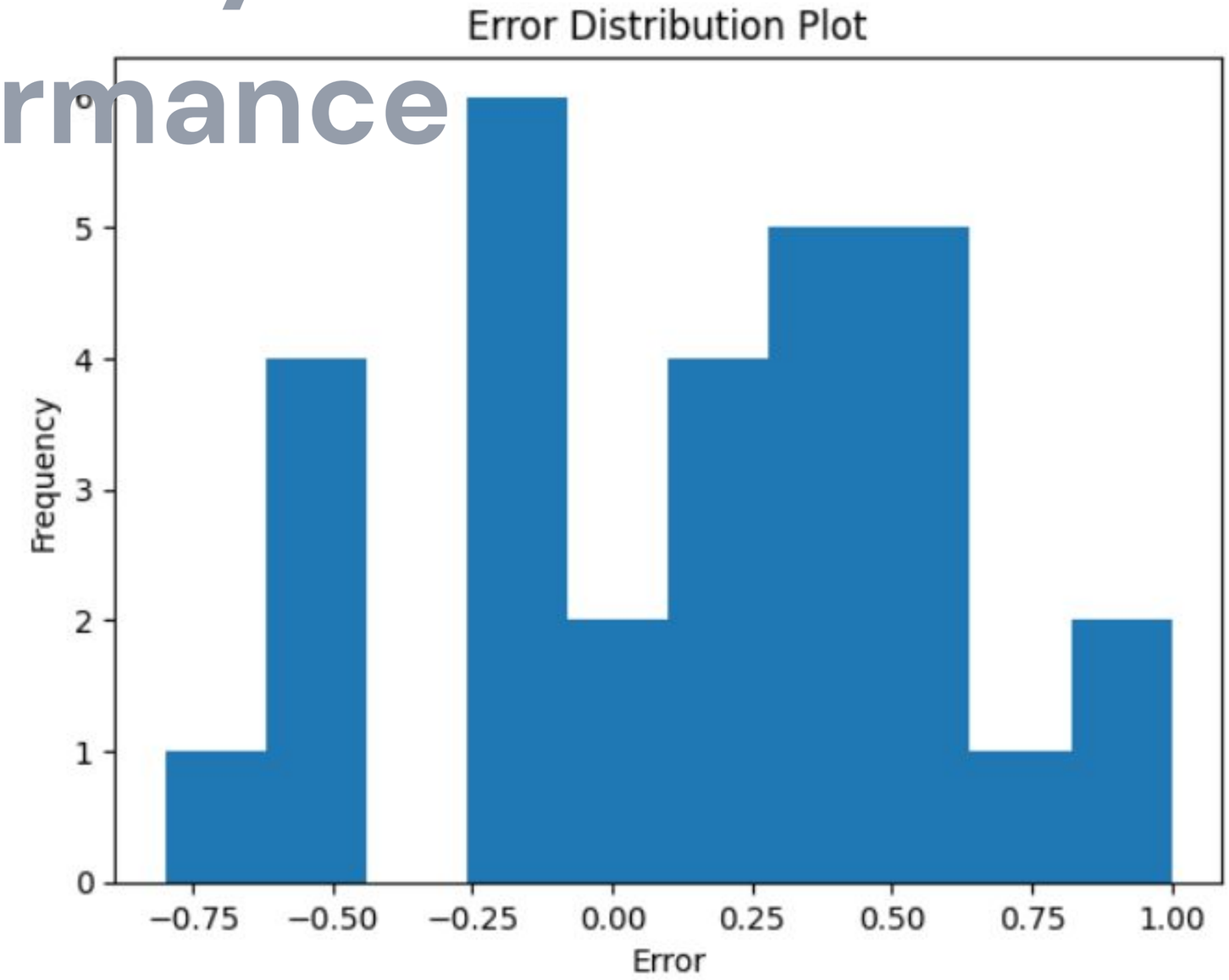
Through comprehensive analysis and evaluation, Nexus uncovers valuable insights into the performance of our system, going over its efficacy in enhancing the evaluation process. Let's explore the outcomes of our efforts and the implications for the future of education.

NEXT➤

Overall System Performance

Index	Marks (Nexus Model)	Marks (Other Model)
1	5	2
2	2.42	2
3	2.4	2.5
4	3.52	3
5	3.63	3.5
6	4	3.5
7	3.86	3.5
8	3.89	4
9	3.57	3
10	3.804	3.5
11	4.68	4.5
12	3.4	3.5
13	2.05	2.5
14	2.32	2.5
15	2.5	2
16	2.02	2
17	2.25	2.75
18	2.25	2.25
19	3.81	3.5
20	3.85	3
21	3.78	3.5
22	4.15	3.5
23	4	3
24	2.6	2.5
25	5.3	5.5
26	5.2	6
27	5.3	5.8
28	2.25	2.5
29	2.3	2.75
30	2.35	2

Overall System Performance



Overall System Performance

- CORRELATION COEFFICIENT: CALCULATE THE CORRELATION COEFFICIENT BETWEEN THE TWO SETS OF MARKS TO MEASURE THE STRENGTH AND DIRECTION OF THEIR LINEAR RELATIONSHIP.

  CORRELATION COEFFICIENT: **0.9164738461819588**

- RMSE QUANTIFIES THE AVERAGE MAGNITUDE OF THE DIFFERENCES BETWEEN PREDICTED AND OBSERVED VALUES, GIVING MORE WEIGHT TO LARGER ERRORS.

  RMSE: **0.4463860810255326**

- MAE MEASURES THE AVERAGE ABSOLUTE DIFFERENCE BETWEEN CORRESPONDING MARKS AND PROVIDES AN INDICATION OF THE AVERAGE DISCREPANCY BETWEEN THE TWO MODELS.

  MEAN ABSOLUTE ERROR: **0.3734666666666666**

References

1. S. Singh, O. Manchekar, A. Patwardhan, U. Rote, S. Jagtap, and H. Chavan, "Tool for Evaluating Subjective Answers using AI (TESA)," in *2021 International Conference on Communication Information and Computing Technology (ICCICT)*, Mumbai, India, 2021, pp. 1-6. DOI: [10.1109/ICCICT50803.2021.9510119](https://doi.org/10.1109/ICCICT50803.2021.9510119).

2. X. Hu and H. Xia, "Automated Assessment System for Subjective Questions Based on LSI," in *Third International Symposium on Intelligent Information Technology and Security Informatics*, Jinggangshan, China, Apr. 2010, pp. 250-254.

3. A. Kashi, S. Shastri, and A. R. Deshpande, "A Score Recommendation System Towards Automating Assessment In Professional Courses," in *2016 IEEE Eighth International Conference on Technology for Education*, 2016, pp. 140-143.

4. S.K. Saha, CH Rao, and D., "Development of a practical system for computerized evaluation of descriptive answers of middle school level students," *Interactive Learning Environments*, vol. 1, pp. 1-14, 2019.

5. S. Vij, D. Tayal, and A. Jain, "A Machine Learning Approach for Automated Evaluation of Short Answers Using Text Similarity Based on WordNet Graphs," *Wireless Personal Communications*, vol. 111, no. 2, pp. 1271-1282, 2020.

Thank

You!